

KNIGHTS OF PERFORMANCE INSIGHTS



PROJECT DOCUMENTATION
TEAM KPI2 - DEPLOY (IMPACT) 2023



PROJECT DOCUMENTATION	1
1 Project Charter 📄	3
1.1 Project Purpose and Justification	3
1.2 Context and Opportunity	3
1.3 Vision and Goal	3
1.4 Project Description and Solution	3
1.5 Personas	3
1.6 Requirements	4
1.7 Technology Stack	4
1.6 Stakeholders	4
1.7 Project Constraints	5
1.8 Product MVP & Post MVP features	5
1.9 Project Roadmap	6
2. Risk Management Plan ⚠	7
2.1 Introduction	7
2.2 Risk identification & Assessment	7
2.3 Risk Mitigation Plan	8
3. Post-MVP features 🚀	10
3.1 Notifications for Both Users	10
3.2 Responsive Screens	11
3.3 Data Export Functionality	11
3.4. Cumulative KPI Option In KPI Creation Menu	12
4. UX/UI Documentation ✎	14
4.1 Tools	14
4.2 User Research	14
4.3 User Experience Design	17
5. Data Science Documentation ↘	22
5.1 Methodology	22
5.2 Required input fields for KPI creation	28
5.3 Metrics of Success: Progress and Performance	29
5.4 Visualisation Rules and Specifications	31
6. Development Documentation 🔧	44
6.1 About	44
6.2 Tech Stack	44
6.3 Getting Started	45
6.4 Table Schema	46
6.6 Running Tests	50
6.7 Features to be implemented	50
7. Team Learnings 📚	51
7.1 Team learnings from the retrospectives:	51
7.2 Summary of individual learnings:	51

1 Project Charter

1.1 Project Purpose and Justification

Organizations like Pro Juventute, with a diverse set of programs and activities, often track their KPIs, many of which cannot be measured automatically. This project aims to move away from traditional methods like Excel sheets and create a centralised KPI tracking system that enhances efficiency and transparency.

1.2 Context and Opportunity

Context: Many organizations define goals/OKRs/KPIs, but not all can be measured automatically.

Opportunity: By creating a dedicated app for this purpose, the project can enhance efficiency, streamline processes, and enable data-driven decisions for diverse teams.

1.3 Vision and Goal

Vision: To build a platform that allows the collection of data that will help Pro Juventute define goals and grow towards them.

Goal: Enable team members to manually input KPI values with minimal effort for visualisation in a dashboard.

1.4 Project Description and Solution

Develop an Open Source, Enterprise-ready web app to facilitate the consistent tracking, amendment, and visualisation of KPIs while supporting visualisation tools integration.

1.5 Personas

Gita (Team Economist):

Purpose: Enable consistent KPI tracking.

Tasks: Frequently insert valid KPI values. They will at max enter the KPIs once per month.

Pains: Confusion and errors in entering data.

Needs: A straightforward method to input values.

Helga (Gatekeeper):

Purpose: Define and maintain KPIs.

Tasks: Collaborate on KPI definition and manage the app.

Pains: Maintenance effort of the excel file that is being used now.

Needs: Ensure data quality.



1.6 Requirements

Functional:

- Single Sign On (SSO)
- CRUD operations for KPIs
- KPI actual data visualisation
- KPI progression & performance visualisation
- Change log traceability
- Option to enter targets for KPIs

Non-Functional:

- UX: Familiar/simple UI, Gatekeeper-defined KPIs
- Tooltips with info for the user

Out of Scope:

- AI assistance for KPI definition
- Team/circle hierarchy definition by gatekeepers
- Automatic data ingestion from DWH
- Integration with Jira, Trello, etc.
- m:n (one KPI can be used by multiple teams with different values)
- 1:n (one KPI can be used by multiple teams)

1.7 Technology Stack

Data science: Python & Flask. Visualisations on Apache Echarts JS library. Flask.

Backend: Python, Flask.

Frontend: Typescript, ReactJs.

Database: PostgreSQL for development, SQLite for testing and ElephantSQL for production.

Cloud Storage: Preference for Swiss region, European entity, and GDPR compliant solutions.

Deployments: Render

1.6 Stakeholders

Product Owner: Martin Seifert

Users of the app: Gita

Gatekeeper (who keeps the content, defines the KPIs) : Helga

Women ++ Luisa Cotreras: Program's project manager

Women ++ Giada Fallo: Program's technology lead

Project Management (PM): Miren Bengochea - mirentxu3@gmail.com

Scrum Master: Yaroslava Kharytonova - yaroslava.ktva@gmail.com

UX/UI Lead: Mariana Hurtado - m@rianahurtado.do

UX/UI Deputy: Elisabeth Saehr - elisabeth.saehr@gmail.com



Fullstack Lead: Alina Bunn - aussie.alina@gmail.com
Fullstack Deputy: Michael - michael.kema.cs@gmail.com
Backend Deputy: Mohamed Shaaban - shaaban.moe@gmail.com
Data Science: Diana Alvarez-Marin - diana.archi@gmail.com
Satelite: Mushfika Ahmad - mushi.ahmad@gmail.com
Mentor: Rumiyan Yankova - rumyana.yankova@capgemini.com
Deputy mentor (1st week) - lily Alemany lily.alemany@capgemini.com

1.7 Project Constraints

The application must adhere to:

- Utilise Open Source Software under the OSI licence
- Comply with Swiss privacy laws and GDPR
- Ensure compatibility with proprietary tools mentioned by the client

1.8 Product MVP & Post MVP features

Feature/ Functionality	MVP	POST - MVP
Single Sign On (SSO)	Production	
KPI input (economist)	Production	
KPI modification (economist)	Production	
KPI CRUD	Production	
Visualisations - KPI overview actual data	Figma Prototype & DS specs	
Visualisations - KPI overview progress metrics	Production	
Visualisations - KPI overview performance & progress metrics from all circles	Production	
Visualisation radial diagram with selective legend	Production	
Change logs for traceability (Activity)	Production	
Profile user screen with user change logs	Production	
Targets for KPIs	Production	
Tool tips (info mouse over)	Production	
Notifications both users		Figma Prototype
Responsive screens		Figma Prototype
Data export		Figma Prototype

Table 1. MVP and post MVP features

Production: These features are delivered as fully functional, working code and are deployed.

Figma Prototype: These features are delivered as figma prototypes because the team did not have time to implement them.

Figma Prototype & DS: The specific type of graphic to visualise the metric and the calculations, are provided in the data science documentation.

1.9 Project Roadmap

Milestones & tasks	Sprint 1:	Sprint 2:	Sprint 3:	Sprint 4:	Sprint 5:	Sprint 6:
Kick-off Meeting						
Requirement Gathering & Understanding						
Tool Selection						
MVP Definition						
Projects final presentation structure						
Development tasks						
Technology Stack Selection						
Frontend Development						
Backend Development						
Integration Backend and Frontend Development						
Testing						
Bug fixing						
Github Repository preparation to delivery Sprint 5 + licence						
UX/UI tasks						
User Research						
High fidelity final screen designs MVP						
Prototyping: Creation of high-fidelity interactive designs.						
High fidelity final screen designs Post-MVP						
Final Presentation designs						
Data Science Tasks						
Data Cleaning and Preprocessing						
Toy data set creation						
Data Visualization rules and specifications + Input fields spec.						
Data Visualization implementation						
Stakeholder Interactions						
PO meetings + demo + feedback						
Mentor meetings + Feedback collection						
Deliverables						
Team retrospective in Miro board						
Individual learnings from each team member (
Project's GitHub repository						
A LICENSE file (GPL v3 License)						
Project's final presentation						
Team's visual for final presentation						

Table 2. Roadmap

2. Risk Management Plan !

2.1 Introduction

The Risk Management Plan for the KPI Tracking Application project outlines our approach to identifying, assessing, and mitigating potential risks. By proactively addressing risks, we aim to ensure the successful delivery of the project within scope, time, and budget constraints. This plan serves as a guide to identify, analyze, and respond to risks that may impact the project's progress.

2.2 Risk identification & Assessment

Risk	Probability	Impact	Assessment	Strategy
Frequent changes in requirements	Moderate	High	Moderate	Mitigate
Lack of expertise in specific technologies	Moderate	Moderate	Moderate	Mitigate
Risk of scope creep in design	High	Moderate	High	Mitigate
Team working in silos	High	Moderate	High	Mitigate
Lack of communication within the team	Moderate	Moderate	Moderate	Mitigate
Health issues within the team	Low	High	Moderate	Accept
Team disagreement on roles and functions	Low	Moderate	Low	Mitigate
Inadequate problem-solving skills	Low	Moderate	Low	Mitigate
Difficulty meeting project deadlines	Moderate	Moderate	Moderate	Mitigate
Inefficient learning process leading to slow progress	Low	Moderate	Low	Mitigate
Challenges in coordinating weekly team meetings	Moderate	Low	Low	Mitigate
Protracted learning phase in early sprints	Moderate	Moderate	Low	Mitigate
Non collaborative team member	Moderate	High	Moderate	Transfer

Table 3. Risk Identification and assessment

2.3 Risk Mitigation Plan

1. Frequent changes in requirements from PO (Mitigate):

- Establish a change review and approval process.
- Document and communicate changes to the entire team.
- Reevaluate the schedule and affected resources if necessary.

2. Lack of expertise in specific technologies (Mitigate):

- Train the team in the necessary technologies or find alternative technologies.
- Partner with members/ mentors from other teams that can help with sharing knowledge

3. Risk of scope creep in design or data (Mitigate):

- Ensure close collaboration between designers and developers.
- Review and prioritize essential features if the designs exceeds development capacity.

4. Team working in silos (Mitigate):

- Organize regular team-building meetings.
- Foster collaboration and information sharing in team meetings and communication channels.

5. Lack of communication within the team (Mitigate):

- Set communication norms and hold regular meetings.
- Create a daily stand up channel so that we can share daily

6. Health issues within the team (Accept/transfer):

- Reassign tasks or seek temporary replacements if a team member falls ill.
- If the issue lasts longer than a week , ask deploy (impact) managers for support

7. Team disagreement on roles and functions (Mitigate):

- Clearly define roles and responsibilities from the start.
- Hold a team meeting to clarify roles and functions in case of disagreement.

8. Inadequate problem-solving skills (Mitigate):

- Provide training and resources to improve problem-solving skills.
- Encourage team members to collaborate on problem-solving.

9. Difficulty meeting project deadlines (Mitigate):

- Carefully plan and allocate resources.
- Monitor progress regularly and adjust schedules as needed.

10. Inefficient learning process leading to slow progress (Mitigate):

- Identify areas of learning that can be expedited.
- Allocate more time and resources to challenging learning areas.



11. Challenges in coordinating weekly team meetings (Mitigate):

- Use technology to facilitate virtual meetings.
- Record and share meeting summaries for absent team members.

12. Protracted learning phase in early sprints (Mitigate):

- Prioritise essential learning and development tasks.
- Adjust sprint planning to accommodate learning requirements.

13. Non-collaborative team member (Transfer):

- Escalate the issue to deployment impact managers.
- Seek a replacement or alternative team member if necessary.

3. Post-MVP features

In this document, we will outline the additional features of our project that will be delivered as Figma prototypes due to time constraints. These features are planned for post-MVP development.

3.1 Notifications for Both Users

Feature Description:

Notifications for both users will provide a menu displaying notifications whenever there are activities related to the Economists KPIs

Purpose:

To enhance user engagement and keep them informed of important updates.

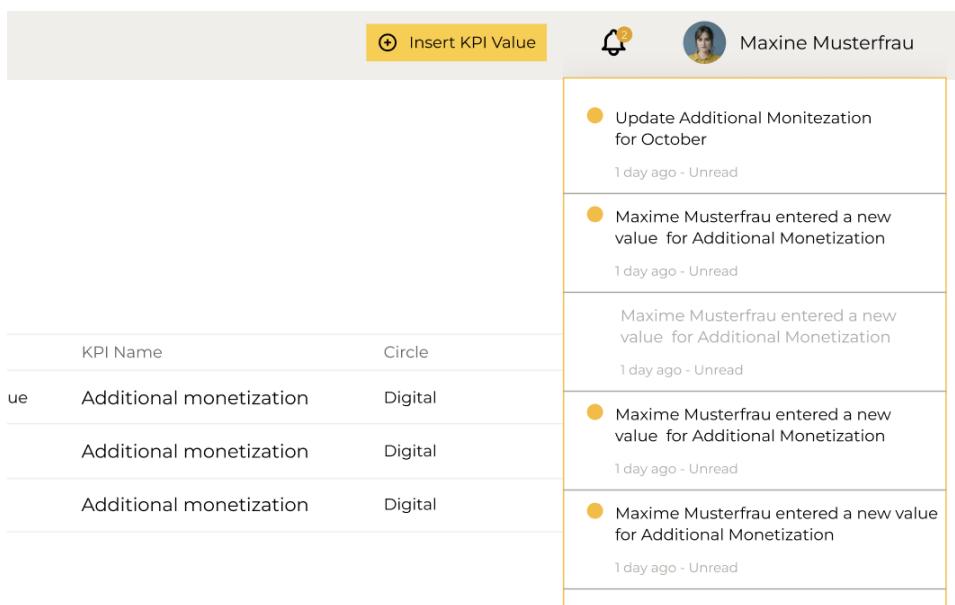


Fig 1. Figma screenshot with notification feature

How It Works:

Whenever one of the Economist's circles has new KPI added, modified, deleted or requires user action, the user will be notified. This also includes reminders of KPI input for the current period, be it monthly, quarterly or annual.



3.2 Responsive Screens

Feature Description:

Responsive screens for mobile and desktop will ensure the application adapts to different screen sizes.

Purpose:

To improve user experience on various devices.

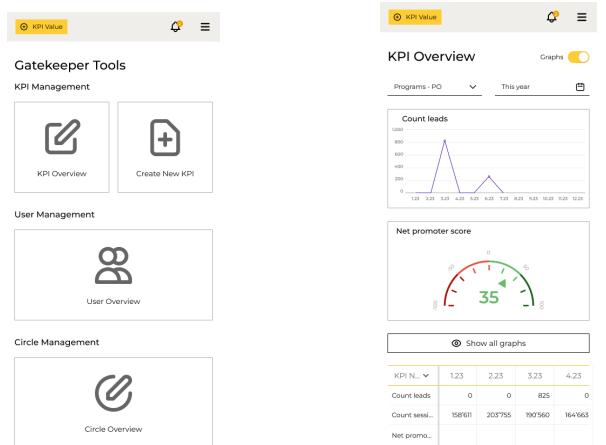


Fig 2. Figma screenshot with responsive screens to different formats

3.3 Data Export Functionality

Feature Description:

Users will have the ability to export data in PDF, CSV, and Excel formats.

Purpose:

To provide users with flexibility in accessing and analyzing their data.

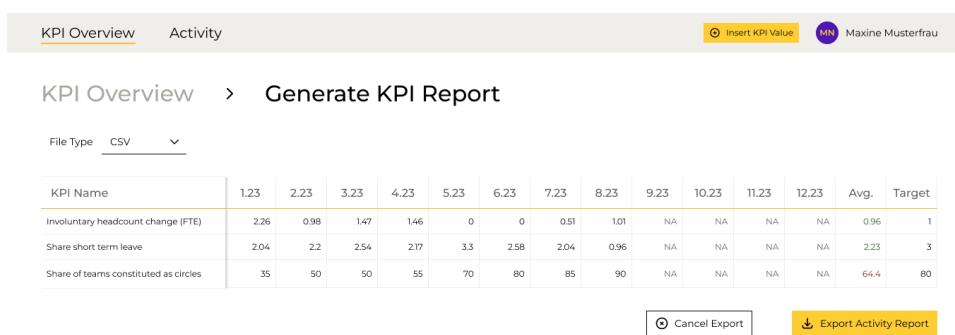


Fig 3. Figma screenshot with data export functionality

3.4. Cumulative KPI Option In KPI Creation Menu

Feature Description:

Cumulative KPI is an option in the KPI creation menu, available once the unit % is selected. Value cumulation has already been implemented in the visualizations of the current MVP, however the option must be added to the KPI creation menu so that per default cumulative values can have their own unit type (% cumulative).

