NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Data Science and Business Analytics"

**Software Project Report on the Topic:**

# Open source recurring donation service via Faster payment system (FPS) via widgets for websites

**Fulfilled by the Student:**

group #БПАД212  Lavrov Andrei Sergeevich

Signature — Surname, First name, Patronymic, if any

26.05.2023
Date

**Checked by the Project Supervisor**:

Konnov Georgy Alexandrovich
Surname, First name, Patronymic (if any), Academic title (if any)

E-commerce Product Owner, Executive Director
Job

Raiffeisenbank JSC
Place of Work (Company or HSE Department)

Date 02.06.2023 

Signature

**Moscow 2023**

# Contents

# Annotation

Our project aimed to create an open source recurring donation service by leveraging the Faster Payment System (FPS) technology. We developed customizable widgets that can be easily integrated into websites, providing a user-friendly interface for accepting recurring donations. The project progressed through several sprints, each focused on specific tasks.

In Sprint 1, we successfully added a widget to the main page using an iframe and created a demo page that would later serve as our landing page. Sprint 2 was dedicated to redesigning the landing page and improving the widget's appearance. We also implemented a subscription option, a button to modify the widget code, and deployed the project from GitHub to a domain for accessibility.

During Sprint 3, we expanded the widget's functionality by adding different sizes and created a login interface for the personal account. Additionally, we began exploring methods to request and obtain the necessary QR code. Sprint 4 involved a comprehensive redesign of the widgets, removing unnecessary dimensions and introducing new ones.

In Sprint 5, we focused on streamlining the donation process further. We eliminated the card payment button from the widgets and successfully implemented requests to generate QR codes using an API. Moreover, we started designing notifications to inform donors about successful payments.

During Sprint 7, our main objective was to enhance the usability of the widgets and make them more user-friendly. Additionally, we developed a configurator specifically tailored to one widget for a particular merchant, offering more customization options.

Throughout the project, we held two important meetings that greatly influenced our progress. In Meeting 1, we received valuable feedback from Raiffeisenbank employees, highlighting areas for improvement in terms of functionality and product presentation. Meeting 2 involved representatives from the System of Fast Payments, where we had the opportunity to demonstrate the efficiency of our service, explore potential expansion ideas, and receive feedback on subscription registration and presentation accuracy. These meetings provided invaluable insights and enhanced our team's communication skills.

In summary, our open source recurring donation service project successfully implemented the Faster Payment System (FPS) technology through customizable widgets for websites. We strived to create an intuitive and secure donation experience, constantly improving the service based on feedback received from industry professionals. The project's open source nature fosters collaboration and innovation, ensuring ongoing development and potential future expansions.

Наш проект был направлен на создание сервиса регулярных пожертвований с открытым исходным кодом, используя технологию Система быстрых платежей (СБП). Мы разработали настраиваемые виджеты, которые можно легко интегрировать в веб-сайты, обеспечивая удобный интерфейс для приема повторяющихся пожертвований. Проект проходил в несколько этапов, каждый из которых был сосредоточен на конкретных задачах.

В Sprint 1 мы успешно добавили виджет на главную страницу с помощью iframe и создали демонстрационную страницу, которая позже послужит нашей целевой страницей. Sprint 2 был посвящен редизайну целевой страницы и улучшению внешнего вида виджета. Мы также внедрили опцию подписки, кнопку для изменения кода виджета и развернули проект с GitHub в домен для обеспечения доступности.

Во время Sprint 3 мы расширили функциональность виджета, добавив различные размеры, и создали интерфейс входа в личный кабинет. Кроме того, мы начали изучать способы запроса и получения необходимого QR-кода. Sprint 4 включал в себя комплексный редизайн виджетов, удаление ненужных размеров и введение новых.

В Sprint 5 мы сосредоточились на дальнейшей оптимизации процесса пожертвования. Мы убрали кнопку оплаты картой из виджетов и успешно внедрили запросы на генерацию QR-кодов с помощью API. Более того, мы начали разрабатывать уведомления для информирования доноров об успешных платежах.

Во время Sprint 7 нашей главной целью было улучшить удобство использования виджетов и сделать их более удобными для пользователя. Кроме того, мы разработали конфигуратор, специально адаптированный к одному виджету для конкретного продавца, предлагая больше возможностей настройки.

На протяжении всего проекта мы провели две важные встречи, которые сильно повлияли на наш прогресс. На встрече 1 мы получили ценные отзывы от сотрудников Райффайзенбанка, которые выделили области для улучшения с точки зрения функциональности и презентации продукта. Во встрече 2 приняли участие представители Системы быстрых платежей, где у нас была возможность продемонстрировать эффективность нашего сервиса, изучить потенциальные идеи расширения и получить отзывы о регистрации подписки и точности представления. Эти встречи дали бесценную информацию и улучшили коммуникативные навыки нашей команды.

Таким образом, в нашем проекте службы периодических пожертвований с открытым исходным кодом успешно внедрена технология более быстрой платежной системы (СБП) с помощью настраиваемых виджетов для веб-сайтов. Мы стремились создать интуитивно понятный и безопасный способ пожертвования, постоянно совершенствуя сервис на основе отзывов, полученных от профессионалов отрасли. Открытый исходный код проекта способствует сотрудничеству и инновациям, обеспечивая непрерывное развитие и потенциальное расширение в будущем.

# Introduction

**Relevance :**

In the modern world, people tend to help each other in different situations. Hundreds of millions are transferred daily from different countries to the accounts of charitable foundations to help those in need. The 21st century is the age of technology and digitalization. Every year there are more and more new services that simplify the donation process. One of these improvements is the fast payment service. It is this service, developed in Russia, that allows you not to enter the details of your bank cards on the Internet, but simply using the camera of your phone to scan the Qr code, which has recently only been gaining popularity and increasingly appears in front of people in supermarkets at the checkout, at the terminal and even on TV screens when watching everyone's favorite gear.