Unlike *amount* and CHF based KPIs, which are always cumulative, percentage based KPIs can be cumulative or incremental. Score based KPIs are incremental.

For example: 'Share of Teams Constituted as Circles' will be cumulative per default as it grows regularly accumulating towards a target, while 'Involuntary Headcount Change' will be considered as an average over time. In this case 'Share of Teams Constituted as Circles' would be a KPI where the option *cumulative* must be selected

Purpose:

To provide users with a more the possibility of creating percentage based KPIs that are cumulative or not. For more details, please refer to the Data Science documentation on section 4.1.4 Definition of Unit Types.

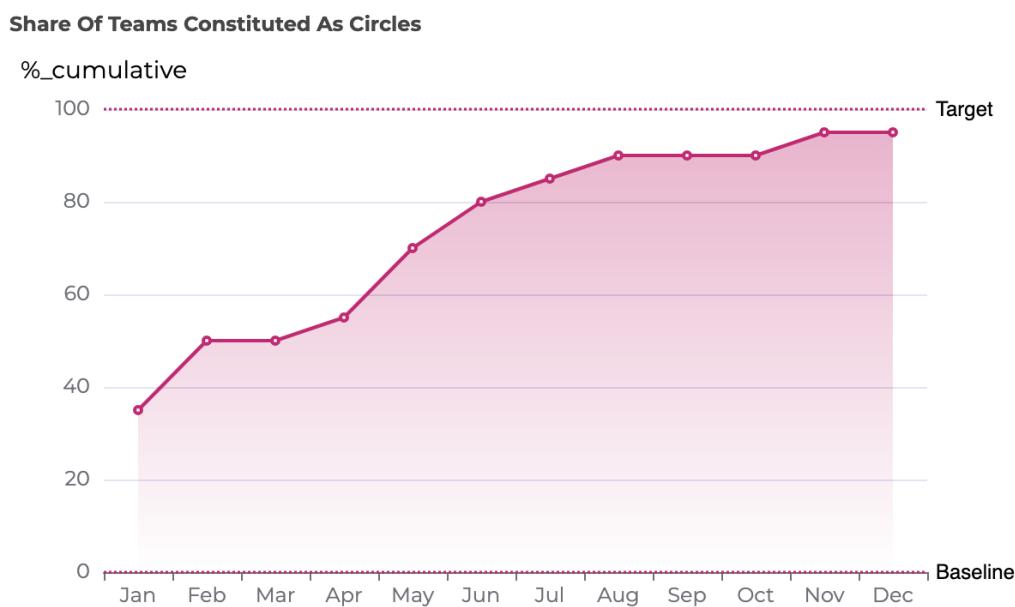


Fig 4. Line plot with per default cumulative KPI (input data is already cumulated).

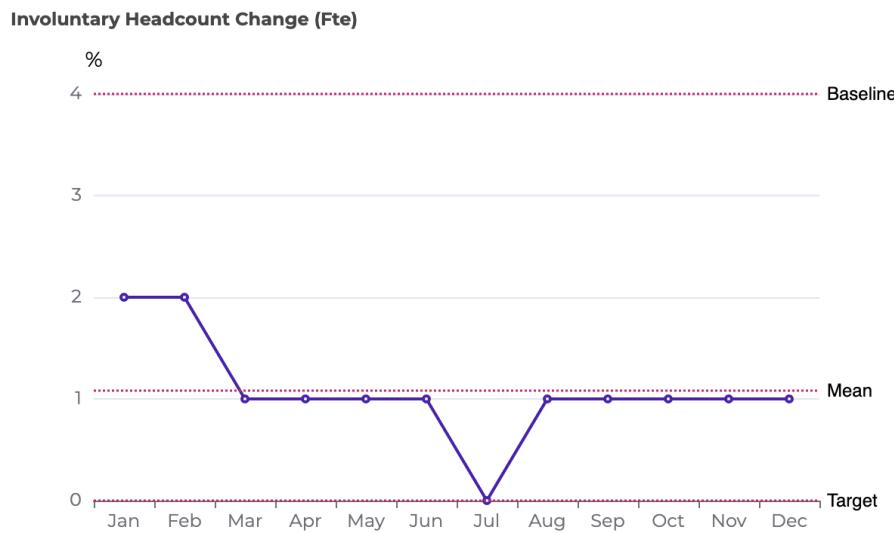


Fig 5. Line plot with incremental KPI. In this case, the mean is provided as reference line.

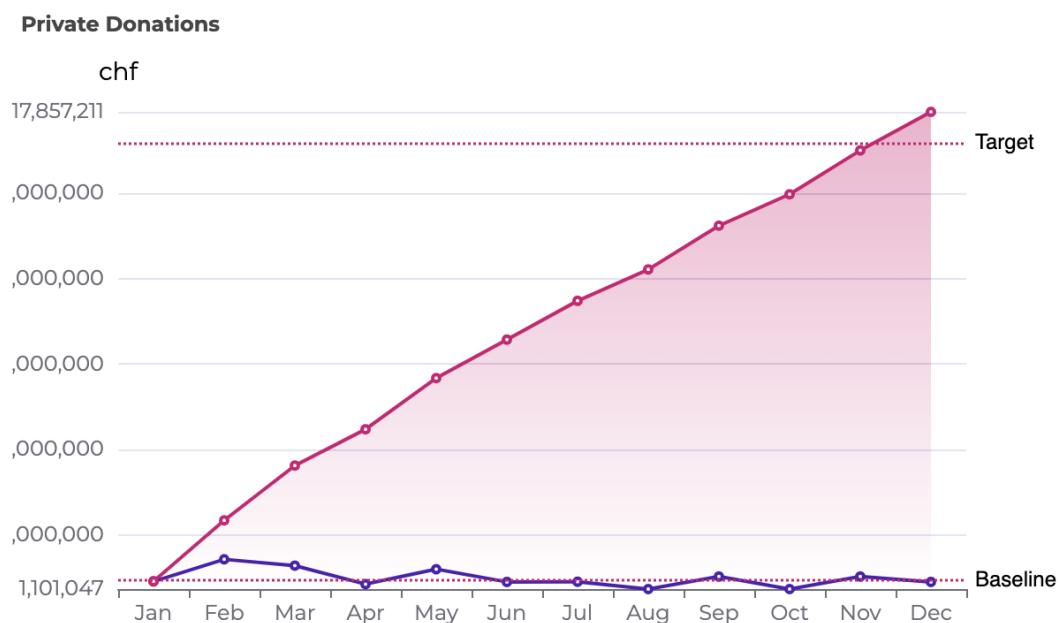


Fig 6. Line plot with CHF or amount units: input data is incremental, but will be cumulated a posteriori. In this case, both the original and the cumulated data points are plotted.

4. UX/UI Documentation

4.1 Tools

Miro board was used for the User Research and Figma for the Screen design and Prototyping.

4.2 User Research

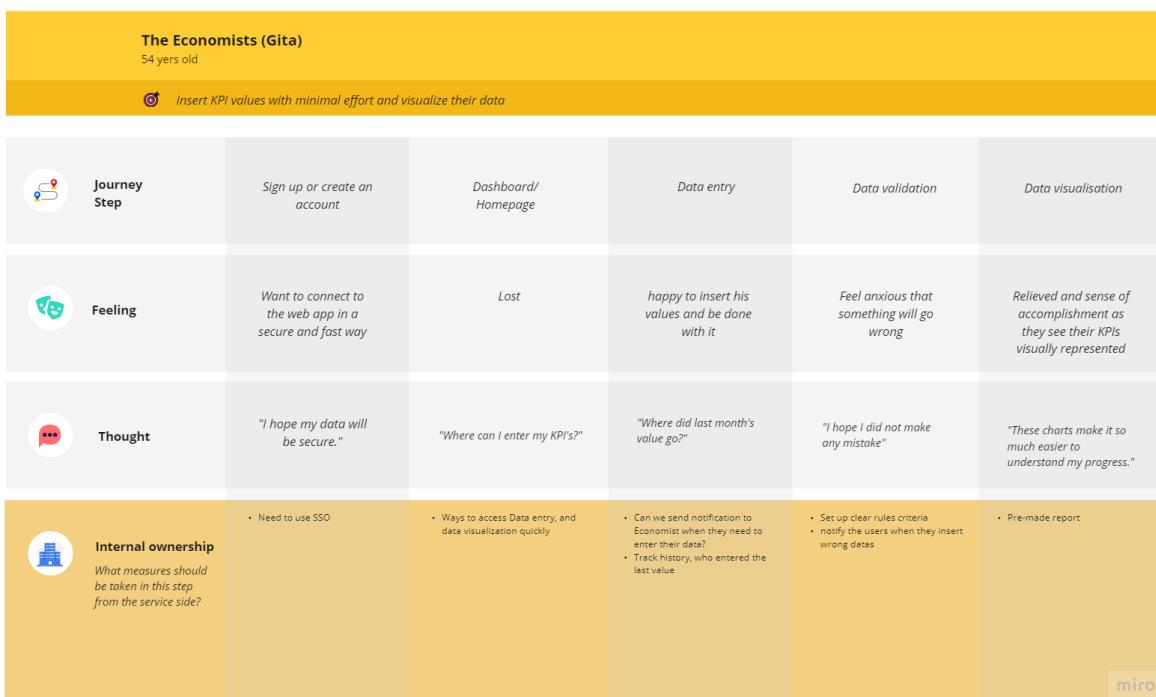
4.2.1. Persona Definition and Problem Space

Personas	
 Gita Purpose : enable consistent tracking of defined KPIs Tasks : frequently input values for previously defined KPIs Pains : confusion when trying to enter what needs to be entered and how to do it correctly (what's new/good) Needs : enter my value and be done with it	 Heiga Purpose : collaboratively define and minimize data maintenance Tasks : define KPIs with technical department + maintain app Pains : maintenance effort Needs : data quality

Part of the client briefing was a well defined problem space, divided into two different personas, with different needs. The economists, who requires a simplified way to input data and the gatekeeper, who defines the parameters of the data the economist has to input.

4.2.2. User Journey Mapping

To empathize better with our users, we mapped their respective journeys:

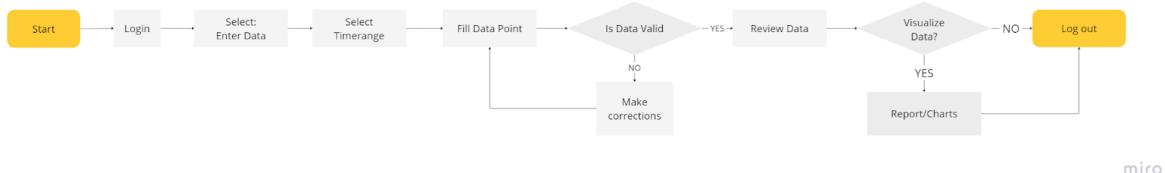


The Gatekeeper (Helga) 45 years old					
🎯 Identify new KPIs and set up data points in the database for the Economists to fill in					
Journey Step	identify new KPI	Sign up or create an account	Dashboard/Homepage	Add new data point	Data validation
Feeling	Excitement	Nervous	Lost	Overwhelmed	Relieved
Thought	"this could help our overview"	"I hope my connection is safe"	"There's a lot of data to manage"	"I hope this fields work for the economists"	"I'm glad it works as it should"
Internal ownership	What measures should be taken in this step from the service side?	<ul style="list-style-type: none"> Need to use SSO Overview of the general circle's data input Tool only accessible to gatekeeper to create and manage data points 	<ul style="list-style-type: none"> Determine the data necessary to calculate the KPI <ul style="list-style-type: none"> Primary data required Secondary data, if any Define the calculation method for the KPI Manual data entry 	<ul style="list-style-type: none"> Launch the new KPI data point Monitor the data point for any issues Regularly check the data for accuracy Adjust the KPI as necessary based on feedback and performance 	miro

4.2.3. User Flows

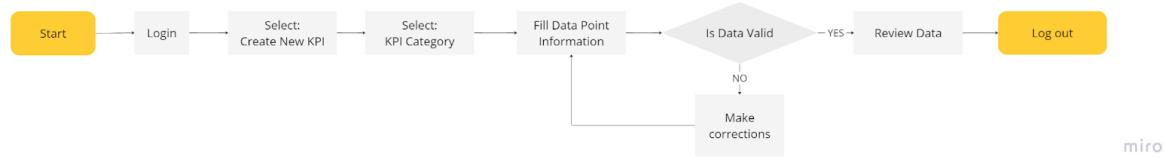
As a way to prioritize the screens to be designed, we created a user flow for each of our personas.

Economist flow:



miro

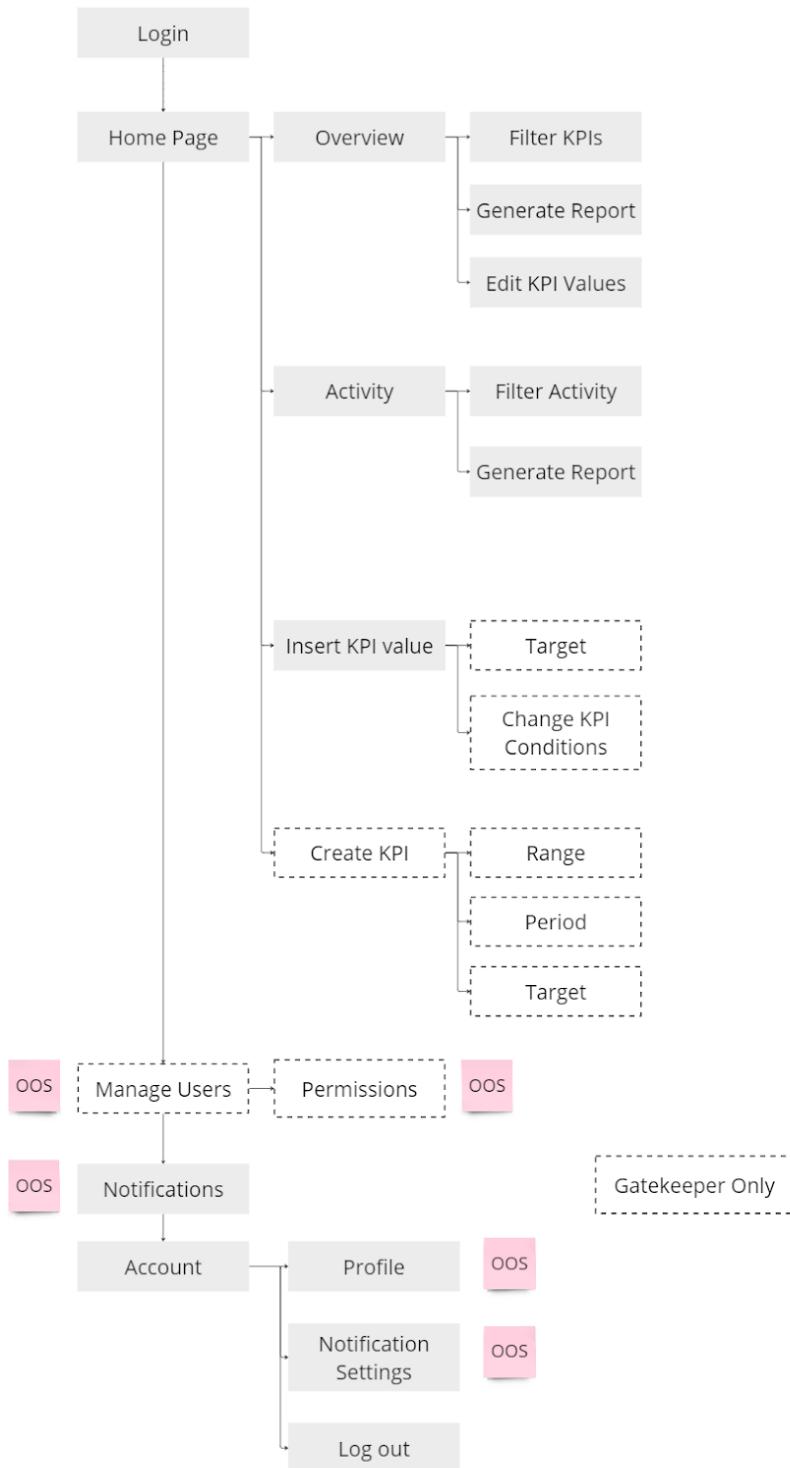
Gatekeeper flow:



miro

4.2.4. Information Architecture

The next step, is to visualise the Information Architecture to have a set structure of the tool to be designed. We have also marked how this product could be expanded in teh future with Out Of Scope features.