We decided to create widgets that accept payment in this way. The user does not need to think about where his card is located, what number and code it has, it is enough just to have a phone in his hands. Even when you walk down the street or enter the subway, after seeing the QR code, you will easily take out your phone, scan it and be able to make a donation. Foundations will no longer have to explain to people how to make a donation for a long time. If earlier and now we see advertisements on the screens of monitors and televisions where they offer to help and send a message to a phone number, which is quite difficult and does not look very reliable, then in the future, a QR code will appear on the same screens, according to which all transactions will take place.

Every day millions of people visit many different sites, and they all support charity in one way or another. They won't have any problems using our service to create their own personalized widget and easily paste the resulting code into their own website. It costs nothing to them because it's absolutely free. Why widgets, because it may seem so complicated and cluttered? In fact, no, now FPS has the opportunity to subscribe to monthly write-offs by pressing one button. Now the clients of these or other services will not have to fill out long questionnaires, enter their data and everything like that, it will be enough to activate the subscription button on our widget and it will register itself. You can cancel it just as easily in the bank's application.

Now almost everyone has their own page on the Internet or even a blog, you are an ordinary user, you can also use our service by creating your own unique widget and posting it in your profile. Perhaps it is your similar post that will save someone's life or just help those in need. Or maybe you are a creative person that other people want to help develop, but don't know how to donate money. With our widgets, such questions should become much less.

**Goal:**

To develop a service where users can register and create their own custom widgets for donations with payment by QR code.

**Tasks:**

Analyze similar solutions already available on the market to understand which way to move and how to improve your final product.

Learn the programming languages JS, HTML, CSS.

Develop the design of the main page and widgets.

Develop a constructor for creating custom widgets by users.

Test the operation of all the main functions on real users.

Present your project to the supervisor for approval of all the achieved tasks.

To check the possibility of making a payment so that the funds are not lost and transferred to other accounts.

**Instruments:**

To construct our service, we employed a variety of tools, including:

- VSCode
- Google Documents
- Power Point
- Jira
- RaifAPI

## Basic terms and definitions

**Donation -** a money, which someone gives to help others.

**Web Page -** an online document that can be viewed through a web browser and contains information, text, images, videos, and other elements.

**User -** a person who uses a product, service, or technology, such as a website or widget.

**Database -** a system for storing and organizing data in a structured manner for efficient retrieval and manipulation.

**Widget -** a graphical user interface element, such as a button or a box, that allows a user to interact with and control various functions of a software application or a website.

**Iframe -** an HTML element that allows a website to embed another HTML document within its own webpage, creating a nested frame within the main content.

**Frontend Development -** designing, coding, and optimizing the user interface and user experience of a website or application, with a focus on the parts of the product that are visible to the end user.

**API -** an API (Application Programming Interface) is a set of protocols and tools for building software applications that allows communication and data exchange between different systems, enabling one application to access the functionality or data of another.

**Asynchronous request -** a type of computer request where the requesting process continues to run without waiting for the requested task to complete, allowing for multiple tasks to be performed simultaneously.

**Custom sum -** the amount that the user enters.

**Open-source code -** code that is made available to the public, allowing anyone to use, modify, or distribute the code as they see fit, typically with the condition that any derivative works must also be open source.

**Faster payment system -** an electronic payment method that enables quick and convenient transfer of funds between individuals and businesses.

**Configurator** - a developer mode where the user can create a widget for their tasks

**URL** - a single user pointer

# Description on functional and non-functional requirements

**Functional requirements**

The main functions that are available to the user

1. Payment by the user using a fixed amount widget
2. Payment by the user using the widget of their own amount
3. Make not a one-time payment, but a monthly one
4. The ability to copy the code of standard widgets
5. The ability to register in the service

After successful registration, an additional section will open for the user. In particular, a personal account, where additional functions will be available to each user

1. Create your own widget with QR code
2. Copy the code of the created widget
3. Save this widget in your personal account for future use

**Non-functional requirements**

The device must support launching a web page in the browser

The device must be able to connect to the Internet

## List of completed tasks

| Sprint | Completed Tasks |
|---|---|
| Sprint 1<br>until 06.12.2022 | 1. Added a widget via an iframe to the main page<br>2. We created a demo page, which will later be reworked into a landing page |
| Sprint 2<br>until 20.12.2022 | 1. The landing page has been completely redesigned<br>2. Changed the appearance of the widget<br>3. Subscription option added<br>4. Button on the widget to change the code<br>5. Adaptive for the site<br>6. Project deployment from github to domain |
| Sprint 3<br>until 18.01.2022 | 1. Added different widget sizes<br>2. Created the interface of the login window to the personal account<br>3. Moved on to learning how to send requests to get the necessary Qr code |
| Sprint 4<br>until 01.02.2023 | 1. Completely redesigned widgets<br>2. Removed unnecessary dimensions and made new, necessary ones<br>3. Removed the subscribe button from widgets<br>4. Linked the card payment page to them |
| Sprint 5<br>until 15.02.2023 | 1. Removed the card payment button from the widgets<br>2. We have sent requests to create a QR code via the API<br>3. Made receiving requests about the payment status<br>4. We started to design notifications about successful payment |
| Sprint 6<br>until 09.03.2023 | 1. Tried to make a registration in your personal account<br>2. Minor changes in the website design<br>3. Minor updates for better performance |

| | |
|---|---|
| Sprint 7<br>until 23.03.2023 | 1. Made widgets easier to use, user friendly<br>2. We made a configurator for one widget for one merchant |
| Meeting 1 | Meeting with Raiffeisenbank employees who pointed out our existing shortcomings in the functionality of the service. And on the mistakes of the presentation of our product |
| Meeting 2 | Meeting with representatives of the System of fast payments via Zoom. At the meeting, we demonstrated the efficiency of our service, told what key problems it can solve. We shared ideas where we can expand our project. We listened to the comments related to the registration of subscriptions, with the inaccuracies of our presentation. After this meeting, each team member gained invaluable experience in communicating with professionals. |

# Main Part

Before starting our team work, it was decided to choose a person who will be responsible for the main communication with the project manager. I, Andrey Lavrov, a 2nd year DSBA student, became such a person. Our supervisor gave us the direction and, accordingly, the topic of our project. In this project, it didn't matter what we came up with, a website, a service or an application, the main thing was to achieve the goal. I decided to first outline the entire project for the team, so that later it would be assembled in parts and go every two weeks to some specific result. The idea of sprints was suggested to us by our supervisor and we accepted this advice.