4.3 User Experience Design

4.3.1. Design System

The Design system values have been chosen based on the pro Juventute's CI/CD, to have a established familiarity with the tool and generate a sense of ownership in the users.

4.3.1.1. Colors

The primary colors come directly from the Pro Juventute'S CI/CD:

Sunglow	Gray	Green	Red
900 #E5A200	900 #656565	Shade 1 #1C6420	Shade 1 #AC1400
600 #FBBB21	600 #ABA8A3	Shade 2 #66BB6A	Shade 2 #E7523D
300 #FECC33	300 #F0EEEB	Shade 3 #BFEAC0	Shade 3 #FFB1A7
50 #FEF9E9	50 #F9F9F9		

The secondary palette was expanded to be used in the visualizations:

Olive	Cyan	Blue	Purple	Magenta	Terracotta
Shade 1 #5B6A00	Shade 1 #08C2DB	Shade 1 #072490	Shade 1 #4E17B0	Shade 1 #D00071	Shade 1 #8C5009
Shade 2 #B5D00D	Shade 2 #93E8F4	Shade 2 #6686FF	Shade 2 #833FFF	Shade 2 #FF79C1	Shade 2 #F59019
Shade 3 #E8F49A	Shade 3 #C5F7FD	Shade 3 #B2C3FF	Shade 3 #CAACFF	Shade 3 #FFD0E9	Shade 3 #FFCB8D



4.3.1.2. Typography

Pro Juventute uses Montserrat, a Google font, on their website and marketing materials. This font is very user friendly and optimized for web. For this reason, it has also been kept for the development of the KPI Dashboard.

text-sm
Font Size: 12px | Line Height: 16px | Font Weight: Regular | Font Decoration: None

text-base
Font Size: 14px | Line Height: 20px | Font Weight: Regular | Font Decoration: None

text-lg
Font Size: 16px | Line Height: 24px | Font Weight: Regular | Font Decoration: None

text-xl
Font Size: 18px | Line Height: 26px | Font Weight: Regular | Font Decoration: None

text-2xl
Font Size: 20px | Line Height: 28px | Font Weight: Regular | Font Decoration: None

text-3xl
Font Size: 24px | Line Height: 32px | Font Weight: Regular | Font Decoration: None

text-4xl
Font Size: 32px | Line Height: 36px | Font Weight: Regular | Font Decoration: None

text-5xl
Font Size: 40px | Line Height: 40px | Font Weight: Regular | Font Decoration: None

text-6xl
Font Size: 48px | Line Height: 48px | Font Weight: Regular | Font Decoration: None

text-7xl
Font Size: 60px | Line Height: 60px | Font Weight: Regular | Font Decoration: None

text-sm
Font Size: 12px | Line Height: 16px | Font Weight: Medium | Font Decoration: None

text-base
Font Size: 14px | Line Height: 20px | Font Weight: Medium | Font Decoration: None

text-lg
Font Size: 16px | Line Height: 24px | Font Weight: Medium | Font Decoration: None

text-xl
Font Size: 18px | Line Height: 26px | Font Weight: Medium | Font Decoration: None

text-2xl
Font Size: 20px | Line Height: 28px | Font Weight: Medium | Font Decoration: None

text-3xl
Font Size: 24px | Line Height: 32px | Font Weight: Medium | Font Decoration: None

text-4xl
Font Size: 32px | Line Height: 36px | Font Weight: Medium | Font Decoration: None

text-5xl
Font Size: 40px | Line Height: 40px | Font Weight: Medium | Font Decoration: None

text-6xl
Font Size: 48px | Line Height: 48px | Font Weight: Medium | Font Decoration: None

text-7xl
Font Size: 60px | Line Height: 60px | Font Weight: Medium | Font Decoration: None

text-sm
Font Size: 12px | Line Height: 16px | Font Weight: SemiBold | Font Decoration: None

text-base
Font Size: 14px | Line Height: 20px | Font Weight: SemiBold | Font Decoration: None

text-lg
Font Size: 16px | Line Height: 24px | Font Weight: SemiBold | Font Decoration: None

text-xl
Font Size: 18px | Line Height: 26px | Font Weight: SemiBold | Font Decoration: None

text-2xl
Font Size: 20px | Line Height: 28px | Font Weight: SemiBold | Font Decoration: None

text-3xl
Font Size: 24px | Line Height: 32px | Font Weight: SemiBold | Font Decoration: None

text-4xl
Font Size: 32px | Line Height: 36px | Font Weight: SemiBold | Font Decoration: None

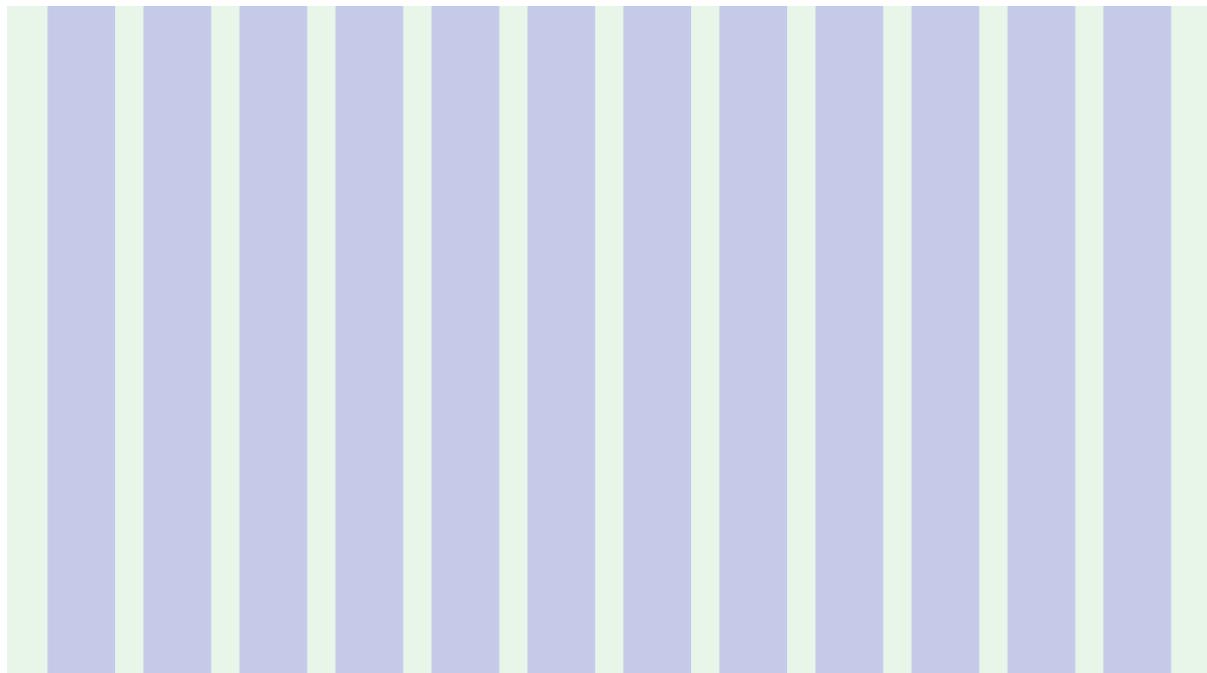
text-5xl
Font Size: 40px | Line Height: 40px | Font Weight: SemiBold | Font Decoration: None

text-6xl
Font Size: 48px | Line Height: 48px | Font Weight: SemiBold | Font Decoration: None

text-7xl
Font Size: 60px | Line Height: 60px | Font Weight: SemiBold | Font Decoration: None

4.3.1.3. Grid

Following industry's best practices, a 12 column grid was the based of the composition of all of our screens:



4.3.1.4. Icons

The already existing set of icons from [heroicons.com](#) were chosen for this project, as they cover most of the needs. This set was expanded with more custom icons as the project needs exceeded the basic offering of this collection.

Outline 2px stroke weight, 24×24 bounding box

For primary navigation and marketing sections, designed to be rendered at 24x24.





4.3.1.5. Elements and Components

With the purpose to streamline the design process and assure consistency through the screens, we created elements and components we would constantly reuse along the project.

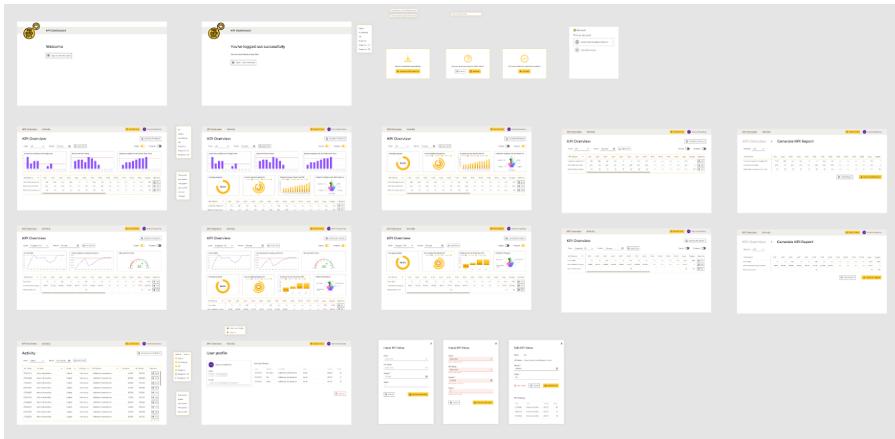
The table component proved to be the most challenging, as this was going to be the main source of information for the user. The yellow hue from Pro Juventute stands out, as this helped create a friendly user interface that's also familiar to the members of the organization. Purple is the main contrast color. And red for error states.

The image displays a grid of 25 wireframes and UI snippets, each enclosed in a dashed purple border. The components include:

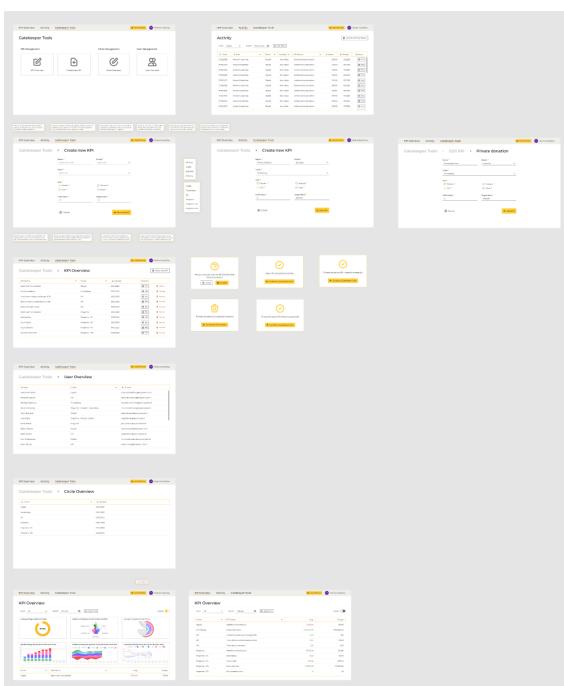
- Top row:
 - Logo: PRO JUVEN TUTE
 - Avatar grid: 2x3 grid of user icons.
 - Text input: Input field with placeholder "Maxine Musterfrau added the value for additional monetization Q4 23 on the 27.09.2023 at 13:30".
 - Bar chart: A bar chart showing values for Digital, Fundraising, HR, Programs, Programs - PO, and Programs - CC.
- Second row:
 - Buttons: A 3x3 grid of yellow buttons labeled "Button".
 - Buttons: A 3x3 grid of orange buttons labeled "Button".
 - Buttons: A 3x3 grid of grey buttons labeled "Button".
 - Buttons: A 3x3 grid of white buttons labeled "Button".
 - Buttons: A 3x3 grid of light blue buttons labeled "Button".
 - Icons: A 3x3 grid of yellow icons labeled "Icon".
 - Icons: A 3x3 grid of orange icons labeled "Icon".
 - Icons: A 3x3 grid of grey icons labeled "Icon".
 - Icon: A single yellow icon labeled "Digital".
 - Icon: A single orange icon labeled "Digital".
 - Icon: A single grey icon labeled "Digital".
- Third row:
 - Labels: A vertical stack of labels with error messages: "Label", "Label", "Label", "Label", "Label", "Label", "Label", "Label", "Label".
 - Switches: A vertical stack of switches.
 - Checkboxes: A vertical stack of checkboxes.
 - Form fields: A horizontal form with fields for "General", "Time", and "Target".
 - Buttons: Three buttons labeled "Edit Current KPI".
- Fourth row:
 - Text inputs: Three text input fields with placeholder "Text".
 - Select dropdowns: Three select dropdowns with placeholder "Select".
 - Date inputs: Three date input fields with placeholder "mm/yyyy".
 - Date inputs: Three date input fields with placeholder "mm/yyyy".
 - Date inputs: Three date input fields with placeholder "mm/yyyy".
- Fifth row:
 - Form fields: A vertical stack of labels, selects, and error messages for "Text" and "Select" fields.
 - Form fields: A vertical stack of labels, selects, and error messages for "Text" and "Select" fields.
 - Form fields: A vertical stack of labels, selects, and error messages for "Text" and "Select" fields.
- Sixth row:
 - Form fields: A vertical stack of labels, selects, and error messages for "Text" and "Select" fields.
 - Form fields: A vertical stack of labels, selects, and error messages for "Text" and "Select" fields.
 - Form fields: A vertical stack of labels, selects, and error messages for "Text" and "Select" fields.
- Bottom row:
 - Table header: KPI Type, Value, Period, Options.
 - Table row: Share of teams constituted as... 80% 09.2023 Yes.
 - Table row: Date, User, Circle, Activity, KPI Type, Value, Period, Options.
 - Table row: 27.09.2023 Maxine Musterfrau Digital New Value Additional monetization 26'000 09.2023
 - Table row: KPI Type 123 2.23 3.23 4.23 5.23 6.23 7.23 8.23 9.23 10.23 11.23 12.23 Avg Target Options.
 - Table row: Share of teams constituted... 30% 50% 50% 50% 70% 80% 88% 90% N/A N/A N/A N/A N/A 64.4% 80% Edit.

4.3.2. Screen Design

For the economists, we created the screens to provide a complete flow for their main need, which is to preview and input their KPIs.