**Sprint 1. Lasted until 06.12.2022**

During this time, we have studied a database of information concerning all the basics of programming languages, which were useful to us during the whole work. From my side, these were FrontEnd development languages, HTML, CSS, JS. It was quite difficult to study this independently from textbooks, so it was decided to purchase a course.

During these two weeks, a demo page was also made, which was supposed to be used in the future, since our project was connected with widgets that needed to be conveyed in some way to the end consumer. (Fig. 1) Another member of the team created our first widget, worked out its design and a small, at that time, functional. We added this widget to the page of our website by inserting it through an Iframe.(Fig. 2)

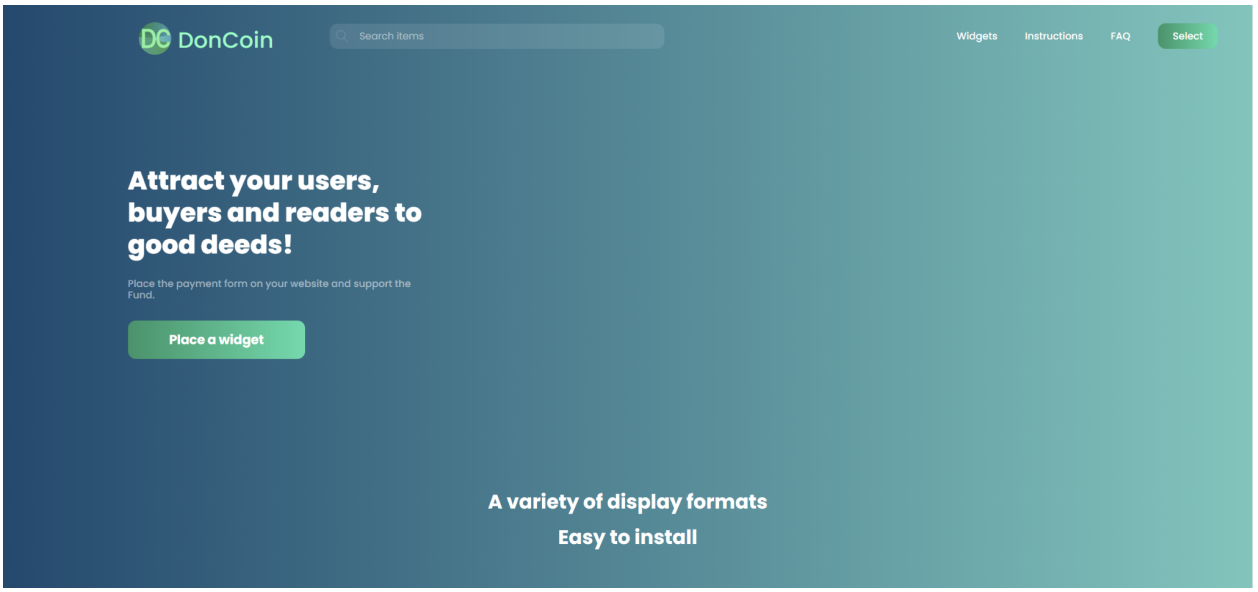

DonCoin
Search items
Widgets
Instructions
FAQ
Select
Attract your users, buyers and readers to good deeds!
Place the payment form on your website and support the Fund.
Place a widget
A variety of display formats
Easy to install

Fig. 1

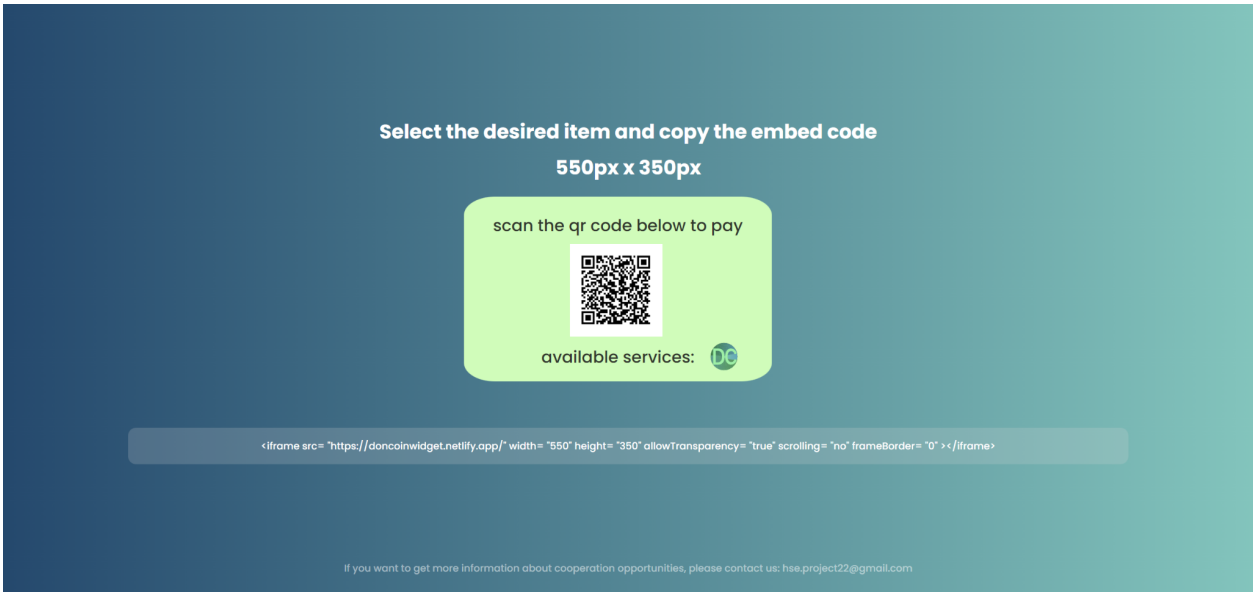

Select the desired item and copy the embed code
550px x 350px
scan the qr code below to pay
available services:
<iframe src="https://doncoinwidget.netlify.app/" width="550" height="350" allowTransparency="true" scrolling="no" frameBorder="0" ></iframe>
If you want to get more information about cooperation opportunities, please contact us: hse.project22@gmail.com