The gatekeeper's needs are completely different. They have an overview of the organization, and can define the characteristics of new KPIs for the economist to input. All of this is reflected on their screens



4.3.3. Prototyping

To test our digital product and facilitate an overview of its functionality, we've created a prototype for the flow of each persona:

- [Economist logs in and adds a new value to an existing KPI](#)
- [Gatekeeper creates a new KPI and reviews current KPIs, Users and Circles](#)

5. Data Science Documentation

5.1 Methodology

5.1.1 Exploratory Data Analysis

A thorough Exploratory Data Analysis was developed in the first place in order to assess viable features to be proposed. The EDA allowed us to assess that:

- The relationships between Circles, KPIs and ranges were at the moment linear, as no KPI was shared between circles (Fig. 7). In the same manner, multiple units, ranges and periodicities were identified and integrated in the KPI creation menu with possible new incoming ones.

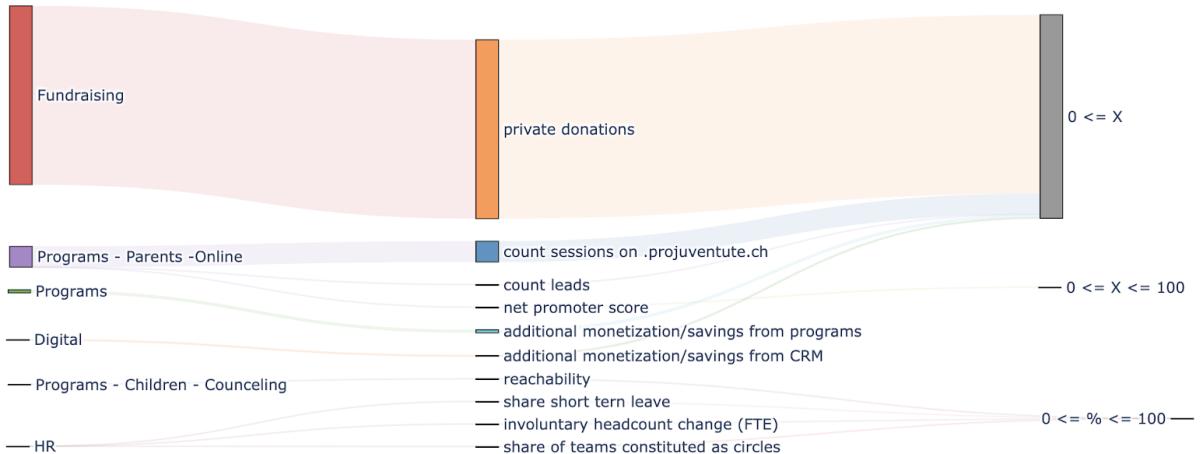


Fig 7. Sankey diagram of hierarchical relationships between Circles, KPIs and ranges.

- An objective metrics of success was required, as each KPI has its own units and ranges, not allowing for a fair understanding of progress towards a goal.
- The definition of a baseline and a target for each KPI was required. Equally, positive and negative trends were identified, meaning that some KPIs are considered as successful if their numbers drop. i.e. Involuntary head count change.
- A forecast feature was considered, but soon discarded after discussion with the PO.

As a next step, given the recent adoption of KPI tracking within ProJuventute, we were faced with lack of data, we could test our features with. We overcame this obstacle by producing a toy dataset generated from the provided one, for part of 2023, and improved



further, with the typical characteristics of a time series data. The pipeline to create the toy dataset was the following:

5.1.2 Toy Dataset Generation

a. First Synthetic dataset using CTGAN:

CTGAN is a collection of Deep Learning based synthetic data generators for single table data, which are able to learn from real data and generate synthetic data with high fidelity. Using the 9 months available for 2023, we proceeded with synthetically generating 10 years of data. Initially, the missing months of 2023 were interpolated in order to have a complete year. On a second stage, 9 additional previous years, from January 2014 on, were produced with CTGAN.

Two exceptions in terms of data generation were considered:

- The KPI “Share of teams constituted as circle” was only considered for 2023, as the goal is time-bound to only one year.
- The KPI “Private Donations” was integrated from the actual dataset provided by the PO.

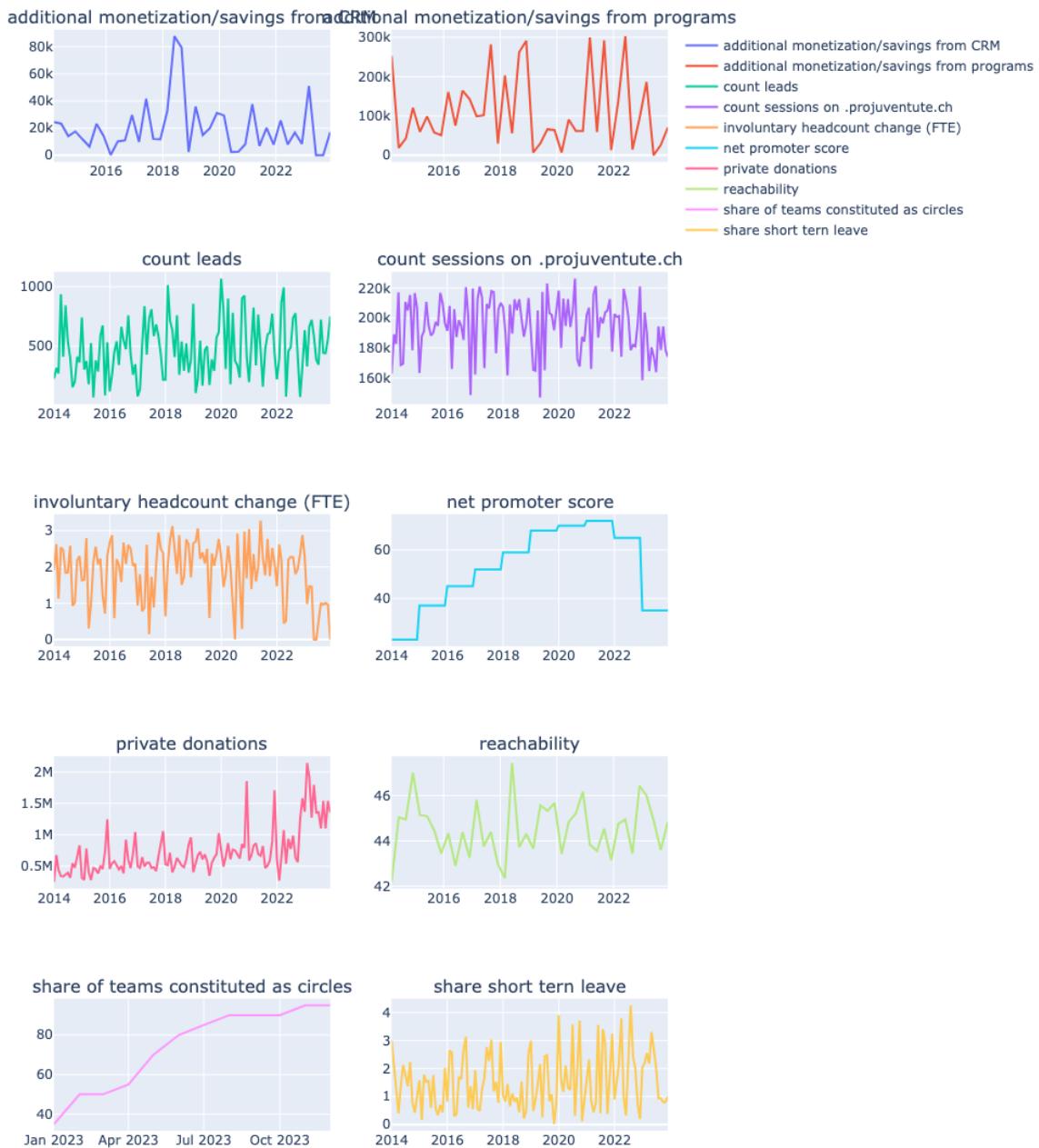


Fig 8. Line plot of synthetic dataset for all KPIs over 10 years.

b. Descriptive statistics of actual and synthetic datasets:

The statistical properties of both datasets were compared to ensure they were preserved in the generation. The delta between the two was evaluated to remain low, in relation to the ranges of the given KPI.

	share short term leave	involuntary headcount change (FTE)	reachability	count sessions on .projoventute.ch	count leads	additional monetization/savings from CRM	additional monetization/savings from programs
count	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000
mean	1.505926	1.978148	44.273426	196304.943241	473.924352	19818.215463	115701.212685
std	1.014453	0.746665	1.192060	18215.342345	246.926143	18622.909995	93065.043460
min	0.010000	0.010000	41.950000	147095.300000	68.250000	228.560000	973.970000
25%	0.687500	1.622500	43.402500	186611.980000	284.692500	7664.455000	46050.892500
50%	1.360000	2.095000	44.355000	200658.660000	437.705000	14379.220000	82655.375000
75%	2.157500	2.545000	45.092500	210691.322500	640.727500	24588.127500	170256.810000
max	4.260000	3.270000	47.440000	226323.750000	1068.920000	87600.480000	346565.140000

	share short term leave	involuntary headcount change (FTE)	reachability	count sessions on .projoventute.ch	count leads	additional monetization/savings from CRM	additional monetization/savings from programs
count	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000
mean	1.777500	0.885000	44.832500	179728.563333	575.916667	17000.000000	70752.667500
std	0.850979	0.681262	0.887285	13847.495742	157.015609	21746.473068	74498.730391
min	0.800000	0.000000	43.600000	158611.000000	337.000000	0.000000	0.000000
25%	0.930000	0.382500	44.522500	171746.000000	467.000000	0.000000	195000.000000
50%	2.040000	0.970000	44.865000	178822.880000	576.500000	8500.000000	48376.335000
75%	2.285000	1.122500	45.175000	191521.750000	701.000000	25500.000000	99629.002500
max	3.300000	2.260000	46.000000	203755.000000	771.000000	51000.000000	186258.000000

	share short term leave	involuntary headcount change (FTE)	reachability	count sessions on .projoventute.ch	count leads	additional monetization/savings from CRM	additional monetization/savings from programs
count	96.000000	96.000000	96.000000	96.000000	96.000000	96.000000	96.000000
mean	-0.271574	1.093148	-0.559074	16576.379907	-101.992315	2818.215463	44948.545185
std	0.163473	0.065403	0.304775	4367.846603	89.910534	-3123.563073	18566.313069
min	-0.790000	0.010000	-1.650000	-11515.700000	-268.750000	228.560000	973.970000
25%	-0.242500	1.240000	-1.120000	14865.980000	-182.307500	7664.455000	26550.892500
50%	-0.680000	1.125000	-0.510000	21835.780000	-138.795000	5879.220000	34279.040000
75%	-0.127500	1.422500	-0.082500	19169.572500	-60.272500	-91.872500	70627.807500
max	0.960000	1.010000	1.440000	22568.750000	297.920000	36600.480000	160307.140000

Fig 9. Statistical description of actual dataset, synthetic dataset and their deltas.

c. Integration of Time Series Characteristics

Intrinsic characteristics of time series data were normalised (0 to 1) and integrated as a product of the generated dataset. The following characteristics were numerically included:

- **Seasonality:** referring to periodic fluctuations in a time series that occur at regular intervals, such as daily, monthly, or yearly.
- **Trend:** Representing the underlying pattern or trajectory of a time series over a long period. This can be upward, downward, or stable, indicating the general direction in which the data points are moving.
- **Noise:** Random variations in a time series from unpredictable factors
- **Non-stationary events:** These are unexpected disruptions that can drastically alter the pattern of a time series. For instance, the Covid-19 pandemic was a non-stationary event that affected various time series data globally, causing significant deviations from their expected patterns.
- Finally, to better highlight the underlying patterns, the time series was smoothed using a moving average technique. This involved calculating the average of data points in a moving window of 5, which helps in smoothing out short-term fluctuations and emphasizing long-term trends.

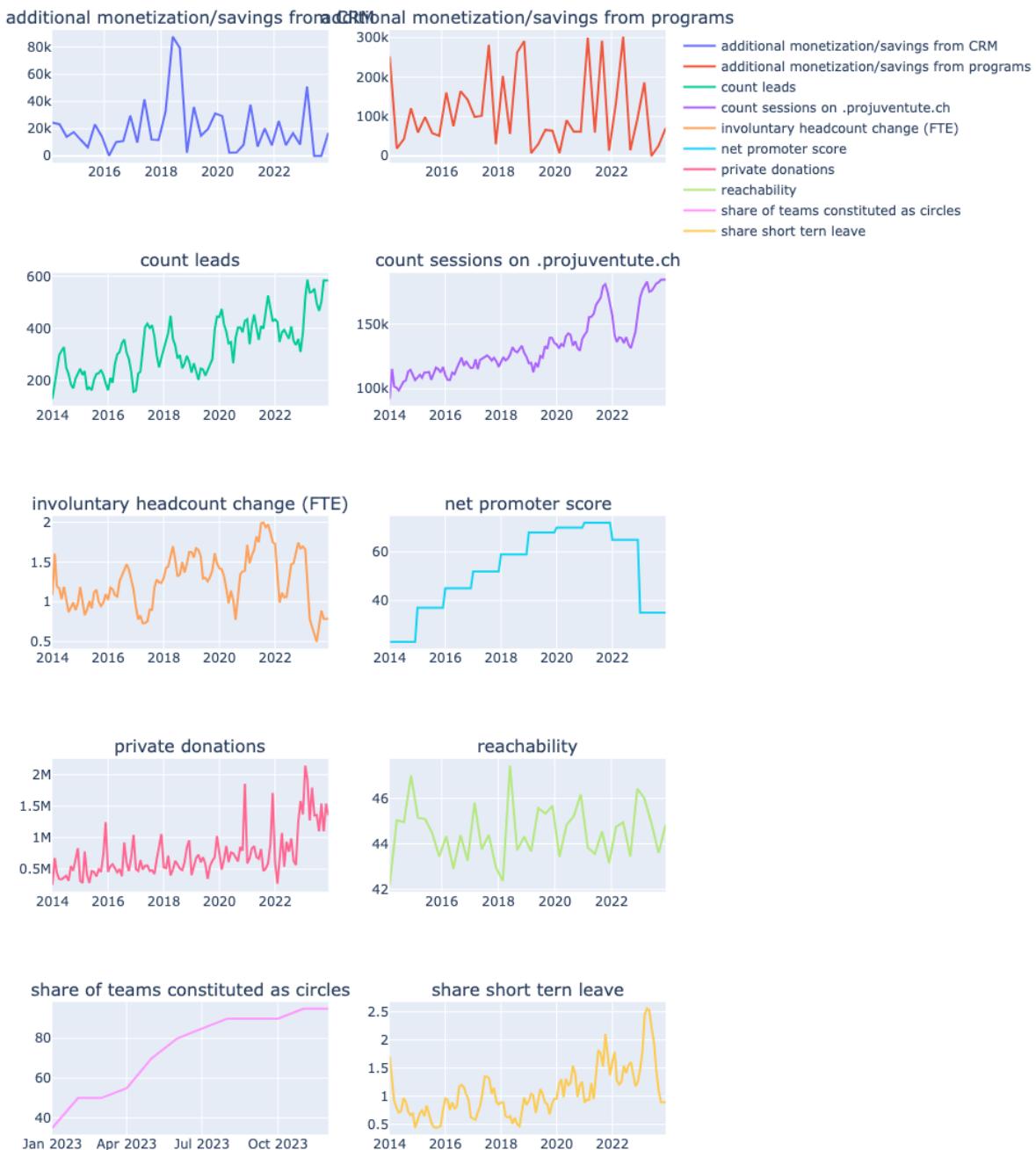


Fig 10. Final time series synthetic dataset for all KPIs.