Fig. 2

**Sprint 2. Lasted until 20.12.2022**

In my opinion, it was the most productive sprint for 2022 and one of the most productive for the entire project. A colossal part of the work has been done on my part. It was after the first sprint that we were introduced to the staff of Raiffeisen Bank, and in particular to the web designer, who at the first call offered her help in any issues related to the placement of objects on the page and with design in general. Also at that moment, I personally experienced difficulties as a project manager of this project, because it was my first experience. The Raif Scrum master was present at the call, who after the call helped me understand how to act and where to move, offered to read a small but very useful Scrum guide, which later of course led to the desired result.

Now in more detail about the technical part of this sprint and what was done.
First of all, it was decided with the designer that such an appearance of the main page is not well suited for our project both in terms of color scheme and design principle. That is why I completely abandoned the previous decision and wrote a new landing page from scratch, taking into account all the tips and comments. So we got the main page, which is still in use today. (Fig. 3.1; 3.2; 3.3)

Due to the change in the overall design, it was impossible to leave our widget in the past design. I completely redesigned the widget that we had, taking as a basis the idea of my teammate. (Fig. 4) The new widget fits well into our new page. Since the project was based on FPS payment, it was no longer surprising to anyone with the usual QR code, and we went towards signing up for the same QR at the touch of a button. I have added a subscription button to our new widget.

Now it is a good tone when your page on the Internet can be opened not only from a home computer, but also from a laptop, tablet and phone. We made an adaptive layout for a laptop. Looking ahead, I will say that we were going to make an adaptive version for all devices, but decided to abandon this idea in favor of more important tasks facing the team at that time. It was during this sprint that I decided to move our project to the domain so that anyone from our team, 4 teams and a bank employee, could see what we were doing. To do this, we deployed the project from Github to the domain https://doncoin.site/