5.1.3 Calculation of Initial Values and Target Values

In addition to the generated values, Initial Values (baseline) and Target Values (target) were produced for each data point. Based on the KPIs' unit, values and description, the following was inferred or corrected:

- Net Promoter Score ranges between -100 and 100.
- Share Short Term Leave, and Involuntary Headcount Change have negative trends and range between 3.5% and 0%, considering the target is to be as close as possible to 0% and 3.5% as the worst case scenario.
- All other KPIs with unit % range between 0 and 100 with a positive trend.
- For all other KPIs with unit *amount* or CHF, baseline and target were extracted out of the data itself. A random position between the 80th and the 100th percentile of the distribution for a specific year was considered as target. In the same manner, a random position between the 1st and the 20th percentile of the distribution for a specific year was considered as baseline.
- The final characteristics of the dataset were assigned as shown in Fig 11. Dynamic baseline/targets are assigned on a yearly basis.

	kpi	circle	periodicity	unit	initial_value	target_value
0	private donations	Fundraising	month	chf	0	1356027.0
1	involuntary headcount change (FTE)	HR	month	%	3.5	0
2	share short term leave	HR	month	%	3.5	0
3	count leads	Programs - Parents -Online	month	amount	0	474.0
4	count sessions on .projuventute.ch	Programs - Parents -Online	month	amount	0	182361.0
5	net promoter score	Programs - Parents -Online	year	score	-100	100
6	additional monetization/savings from CRM	Digital	quarter	chf	0	31170.0
7	additional monetization/savings from programs	Program	quarter	chf	0	252686.0
8	reachability	Programs - Children - Counseling	quarter	%	0	100
9	share of teams constituted as circles	HR	month	%_cumulative	0	100

Fig 11. KPIs with new created baseline and target values.

5.1.4 Definition of Unit Types

Additional units types were defined considering the current KPIs but also any possible KPI that might be added in the future. The following units were created:

Unit	Description	Cumulation and Range
CHF	Refers to monetary values, financial metrics, or any other data that involves financial transactions or valuations in CHF.	Cumulative (but expected to be input on an incremental basis). Ranges between 0 and X.

%	KPIs using this unit might refer to ratios, proportions, or growth rates. Percentages are commonly used to show changes over time, efficiency rates, or any other relative metric.	Cumulative or Incremental (refer to section 3.4 for cumulative option for %). Ranges between 0 and 100, or inversely if the trend is negative,
Amount	Depending on the context, it could represent counts of tangible items or intangible metrics.	Cumulative (but expected to be input on an incremental basis). Ranges between 0 and X.
Score	KPIs with this unit might refer to performance ratings, customer satisfaction ratings, or any other evaluative measure.	Incremental (non-cumulative). The range is customised depending on the score type.

Table 3. Created units, description, range and cumulative characteristics.

5.2 Required input fields for KPI creation

Based on the previous analysis, the following fields were defined as required for a valid KPI entry, and integrated in the UX. The tooltip definitions are provided below:

- a. **Name:** "Enter the unique name of the KPI to identify it easily on the dashboard. For example, 'Customer Satisfaction', 'Private Donations', etc."
- b. **Circle:** "Select the team responsible for this KPI. This helps in organising and assigning the KPIs to the appropriate teams for monitoring and management."
- c. **Periodicity:** "Choose how often this KPI should be updated or reviewed. Options can range from monthly, quarterly, or annually. This helps in tracking the progress and performance over a specified time frame."
 - o Monthly
 - o Quarterly
 - o Yearly
- d. **Target Measurement:**
 - o **Unit of measurement:** "Specify the unit used to measure this KPI. This provides context to the values and goals set for the performance indicator. Please beware if your KPI has an upward or downward target, i.e. 'Involuntary Head Count Change' is a metric that one would want to reduce"
 - **Percentage:** "Select this option if the KPI is measured in terms of a percentage. For instance, it can be used for metrics like customer

satisfaction rate or task completion rate." This value is automatically set to initial value = 0%, and target value = 100%.

- **CHF:** "Choose this option if the KPI is to be measured in Swiss Francs (CHF). Ideal for financial KPIs related to revenue, expenses, or budgets specifically dealing with this currency." This value cannot be negative.
- **Amount:** "Opt for this unit if the KPI is quantified as a numerical value or count, such as the number of visits to a website or number of volunteers." This value cannot negative.
- **Score:** "Select 'score' if the KPI is measured using a rating or scoring system, such as employee performance rating, product quality score, or customer feedback rating."
- **Initial Value:** "Enter the starting value of the KPI. This is the baseline value from which progress or regression will be measured over time." This value is set to zero if not provided.
- **Target Value:** "Input the desired value you aim to achieve for this KPI within the specified period. This sets a clear goal for the team to work towards." This value is required.

5.3 Metrics of Success: Progress and Performance

5.3.1 From Measure to Metrics

The Merriam-Webster dictionary defines a measure as "an amount or degree of something." Several measures can refer to completely different units, making them incomparable. This was clearly observed in our dataset, with four different units and multiple ranges. In addition, a measurement alone does not give a proper idea of its meaning. What actually does the value "5,000 CHF donation" mean? Is it a high value or low value? We cannot answer this question until we have provided a frame within which this measure can be judged.

On the other hand, "a metric is a standard of measurement" as defined by the Merriam-Webster dictionary. Although similar, the metric defines to a standard measurement, opening the possibility of comparison within a given frame. The metric is a derivative of measure.

As referred in the article [The Difference Between Quantification, Measure, Metric, and KPI](#): **KPI is a metric (it doesn't imply that any metric is a KPI). KPI stands for Key Performance Indicator. These three words define the function of a KPI:**

- **Indicator.** It should show the number. For example: "How good our customers are engaged" is not really an indicator; "Average customer engagement score according to the monthly survey" is an indicator.
- **Performance.** It should be connected to a performance. "The number of computers in an office per employee" is a metric, but it is not really connected to the ultimate business performance. If you double the number of computers, you don't expect your profits to be increased.

- Key. It should be important for your business, department, team or whatever your current scale is.

Taking these considerations into account and the fact that the KPI tracking app purpose is to extend this practice indefinitely over the years to come, our team engaged in proposing two metrics of success: Progress and Performance.

5.3.2 Metrics of Progress and Performance

Rather than one single metric of success, we offer two metrics focused on two different time perspectives. While *Progress* refers to the process of improving overtime in relation to a given KPI time frame, *Performance* gives an idea of how efficiently that progress is achieved in relation to historical data. Both values are represented in percentage.

The nuance between the two lies in the frame used to compare a given measure:

- In the case of *Progress*, the reference frame is created by the **baseline and target** lines defined within the KPI time horizon, as defined in the session 4.1.3. *Progress* ranges between 0 and 100%, the latter being reached once the set target has been attained. *Progress* above 100% is an indication of accomplishment over target.
- In the case of *Performance*, these reference lines are replaced with the **minimum and maximum** values for a given KPI, over historical data.

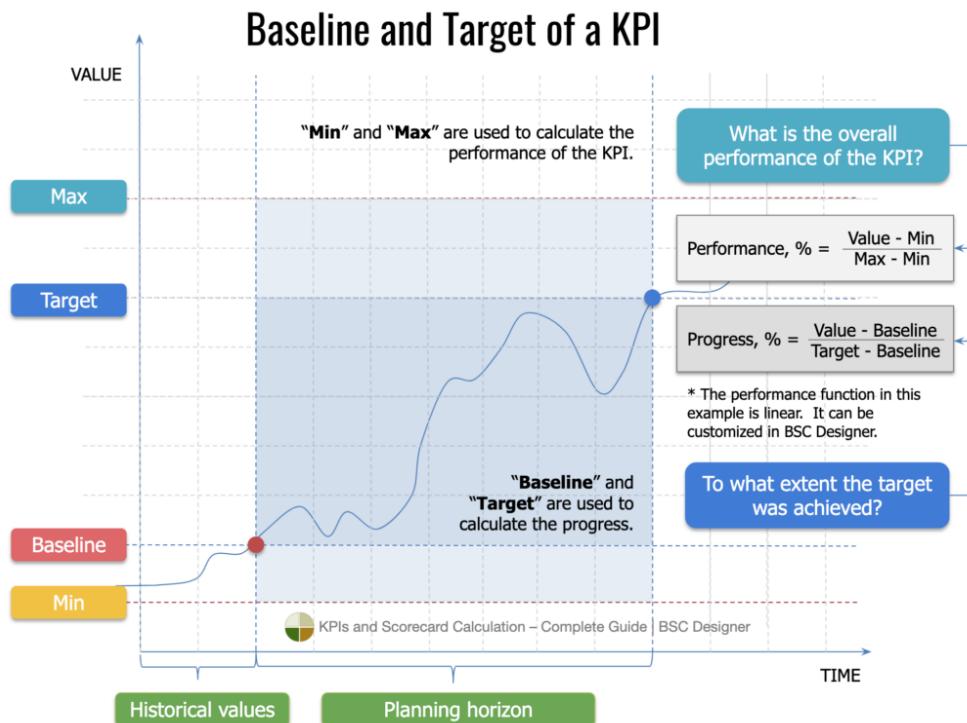


Fig 12. Diagram explaining the specific reference lines used in the calculation of Progress and Performance. Source: <https://bscdesigner.com/calculate-metrics.htm#terms>

Our product offers the metric of *Progress* to the Economist per default, as it is easier to grasp and refers only to the current time frame of the KPI. In addition, the default Gatekeeper screen will have a plot dedicated to the metric of *Performance*, providing her/him with further insights in relation to how efficient is *Progress* being obtained.

5.3.3 Accumulation In Relation To Time

Both *Progress* and *Performance* can be cumulative or incremental. In the context of our product, these metrics inherit the characteristics of their own KPI. i.e. Consider *Progress* and *Performance* of “Private Donations”, both will be cumulative over a given period of time, while the same metrics for “Net Promoter Score” will be averaged over the same time period.

Our product allows any user to select a specific year and month in order to produce the visualisations of the requested KPI's. Per default, the defined time range for the calculation of *Progress* and *Performance* will run **from January of the selected year, until the selected month**. This feature allows having the freedom of creating KPIs without a given timeframe, while providing metrics that work on running year basis, as discussed with the PO.

5.4 Visualisation Rules and Specifications

Our product offers three main kinds of interactive visualisations, all offering extra information on mouse over, which allows decluttering the interface of legends when not strictly needed. All plots are available [here](#), embedded in an HTML file for eased testing.

Based on the kind of the data they display and to whom it is shown, these are the three main families:

- Visualisation of actual data points
- Visualisation of metrics of Progress for a specific circle
- Visualisations of metrics of Progress/Performance for Projuventute (all circles)

5.4.1 Visualisation of actual data points

These visualisations correspond to the default screen offered to the **Economist**. They use as specific colour code **purple and magenta tones**. Moreover, they show not only the actual KPI measure values but also their corresponding target and baseline, giving an immediate idea of the distance to a given goal.

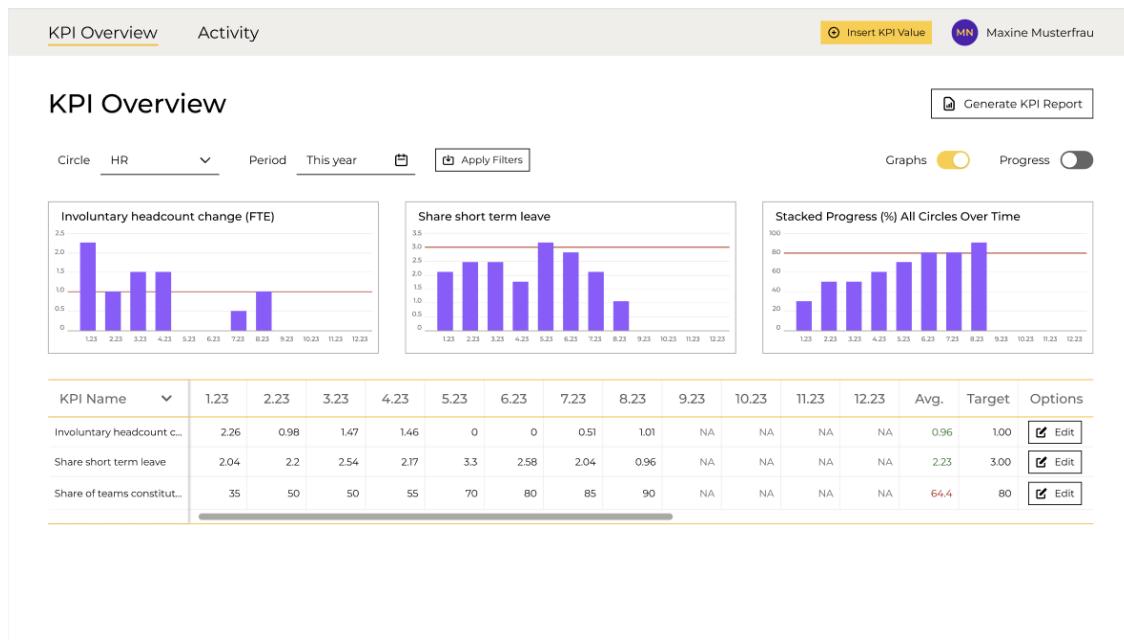
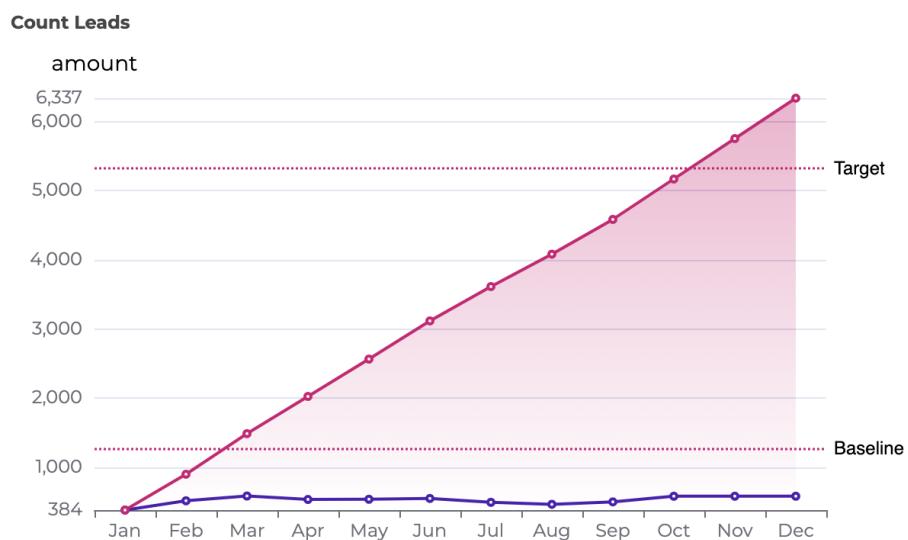


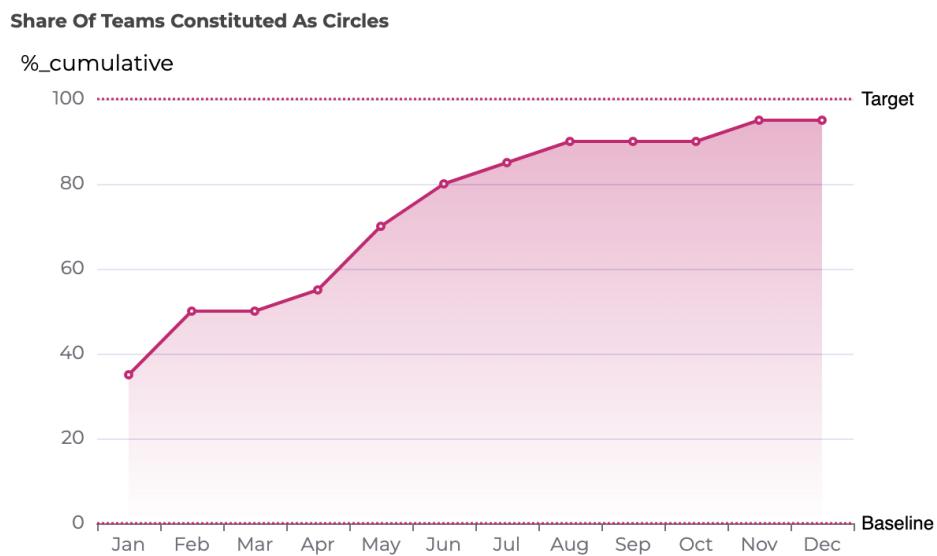
Fig 13. Default Economist Screen.

They are created in a parametrised way, as their type is defined by a set of rules and conditions, so they can accommodate any future KPI:

- **If periodicity is month:**
 - **If the unit is CHF or amount:** line plot with a line for incremental values and a second for cumulative values with a coloured area.



- **If the unit is '%_cumulative':** line plot with a line for cumulative values with a coloured area.

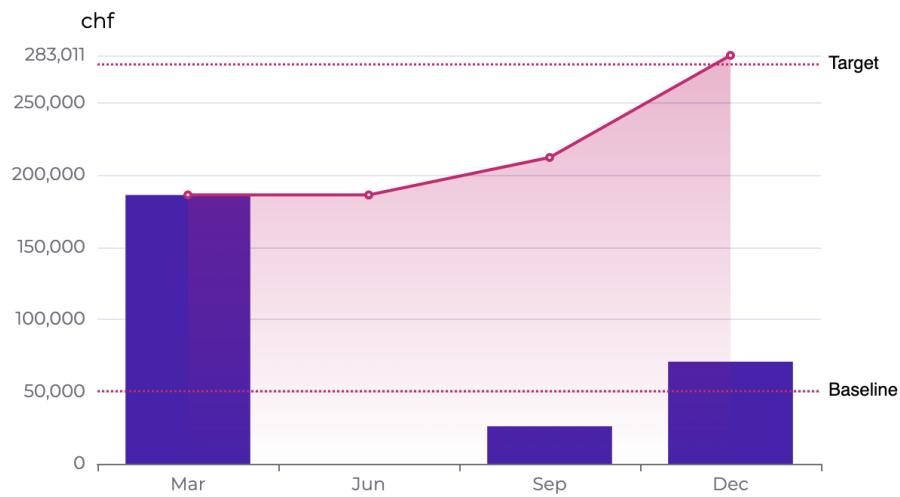


- **If the unit is % or score:** line plot with a line for incremental values and a mean line for reference.



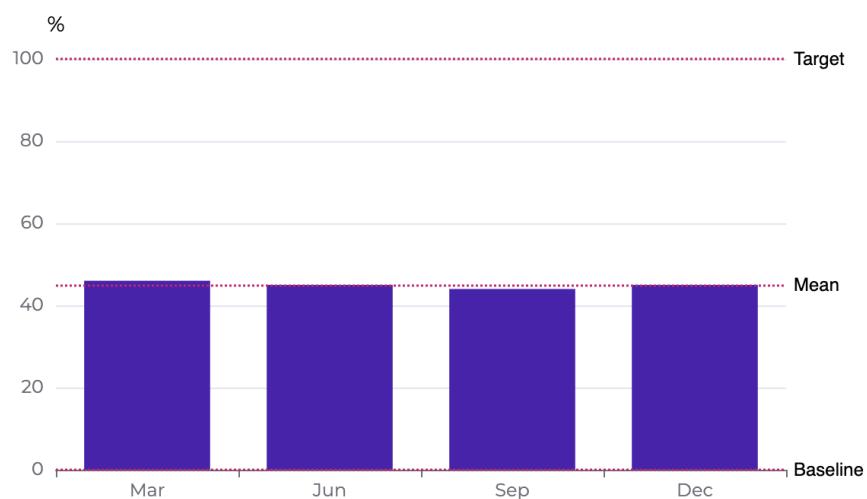
- **If periodicity is quarter:**
 - **If the unit is CHF or amount:** incremental bar plot with a line for cumulative values with a coloured area.

Additional Monetization/Savings From Programs



- **If the unit is '%_cumulative':** cumulative bar plot. No case present in the current dataset.
- **If the unit is % or score:** incremental bar plot with a mean line for reference.

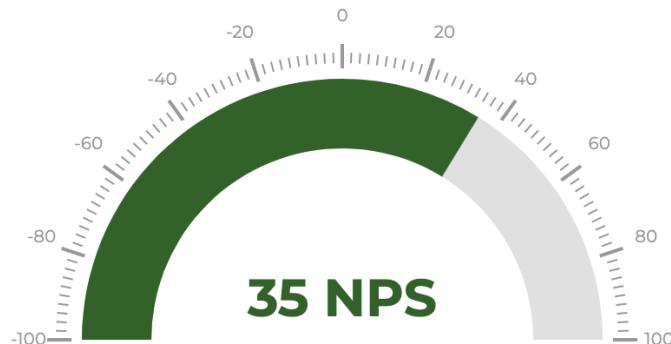
Reachability



- **If periodicity is year:**
 - **Regardless of unit:** since in this case only one value is provided, a gauge graphic will be plotted with the corresponding actual value, baseline and target. The

gauge will shift between three colours: red if under 33% of the range, yellow if between 33 and 66% of the range, green if above 66% of the range. This colour choice allows for a quick identification of the value in relation to a numerical context.

Net Promoter Score



5.4.2 Visualisation of metrics of Progress for a specific circle

These visualisations correspond to the second screen offered to the Economist, which can be added to the first one, or selected individually, after activating the toggle menu *Progress*. Rather than showcasing actual data points, these plots show different aspects of *Progress* in the context of a whole circle. The purpose of these plots beyond informative is to offer a **gamification** of KPI tracking, by providing graphics that will represent success if completely filled out. This **gamification** is extended outside the circle by allowing each Economist to have a general overview of their own circle *Progress* in relation to other circles and Projuventute as a whole.

These graphics are easily recognisable by their **yellow tones**, in contrast to the actual data plots in purple and magenta. Rules based on periodicity or unit type are not used here, as *Progress* plots are not subject to the contingency of future KPI types and are always represented in percentage.

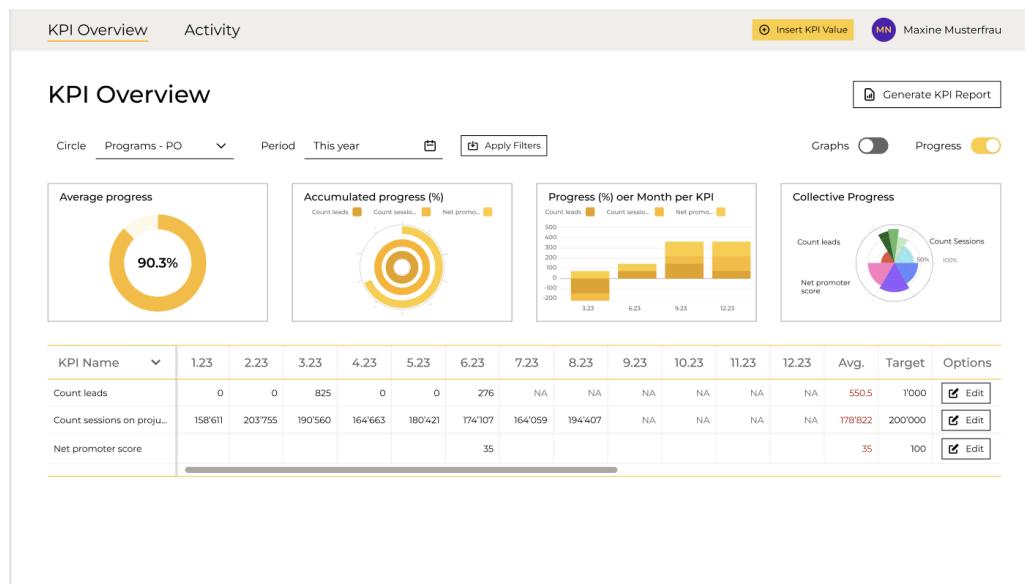
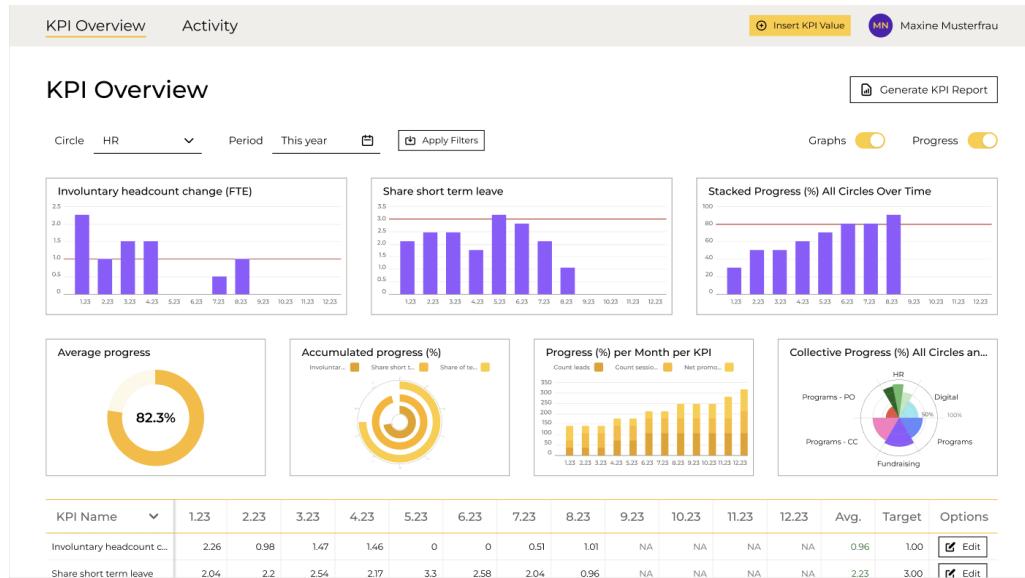
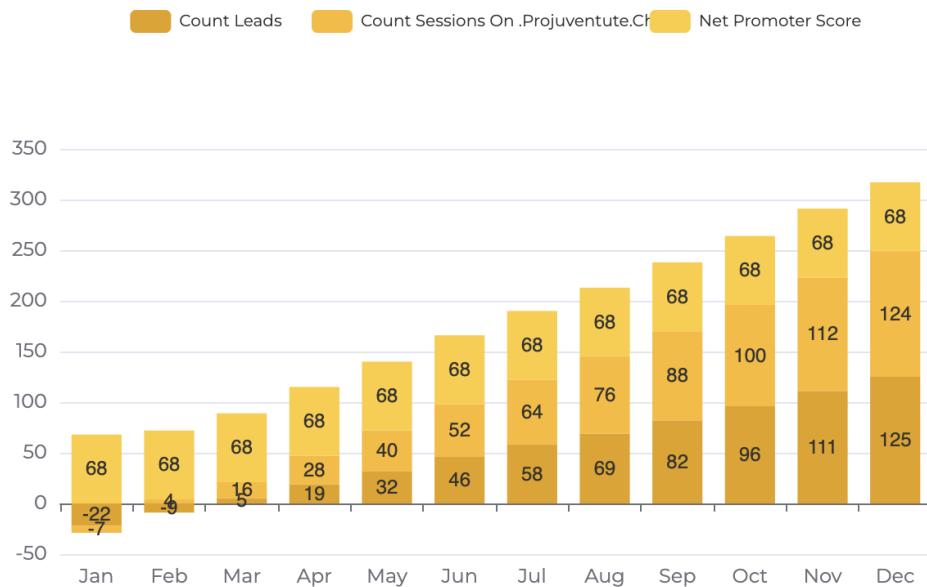


Fig 14. Economist Screen(s) with Progress plots for selected circle.

These visualisations show metrics of *Progress* for the whole selected circle, meaning that values for all KPIs of a specific circle can be either stacked or averaged. This comprehensive view on circle Progress is represented in four different kinds of graphics: bar, circular, donut and radial.

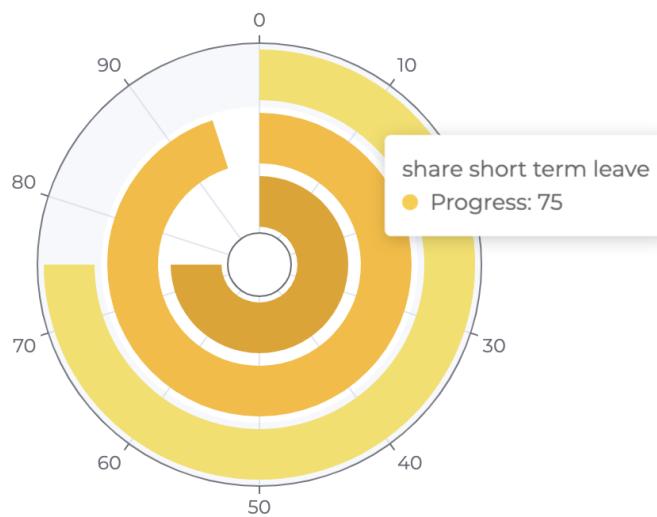
- **Stacked Bar plot:** represents the stacked *Progress* of all KPIs within a selected circle over the time from January of the selected year until the requested date. The plot is responsive to items selected in the legend, so that specific comparisons can be made on the fly.

Progress (%) per Month per KPI



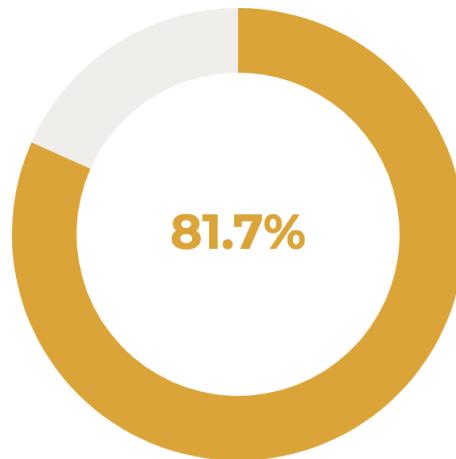
- **Circular (bar) plot:** represents the actual achieved *Progress* until the selected date for each KPI. This plot shares the same colour code as the bar plot, allowing them to share one common legend. All information are on display by mouse over.

Average Progress (%) Per KPI



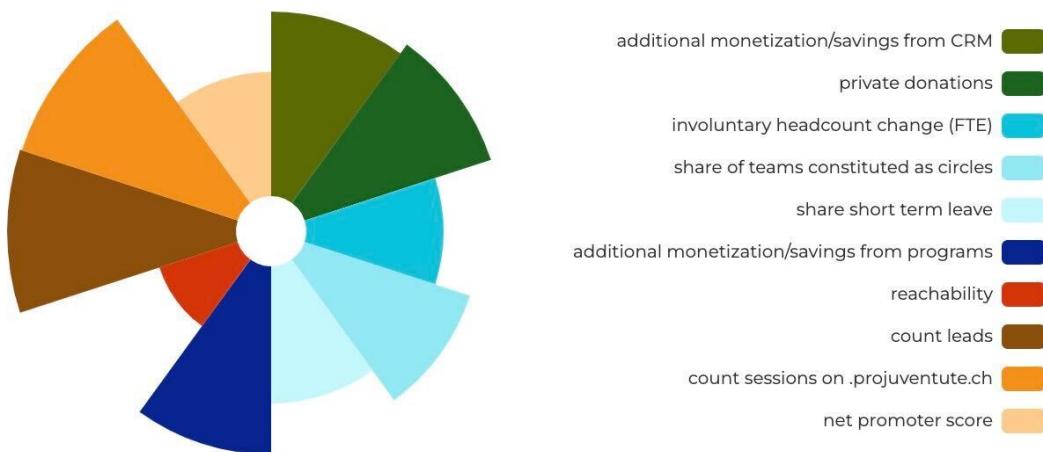
- **Donut plot:** represents the averaged *Progress* of all KPIs together at the selected date, providing a clear idea of how the circle has progressed as a whole.