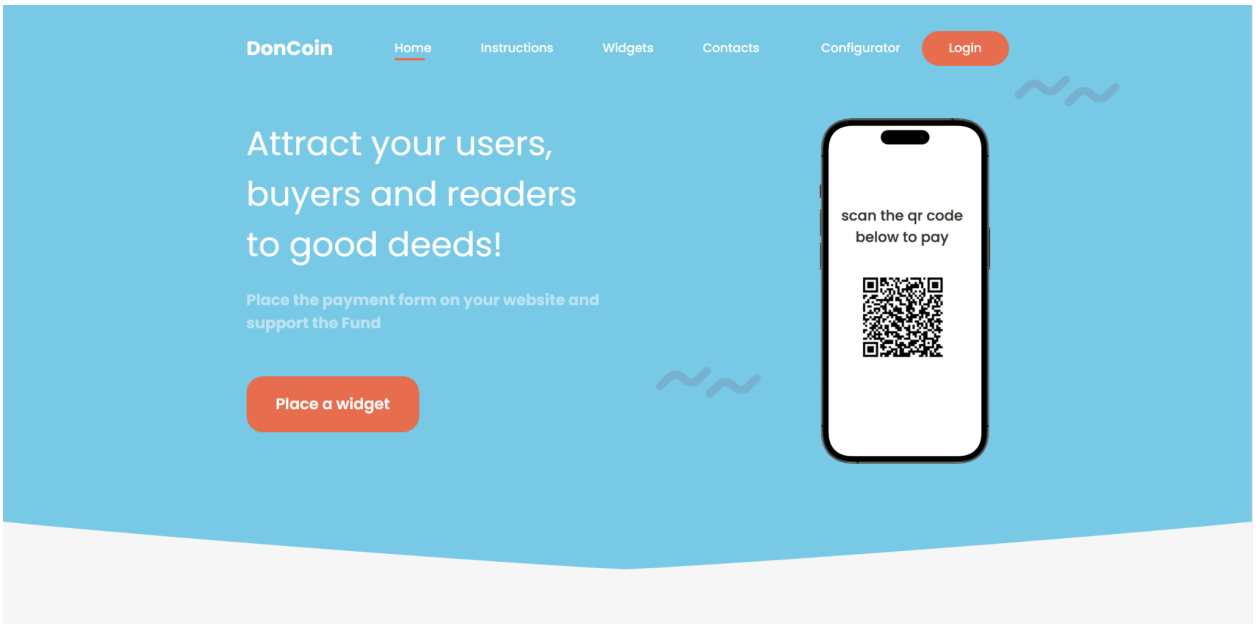



Fig. 3.1

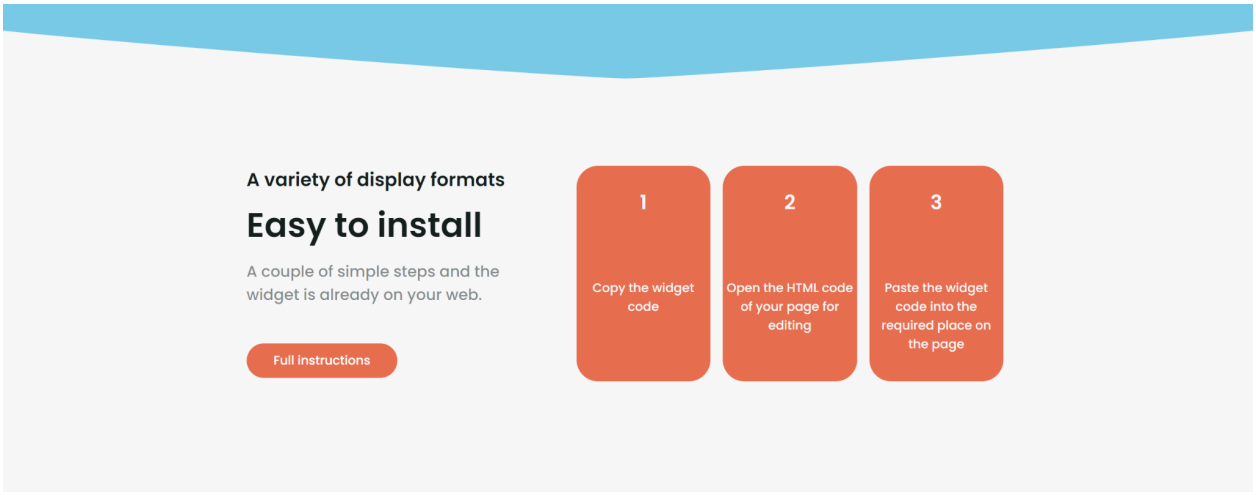



Fig. 3.2

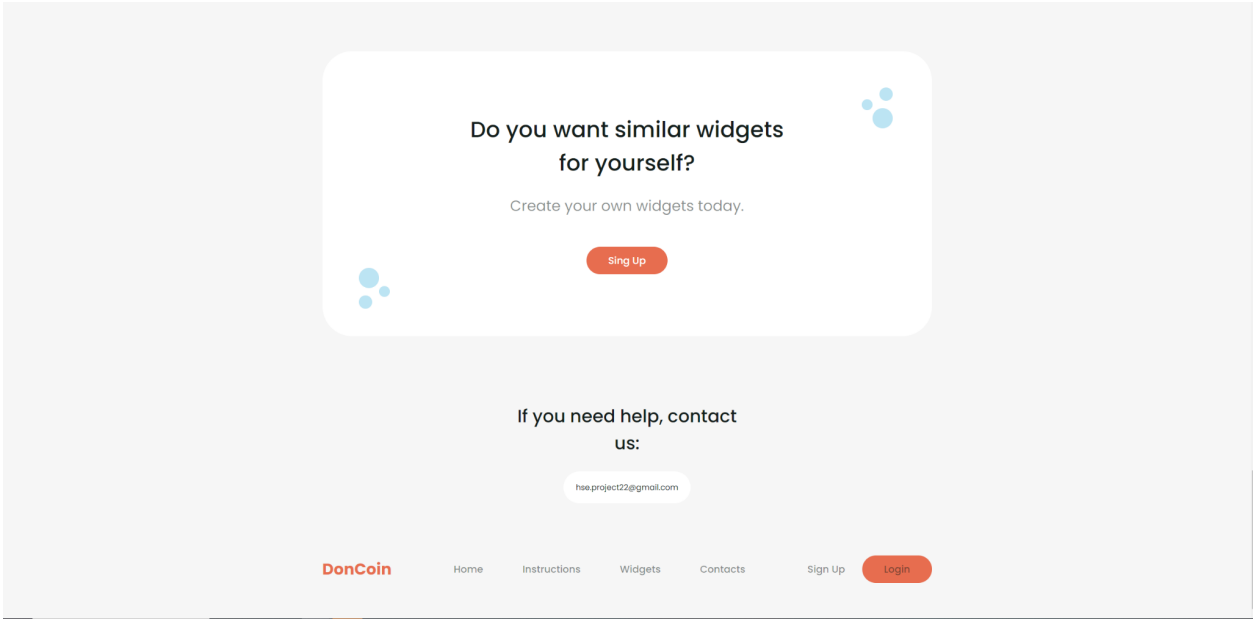



Fig. 3.3

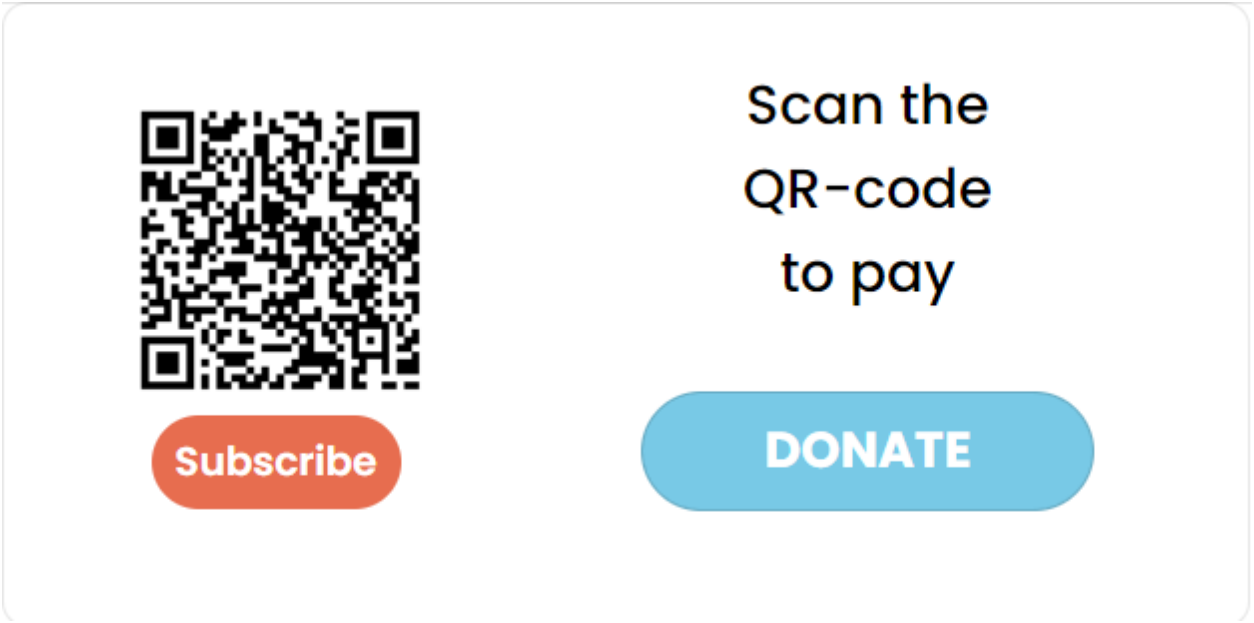



Fig. 4

**Sprint 3. Lasted until 18.01.2023**

During this sprint, I added a few more widget sizes, and to be precise, the widget was redone again, but not from scratch, but its design was slightly changed, the dimensions were more standard and a new size was added, smaller and thus more versatile and a button widget.