Average Progress (%) For Entire Circle, All KPIs Included



- **Radial plot:** represents Progress until the selected date for every KPI in Projuventute. Each circle is recognizable by an assigned colour, and each of its KPIs with nuances of the latter. This plot offers a **collective understanding of success in a gamification process**, as the latter will be fully achieved only when all wedges together composed a full circle. The plot is responsive to items selected in the legend, so that specific comparisons can be made on the fly.

Collective Progress (%), All Circles And KPIs Included



5.4.3 Visualisations of metrics of Progress/Performance for Projuventute (all circles included)

These visualisations correspond to the default screen offered to the **Gatekeeper**. This screen is designed as a cockpit, providing an overall overview of *Progress* and *Performance* at the level of KPIs, circles and Projuventute as a whole. Within this **holistic context**, it is possible to **select KPIs belonging to different circles**, or selecting specific circles by simply using the interactive legends as selection menu. These visualisations use as colour chart a **wide palette**, allowing each circle to have its own assigned colour, and its respective KPIs as nuances of the latter.

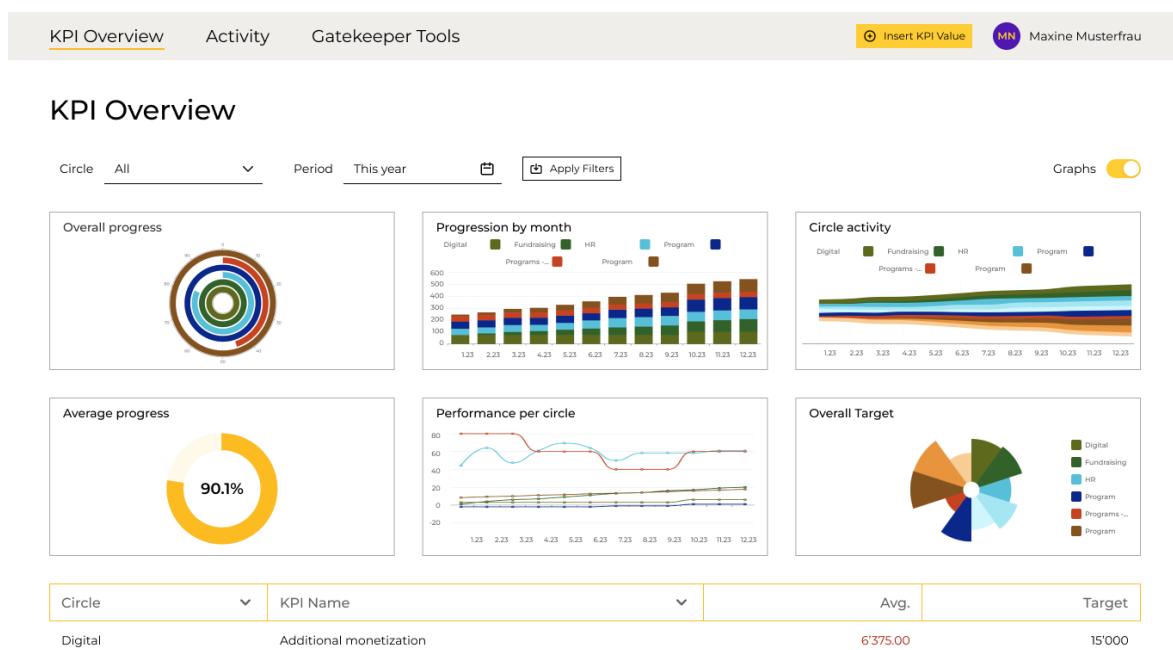
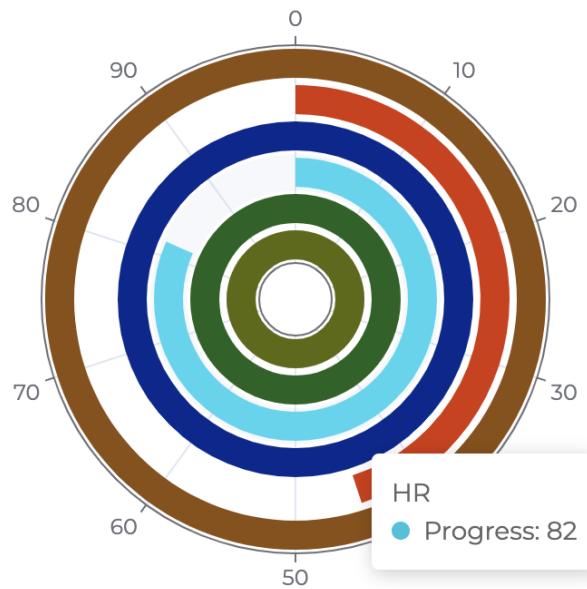


Fig 15. Gatekeeper default screen with Progress and Performance plots for all circles simultaneously.

The plots used in the cockpit are presented in three columns. The first one tracks average *Progress*, the second displays both cumulative *Progress* and *Performance* for all circles over time, and the third one showcases *Progress* at KPI level, both in a cumulative manner over time and as average *Progress* at a give point in time. These are the six types of visualisation composing this cockpit :

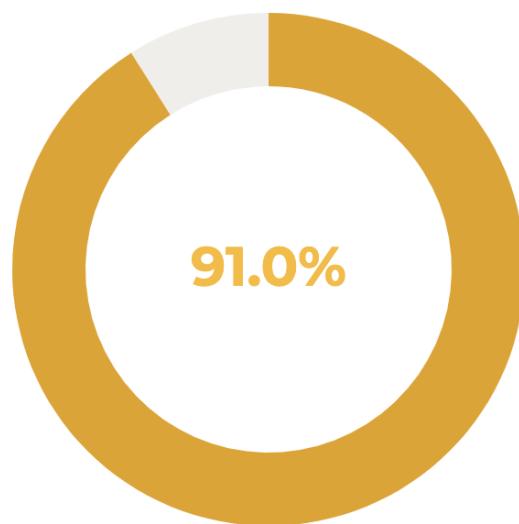
- **Circular (bar) plot:** represents the actual achieved *Progress* until the selected date for each Circle (average of its KPIs). This plot shares the same colour code as the bar and the line plots, allowing them to share one common legend. All information are on display by mouse over.

Average Progress (%) Per Circle



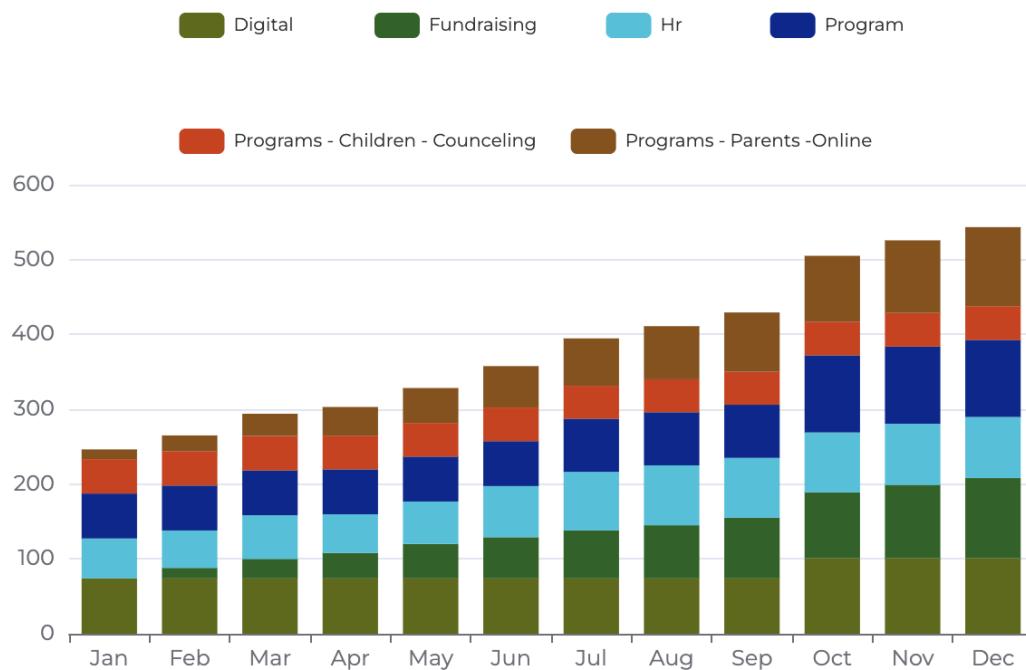
- **Donut plots:** represents the averaged *Progress* of all circles together at the selected date, providing a clear idea of how Projuventute has progressed as a whole.

Average Progress (%) For Projuventute, All Circles Included

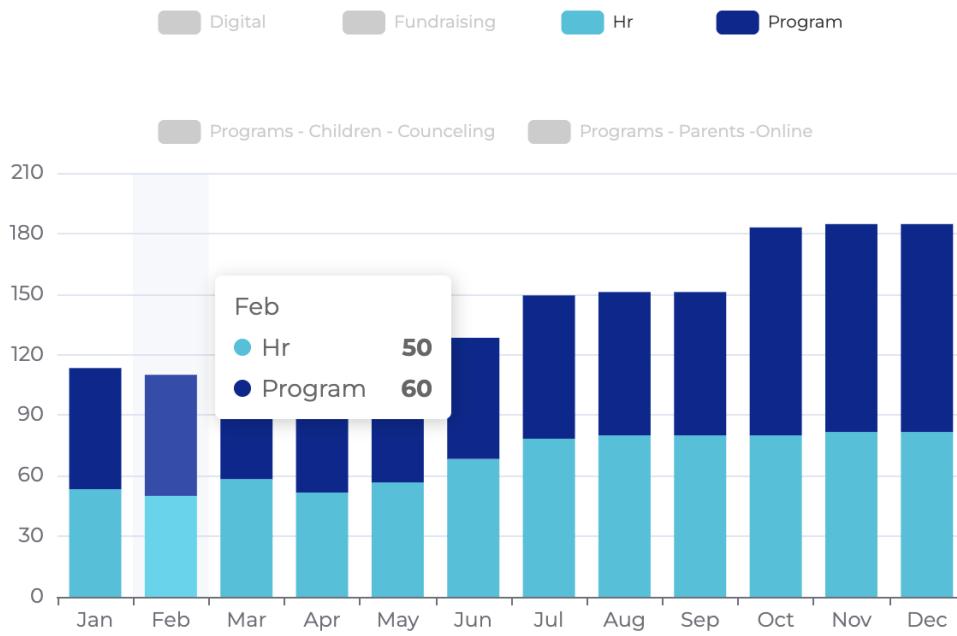


- **Stacked Bar plot:** represents the stacked Progress of all circles within Projuventute over the time from January of the selected year until the requested date. The plot is responsive to items selected in the legend, so that specific comparisons can be made on the fly.

Stacked Progress (%) Of All Circles Over Time

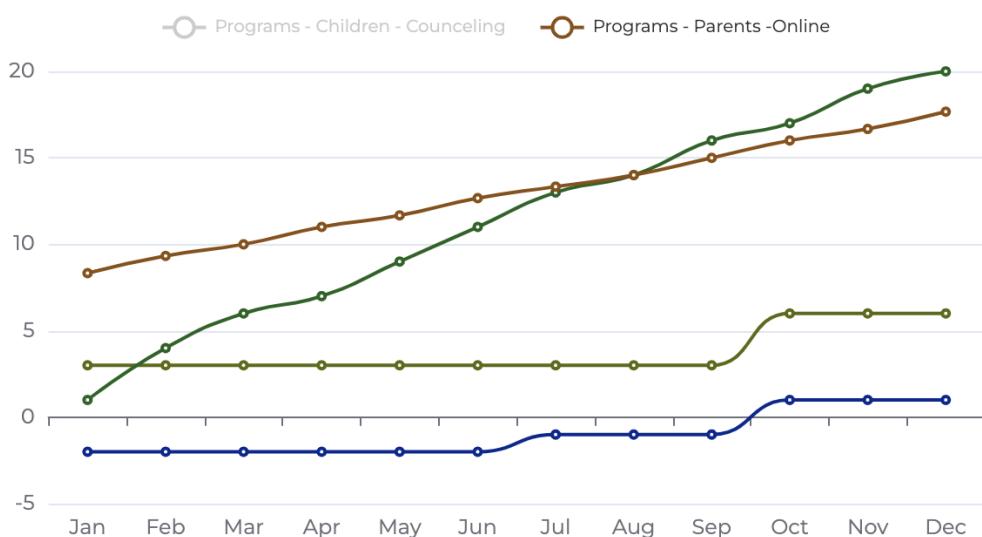


Stacked Progress (%) Of All Circles Over Time



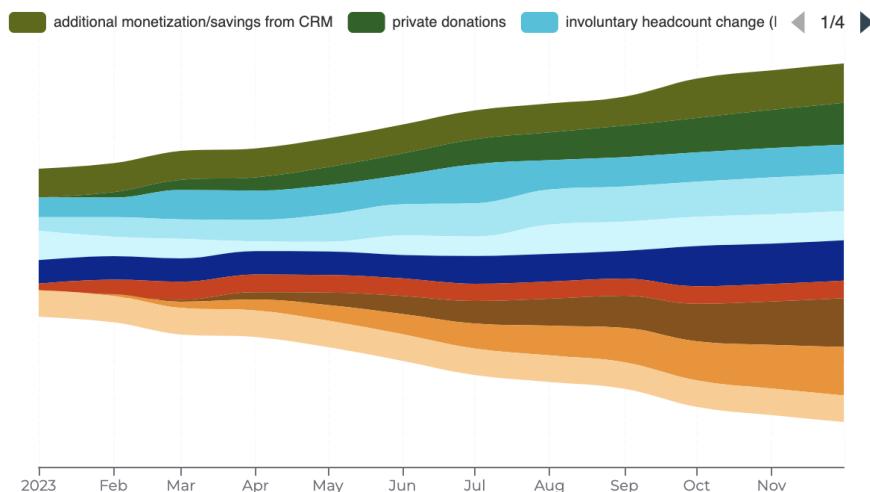
- **Line plot:** represents the averaged *Performance* of all circles within Projuventute over the time from January of the selected year until the requested date. The plot is responsive to items selected in the legend, so that specific comparisons can be made on the fly.

Historical Performance (%) Per Circle Over Time

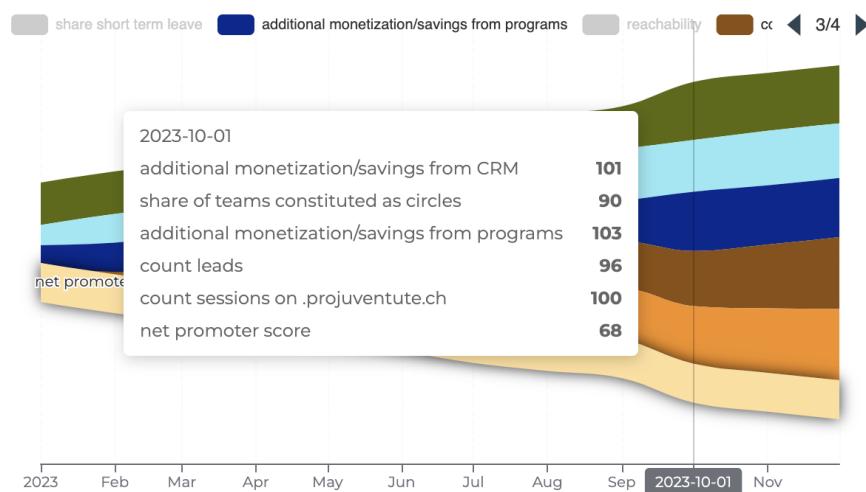


- **Radial plot:** already presented in the Economist Progress screen.
- **Streamgraph plot:** represents the relative *Progress* of all KPIs within Projuventute over the time from January of the selected year until the requested date. Each area in the plot is displayed as relative to all the others, giving a clear idea of how one KPI performs better or worse than another at a given moment or over a time span. I.e. Increase in progress during special seasons such summer. The plot is responsive to items selected in the legend, so that specific comparisons can be made on the fly.

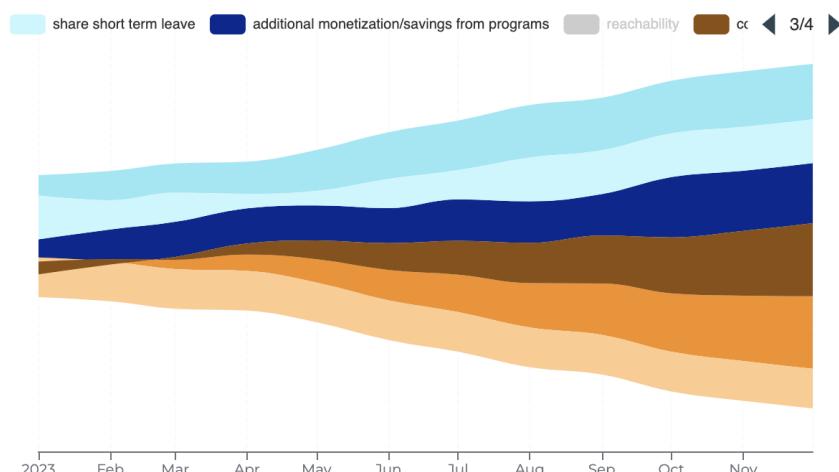
Collective Progress (%) Over Time, All Circles And KPIs Included



Collective Progress (%) Over Time, All Circles And KPIs Included



Collective Progress (%) Over Time, All Circles And KPIs Included





6. Development Documentation

6.1 About

This Flask application provides a comprehensive backend for managing users, circles, Key Performance Indicators (KPIs), and their associated values.

The application is designed with a focus on security and error handling, using Flask-JWT-Extended for secure token-based authentication and Flask-CORS for managing Cross-Origin Resource Sharing. It utilises environment variables for sensitive information, loaded securely from a .env file.

The application also includes a token blocklist feature to revoke JWTs when a user logs out, enhancing the security of user sessions.

The endpoints provided allow for robust operations such as user login, user details retrieval, user logout, circle creation, circle retrieval, KPI creation, KPI update, KPI retrieval, and KPI value operations. Furthermore, it includes the functionality to fetch change logs for KPI values and to get the most recent change log entries for a particular KPI value.

The application uses Flask's application context for managing the database connection and creating the database tables upon startup. It leverages SQLAlchemy's relationship function to define the relationships between different models.

6.2 Tech Stack

Client: Typescript, React

- [Material UI](#)
- [React Router](#)
- [Apache Echarts](#)
- [MSAL React](#)

Server: Python, Flask

Database: Postgres

Note: Please see [requirements.txt](#) and [package.json](#) for the list of all dependencies. For any testing and running scripts, please refer to the [backend-README](#) and [frontend-README](#) on Github



6.3 Getting Started

6.3.1 Getting Started Backend

We are using PostgreSQL as a Database. If you don't have it locally, please install it from PostgreSQL official site.

macOS (in your Terminal):

1. cd backend - Navigates to the Backend Directory
2. python3 -m venv venv - Sets up your Virtual Environment
3. source venv/bin/activate - Activates your Virtual Environment
4. pip install -r requirements.txt - Installs dependencies
5. psql < seed.sql - Sets up and seeds your Database
6. python3 app.py - Runs the server

6.3.1 Getting Started Frontend

1. cd frontend
2. npm install

Then you can run the following commands:

npm start

- This runs the app in the development mode.
- To view the application, open <http://localhost:3000> in your browser.
- If you make edits, the page will automatically reload.
- Any lint errors will be displayed in the console.

npm test

- This launches the test runner in an interactive watch mode.
- Refer to the section about running tests for more details.

npm run build

- This command builds the app for production, saving it to the build folder.
- React is correctly bundled in production mode, ensuring optimal performance.
- The resulting build is minified, and filenames include unique hashes.
- Once built, your app is ready for deployment!
- For further details, see the deployment section.

npm run eject

- **Caution:** Ejecting is irreversible!

- If you find the default build tools and configurations limiting, the eject command is available. This will remove the single build dependency from your project.
- Instead, all configuration files and dependencies (like webpack, Babel, ESLint, etc.) will be directly copied into your project, giving you complete control. However, this means you'll be managing these configurations on your own.
- After ejecting, all commands will still function, but they'll reference the copied scripts, allowing for custom modifications.
- You're never required to use eject. The included features cater to both small and mid-sized deployments, and there's no obligation to use this functionality. It's provided for those who need deeper customization.
- This format will look clean and organized when pasted into Google Docs. You can further enhance it with Google Docs' formatting tools like bolding, bullet points, and headings as needed.

6.4 Table Schema

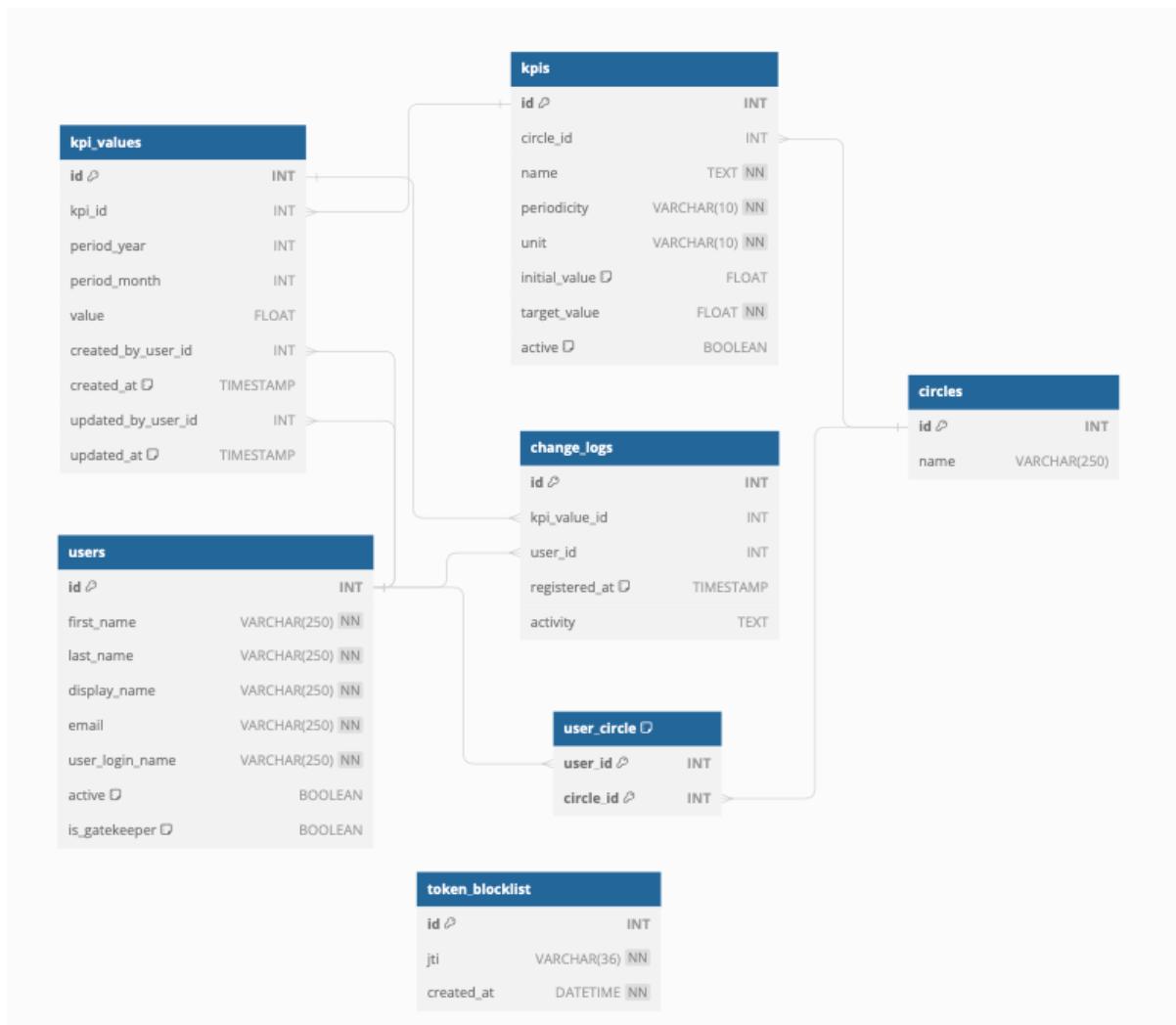


Fig 16. Database Schema



6.5 API References

Endpoint	Method	Parameters/JS	Description	Response
/login	POST	email (string, required)	User Login	User details as Dictionary and JWT token
/logout	POST	None	User Logout	Success message
/users/<int:user_id>	GET	id (int, required)	Retrieve specific user details	Details about the user as a dictionary
/users	GET	None	Retrieve all users	List of all users



/circles/<int:circle_id>	GET	id (int)	Retrieve specific circle details	Details about a circle as a dictionary
/circles	POST	name (string, required)	Add a circle	Success message
/circles	GET	None	Retrieve all circles	A list of all circles as dictionary
/kpis/add	POST	Various parameters (like name, circle_id, etc.)	Add a KPI	Success Message
/kpis/<int:kpi_id>/edit	PUT	Various parameters	Edit KPI details	Success message
/kpis/<int:kpi_id>	GET	id (int)	Retrieve specific KPI details	Return KPI as a dictionary



/kpis	GET	None	Retrieve all KPIs	List of all KPIs as dictionary
/kpi_values/add	POST	Various parameters	Add a KPI Value	Success message
/kpi_values/<int:kpi_value_id>/edit	PUT	Various parameters	Edit KPI Value details	Success message
/kpi_values	GET	None	Retrieve all KPI values	A list of all KPI values as dictionary
/kpi_values/change_log	GET	Various parameters for filtering	Retrieve change log for KPI values	Specific response related to change logs
/kpi_values/<int:kpi_values_id>/change_log	GET	kpi_values_id (int, required)	Retrieve recent change logs for a KPI value	Recent 3 rows as a dictionary



6.6 Running Tests

To run tests, run the following commands in your terminal:

1. cd backend
2. python -m pytest

6.7 Features to be implemented

Automatic input

We could build a pipeline to automatically ingest values from a DWH (data warehouse). Currently, we maintain two datasets of user profiles and authentications, one is in Microsoft Azure Active Directory and the other one is in our Postgres database. In the app, users have to log in their valid Microsoft account first, and our app will check in the database if the user is authenticated to use the app. The data in our app to authenticate users can be manually updated by IT managers, or ingested from a DWH connected to AD.

KPI to team/circle relation: 1:n

Some KPIs with respective values can be used in multiple circles to compute metrics and be used for data visualisations. To implement this, change the column *circle_id* in table *kpis* to a list of ids, or add another relation table with *kpi_id* and *circle_id* to show which kpis can be used by which circles.

KPI to team/circle relation: m:n

Some KPIs can be reused in multiple circles so that they don't need to create the same KPI structure for different circles. In this scenario, KPI values are not shared between circles. We can modify the solution above: the relation table with *kpi_id* and *circle_id* is used for determining which kpi (structure not values) can be reused among circles; the column *circle_id* represents which circles can access the values of this kpi. If we would like to introduce different details for the reusable kpi, e.g. different target values for different circles, we can split the column in table *kpis* into the relation table with *kpi_id* and *circle_id*.

7. Team Learnings

7.1 Team learnings from the retrospectives:

Learning from the expertise of other team members: As a team, we've realized that tapping into the expertise of our fellow team members is a valuable skill. It involves actively seeking and absorbing insights, best practices, and wisdom from colleagues with diverse skill sets.

Learning of new platforms and technologies: As a team, we've come to appreciate the importance of acquiring proficiency in new platforms and technologies. Meaning we kept ourselves updated with the new tools we didn't use before and systems that allowed us to expand our capabilities and stay in line with the features we wanted to develop.

New development skills : In the context of our project, our team actively embraced new skills development. For instance, the developers delved into microservices architecture and SSO, the data scientist delved into advanced statistical modeling techniques, and the UX team deepened their proficiency in user testing methodologies. This ongoing pursuit of new skills made us a more versatile and capable team, better prepared to address a wide array of challenges in our KPI app development.

Team collaboration: During our project, we proactively worked on improving our team collaboration skills. We conducted regular cross-functional meetings, chats, encouraging the exchange of insights and ideas. This collaborative atmosphere of open communication and mutual support allowed us to combine our expertise and carry out tasks more efficiently, ultimately ensuring the success of our KPI app development.

Real-life practice of new knowledge : In our project, we made a conscious effort to practise new learnings from each sprint. For instance, after gaining front-end development skills in Sprint 2-3, the development team immediately applied these skills to improve the KPI app's user interface in subsequent sprints. This proactive approach ensured that the new skills and knowledge gained in each sprint were effectively integrated into the project, resulting in continuous improvement and growth.

7.2 Summary of individual learnings:

In our commitment to transparency and continuous improvement, we believe it's essential to highlight the individual learning experiences of each team member as part of our documentation. These individual learnings not only showcase the growth and development of our team members but also provide insights into the diverse skill improvement targets pursued by each member.



Common Skill Improvement Targets: 🎯

Collaboration and Teamwork: Several team members aimed to improve their collaboration skills and effectively work within cross-functional teams. They all made progress in this aspect by actively participating in team activities and meetings.

Technical Knowledge and Tools: Many team members sought to enhance their technical knowledge and proficiency with various tools, such as GitHub, Avion, and data visualization libraries. They made substantial progress in these areas by actively using these tools in their work.

Mentorship and Guidance: A recurring theme was the desire to mentor junior team members and provide structured guidance. Team members like Alina, Michael, and Miren made significant strides in this regard by mentoring, conducting knowledge-sharing sessions, and facilitating integration of new team members.

Individual Skill Improvement: 🌟

Diana: Explored further into JS data visualization libraries and how to collaborate in a full stack team.

Elisabeth: Learned more about GitHub, Avion, and user stories. Improved her user-centered design skills.

Alina: Developed leadership skills, mentored junior developers effectively, and improved project management abilities.

Mariana: Enhanced data visualization skills and made substantial progress in UX research and design.

Michael: Collaborated effectively on frontend and backend development, improved communication skills, and gained insight into project progression.

Miren: Improved project planning, enhanced stakeholder engagement, and learned more about the app creation process.

Yara: Gained deeper knowledge of GitHub and Avion, improved communication within the Scrum framework, and facilitated Scrum events effectively.

Mohamed: Learned how to integrate SSO on both ends, how to use flask-jwt-extended, and how to collaborate with a team where each is responsible for the full development cycle.

Overall Impact:

The team collectively achieved significant progress in their skill improvement targets.

Collaboration, technical knowledge, mentorship, and project management skills were common areas of improvement across the team.



The team's commitment to continuous learning and development is evident in their achievements over multiple sprints.

As a team, we are very thankful for all these great experiences and how much we have learned. 🙏✨ By sharing these individual learning experiences, we aim to not only celebrate the growth of our team members but also inspire and encourage further learning and development within our team. It showcases our dedication to fostering a culture of improvement and collaboration. 🌱😊