I wrote the interface of the login window to my personal account.

We started studying the API, how to send requests to get the necessary QR code.

**Sprint 4. Lasted until 01.02.2023**

Widgets have been redesigned again and it was decided to abandon the button widget, because with this format the meaning of our project is completely lost, since there is no place for the QR code with which all actions are connected. (Fig. 5)

I removed the subscription button from the widgets, due to the fact that we learned that subscriptions should be issued directly at the time of payment via FPS technology.

We have linked a button to our widget-a link to redirect to payment by card. (Fig. 5)

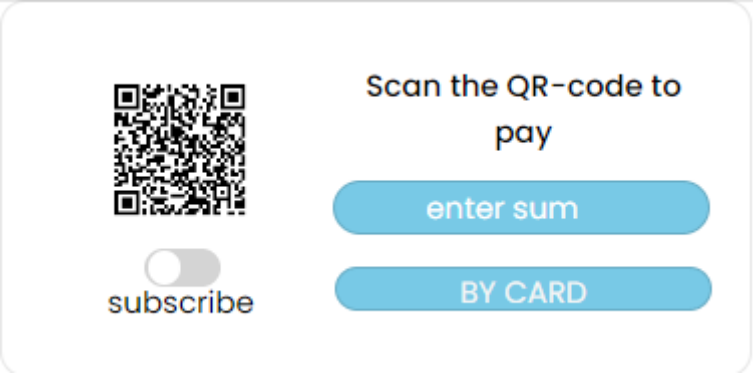


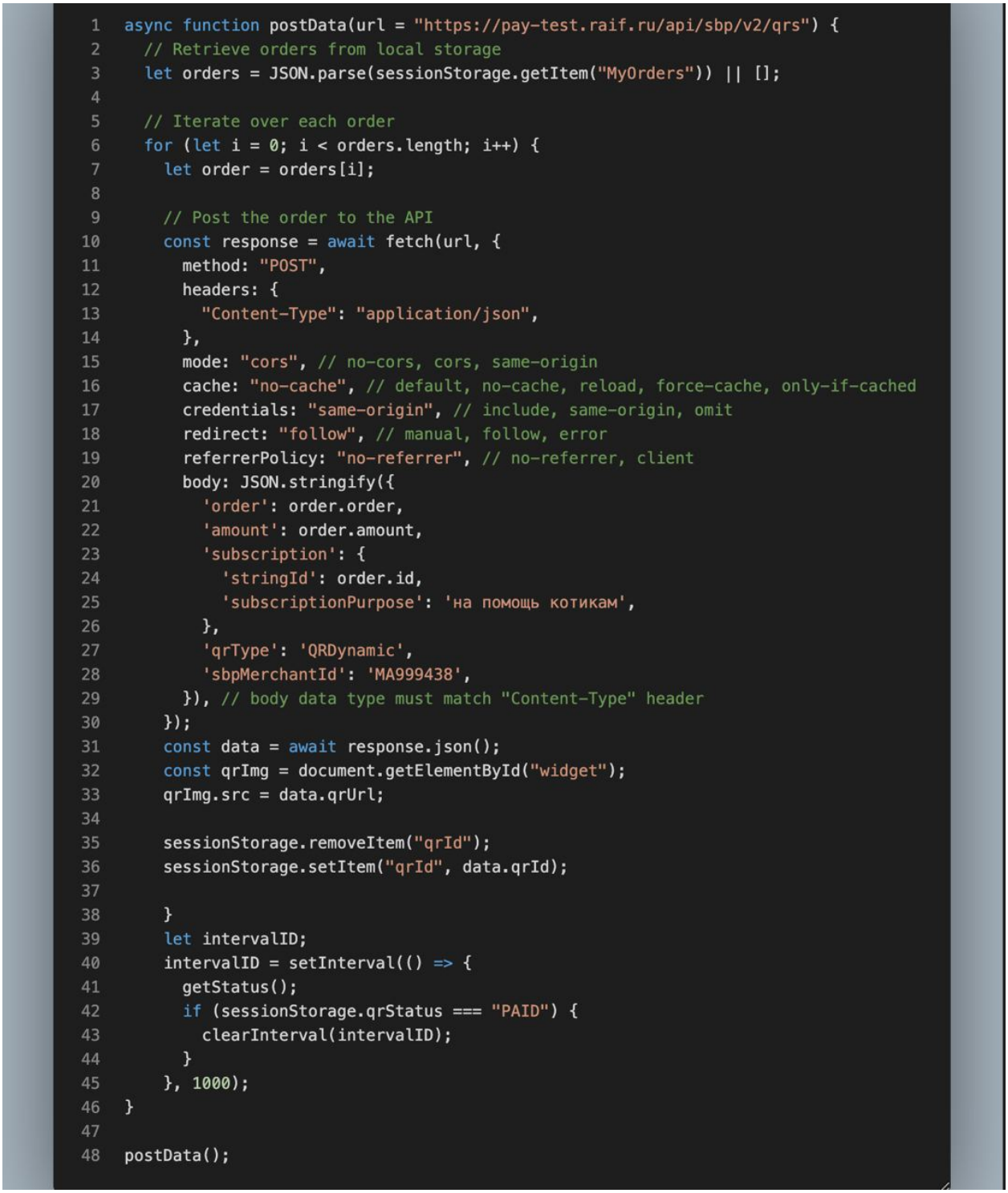

Fig. 5

**Sprint 5. Lasted until 15.02.2023**

We were told that card payment on a separate page is possible only under a special license or when acquiring is enabled, which was impossible for us to do, so we had to abandon this idea and completely cut this function from our widgets.

Finally figured out the REST API. Together with my colleague on the team, we were able to make asynchronous requests to the server to receive, process and send QR-code at the request of users using a function on widgets. On my part, a Post request was written to send data to the server from users. (Fig. 6)

We also wrote payment notifications so that after a successful payment, the user understands that his money has not just disappeared from the account, but has gone through our service. The design for these notifications was invented by me. (Fig. 7) All members of our callers liked this appearance, which certainly pleased them.



```
async function postData(url = "https://pay-test.raif.ru/api/sbp/v2/qrs") {
  // Retrieve orders from local storage
  let orders = JSON.parse(sessionStorage.getItem("MyOrders")) || [];

  // Iterate over each order
  for (let i = 0; i < orders.length; i++) {
    let order = orders[i];

    // Post the order to the API
    const response = await fetch(url, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      mode: "cors", // no-cors, cors, same-origin
      cache: "no-cache", // default, no-cache, reload, force-cache, only-if-cached
      credentials: "same-origin", // include, same-origin, omit
      redirect: "follow", // manual, follow, error
      referrerPolicy: "no-referrer", // no-referrer, client
      body: JSON.stringify({
        'order': order.order,
        'amount': order.amount,
        'subscription': {
          'stringId': order.id,
          'subscriptionPurpose': 'на помощь котикам',
        },
        'qrType': 'QRDynamic',
        'sbpMerchantId': 'MA999438',
      }), // body data type must match "Content-Type" header
    });
    const data = await response.json();
    const qrImg = document.getElementById("widget");
    qrImg.src = data.qrUrl;

    sessionStorage.removeItem("qrId");
    sessionStorage.setItem("qrId", data.qrId);

    }
    let intervalID;
    intervalID = setInterval(() => {
      getStatus();
      if (sessionStorage.qrStatus === "PAID") {
        clearInterval(intervalID);
      }
    }, 1000);
}

postData();
```

Fig. 6

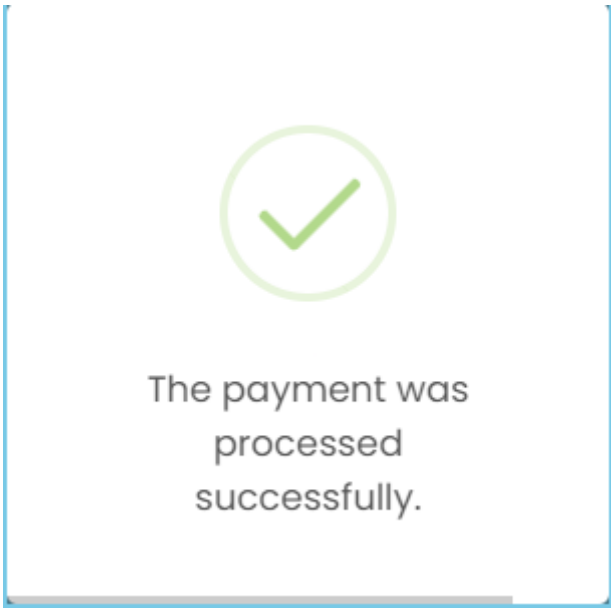


Fig. 7

**Sprint 6. Lasted until 09.03.2023**

For the most part of the team, we tried to register for a personal account on our service, but we could only do it on a local server and therefore decided to temporarily refuse registration.

This sprint has no visible results, but it is by no means less significant for the overall picture, because work has been done to improve the performance of our service, a large number of bugs and holes left during the initial development have been removed.

**Sprint 7. Extended until 23.03.2023**

For the menu, it turned out to be one of the most difficult, but also one of the most significant sprints. Because it was during this period that we managed to implement the idea with a configurator for widgets. This direction did not leave me almost from the very beginning of our project, but the lack of knowledge did not allow me to implement it in any way. But for the whole project, we managed to get the necessary skills and finally cope with the task. (Fig. 8) This configurator still has limited functionality for modification, but you can easily add new features for customization of widgets there. Also, so far, the configurator works for one merchant, us. This was done for ease of development. In the future, it is assumed that when registering with the bank, customers will receive a unique MerchantId, which they will use for themselves in the configurator.

And of course, where it is already without this, but the widgets have been redesigned again, making them understandable and easy to use. Added a feature that after a successful payment, customers saw the original logo again and their QR code disappeared. Removed the extra text from the widgets. (Fig. 9)

Finally, we managed to make payments of fixed amounts on our largest widget. (Fig. 10)

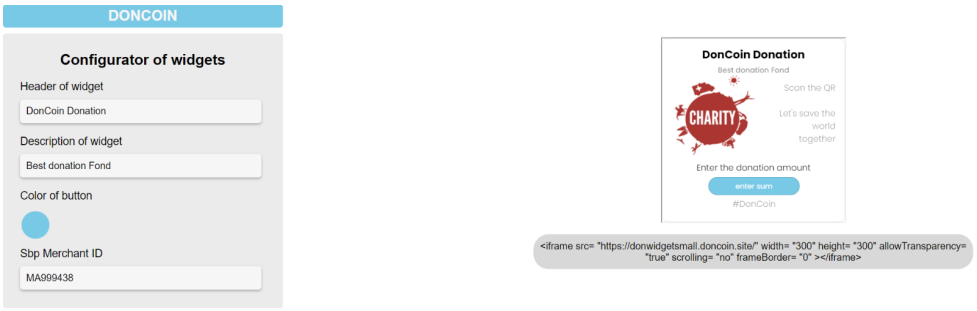



Fig. 8


Fig. 9

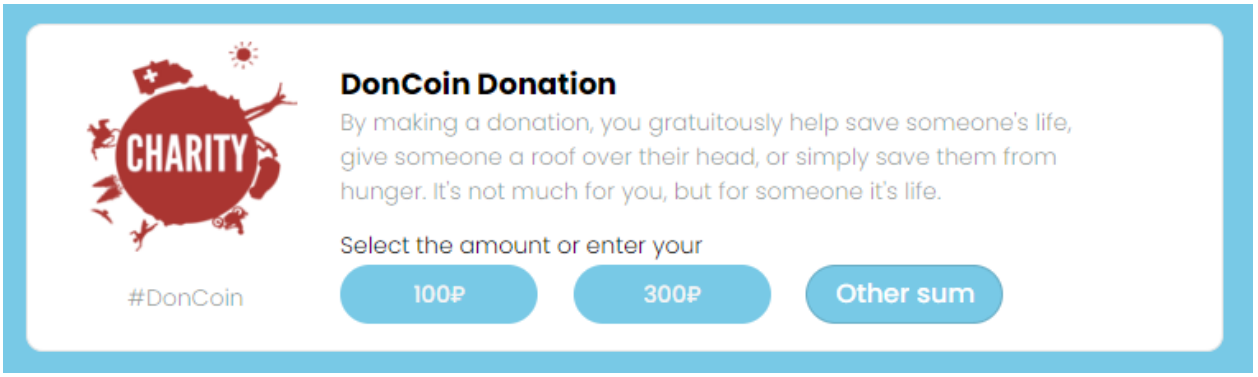



Fig. 10

**Meetings**

Since our project had 4 teams from different directions of the FCN and from different courses 2-4, the topics were divided one by one into two teams. Thus, there was such a kind of competition among all those who were engaged in this direction, and at the same time, we all learned from each other's mistakes, gave ideas and helped to find solutions to various problems.

Our supervisor organized 2 significant meetings on our topics. The first meeting was with the bank's employees, who were invited and who were not aware of what we were doing all this time. Our task was to convey to them the relevance of our project, its problems and how we found a solution to this problem. This meeting was a trial and allowed us to find out the wishes of real users. The points worth focusing on were coordinated. We were also able to see technical errors that were immediately corrected.

All this allowed us to prepare well for the next meeting with the creators of FPS. With the people on whose technologies we made our project. Our team was able to focus attention on the fact that with the help of their fast payment service, it is possible to solve not quite banal problems that they overlooked. We showed them our view on many things that will be useful in the work of their services. FPS representatives highly appreciated the work we have done. Its relevance was recorded and the horizons of its expansion were discovered.

## Review and comparison of sources and analogues

| | Our service | GiveLively | Boomstarter | CloudTips |
|---|---|---|---|---|
| Web-Page | + | + | + | + |
| Free access | + | - | + | - |
| UserFriendly | + | + | - | + |
| Payment by QR | + | - | - | + |
| Examples of Widgets | + | + | - | + |
| Custom Sum | + | + | + | + |

Due to this table, we can conclude that after analyzing analogues on the market, we took into account all the points that users need and made them. First of all, our main advantage is the fact that the service is completely free for full functionality. Another advantage is that payments are made using the system of quick payments via QR-code, thereby users' card data does not get into the Internet. Also on the main page there are examples of our widgets, thanks to which users understand what they will get in the end.

For this comparison we have used some sources:

- GiveLively: https://www.givelively.org/
- Boomstarter: https://boomstarter.ru/start
- CloudTips: https://cloudtips.ru/

| | Our service | SberLead | Raif | Tinkoff |
|---|---|---|---|---|
| Ease of use | + | - | + | - |
| Free access | + | - | + | - |
| UserFriendly | + | + | + | + |
| Customization | + | + | + | + |
| Examples of Widgets | + | + | + | - |
| Ability to copy the code | + | - | + | + |

Due to this table, we can conclude that after analyzing the analogues on the market, we took into account all the points that users need and made them. This comparison is aimed at analyzing similar solutions for widget configurators. We see that not all solutions on the market are free. Some options, for example from Tinkoff, are quite complicated for an ordinary user who is far from programming, our solution, as well as the option from Raiffeisen Bank, are convenient and easy to use even for a person who does not know how to programming. Simplicity is the main feature of this service.

For this comparison we have used some sources:

- Tinkoff: https://www.tinkoff.ru/kassa/develop/widget/install/
- SberLead: https://sberlead.ru/vidgets
- Raif: https://pay.raif.ru/pay/configurator/#/

# Conclusion

During the project, I was able to analyze existing solutions in order to understand the relevance of the problem and determine the vector of further development.

I have studied various FrontEnd development languages, namely HTML, CSS, JS, as well as documentation for the RaiffeisenBank API in OpenAPI format. We managed to work directly with the API and asynchronous requests.

To come up with my own idea of a service, I had to analyze several similar services and highlight their main advantages and disadvantages.

As a result of the project, it was possible to get the skills that were necessary to create a configurator. The result of this was the creation of the configurator itself for one merchant.

I created a demo page that served as the basis for the entire project, and all widgets and other functionality were subsequently added to it.

The project was deployed on its own hosting on the Internet.

At the moment, there is a page on the Internet that hosts two fully working test widgets with the ability to pay for FPS and subscriptions, a configurator for customizing widgets to your brand, color and taste.

This project allowed us to understand teamwork, how to manage a small team of people, what scrum is and how to achieve certain goals at a given time.

## List of resources:

- Raif API: https://pay.raif.ru/doc/sbp.html#operation/get-sbp-v1-qr-qrId-payment-info
- HTML/ CSS/ JS courses: https://developedbyed.com/courses/enrolled'
- Sweet Alert: https://sweetalert2.github.io/#download
- GiveLively: https://www.givelively.org/
- Boomstarter: https://boomstarter.ru/start
- CloudTips: https://cloudtips.ru/
- Resources for additional learning JS and async fetch for payment: https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/JSON
- Tinkoff: https://www.tinkoff.ru/kassa/develop/widget/install/
- SberLead: https://sberlead.ru/vidgets
- Raif: https://pay.raif.ru/pay/configurator/#/
- Netlify (our hosting): https://www.netlify.com/

The project is a team project and is carried out by a group of 4 people:

1. Lavrov Andrei Sergeevich
2. Talagaeva Polina Vladislavovna
3. Levankov Artem Artemovich
4. Malikov Arseniy Ivanovich

This report was written by team member Andrei Lavrov