UNICORN VYSOKÁ ŠKOLA S.R.O.

DIPLOMA THESIS

2025

Bc. Filip DITRICH

UNICORN VYSOKÁ ŠKOLA s.r.o.

Software Engineering and Big Data

DIPLOMA THESIS

Cashless festival data analysis and analytical dashboard development

Author: Bc. Filip Ditrich

Supervisor: Mgr. Václav Alt, Ph. D.

Prague 2025

Statutory Declaration

I hereby declare that I have written my Diploma Thesis on the topic of *Cashless festival data analysis and analytical dashboard development* by myself, under the guidance of my thesis supervisor, using only the technical publications and other information sources which are all quoted in the thesis and listed in the bibliography. I declare that artificial intelligence tools have been used only for support activities and in accordance with the principle of academic ethics.

As the author of this Diploma Thesis, I also declare that in association with its writing, I have not violated the copyright of any third party or parties, and I am fully aware of the consequences of provisions of s. 11 et seq. of Act No. 121/2000 Coll., the Copyright Act.

Furthermore, I hereby declare that the submitted hard copy of this Diploma Thesis is identical to the electronic version I have submitted.

In .... on ....

Bc. Filip Ditrich

Acknowledgements

I would like to express my deepest and most sincere gratitude to my supervisor, **Mgr. Václav Alt, Ph. D.**, for his exceptional guidance, prompt feedback, and unwavering moral support throughout this challenging journey, particularly during the most demanding phases of the thesis.

Special thanks go to our partner event organizer and NFCtron for providing the opportunity to conduct this research and for their invaluable assistance during the initial stages of the project.

I am also profoundly grateful to my partner for her endless patience and encouragement, and to my family and friends, whose understanding and support made this work possible.

Cashless festival data analysis and analytical dashboard development

Abstract

This thesis analyzes payment transaction data from a larger Czech festival that used the NFCtron payment system. The research is conducted on over 141,000 transactions from more than 10,000 unique attendees and focuses on 29 research questions related to Cashflow and Revenue Sources, System Performance, Beverage Consumption, and Customer Behavior analysis.

The methodology ranges from establishing a local environment for the analysis, through data extraction cleaning and preparation for a local database, to data anonymization due to privacy concerns.

All 29 research questions are thoroughly answered and presented with rich visuals using a variety of charts and tables in each of the four major research areas. Moreover, the findings led to the development of an interactive analytical dashboard prototype using Dash and Plotly technologies that demonstrates the key insights.

The process of the analysis and dashboard development is thoroughly described in the thesis, including the main technical challenges faced and used solutions to overcome them. Although the dashboard app is still in the prototype stage, the analytical findings, data-obtaining techniques and anonymization methods provide a solid foundation for future development and have already contributed to improvements in the personal line of work.

Keywords: festival data analysis, cashless payments, data visualization, data anonymization, interactive analytic dashboard, Python, Dash and Plotly

Contents

Introduction

Background and Motivation

Payments at festivals are a crucial part of the successful event management. The shift from cash to cashless payments has been a significant trend in the last decade that has brought many benefits to both festival organizers and attendees[1].

However, traditional cashless payment systems using payment terminals are not only expensive to reliably implement at a venue, where the internet connection is often unreliable, but also do not provide any insights into the data generated by the transactions. In the best case scenario, the organizers are able to generate a report of processed transactions made by each terminal.

Which, frankly, is not enough to make any actionable decisions based on the data. Moreover, given the event organizers are not data scientists, they often lack the knowledge and tools to analyze the data and extract valuable insights from it.

That is where NFCtron comes in and offers a solution that not only provides a reliable cashless payment system, with credit-based NFC chip bracelets supporting offline mode or card terminal payment solutions, but also provides a comprehensive B2B platform. This platform allows organizers, vendors, and event third-party partners to benefit from the data that the system operates with[2].

The system is a full-scope solution that provides from initial online ticket sales and online credit top-up, through attendee check-in, on-site credit top-up, attendee access control, security monitoring, vendor sales, and inventory management. Most importantly, it provides fast and reliable payment processing with the real-time data analytics and reporting, all the way to the post-event automatic settlement and reporting.

It simply provides everything an event organizer needs to successfully and efficiently manage their event without the need to worry about any technicalities or staff management. NFCtron not only provides the system as a service, but also experienced Event Managers, cashiers, check-in brigadiers, and other staff to operate the system and the event itself. This allows organizers to focus on the event communication, marketing, line-ups, and other important aspects of the event.

Put, NFCtron offers organizers a peace of mind and a guarantee that their event will be a success.

NFCtron Company

is a Czech company that has been operating since around 2019. In its early beginning during a COVID-19 pandemic was on the verge of survival because of the event industry being paralyzed by the government restrictions. However, the company survived and even in these challenging times managed to turn the disadvantage into an advantage by focusing on the core system and product development[3].

Years later, the system became robust and reliable. It is now used by many event organizers across the Czech Republic and Slovakia and is currently expanding to other countries in Central Europe such as Austria[4], Poland, and Germany. In its primary market – the Czech Republic – NFCtron penetrated the market and is now the leading cashless payment system provider for festivals and other events.

In recent years, the company has also been focusing on expanding to the Payments market, focusing both on Card Acquiring and Card Issuing. From the acquiring part, it is now actively developing its own SoftPOS solution that will allow vendors to accept payments via mobile phones. On the other hand, on the Issuing part, it is also working on Card Issuing; that will allow the company to issue its own NFCtron branded payment cards in cooperation with Mastercard[5].

A big part of the successful market penetration was the company's focus on the business-to-business (B2B) side of the business with event organizers. Giving the organizers amounts of data improving their decision-making and providing them with insights allowing them to optimize their events. And most importantly, providing them with economic aspects and cashflow optimizations that allow many events and festivals to survive and continue to operate.

Personal Position and Motivation

I have been with the company from the COVID-19 times. My current position is a Chief Product Officer (CPO), and I am responsible for the product development and the product management of all the products and services that NFCtron offers. This allows me to have a deep insight into the system and most importantly, access to all the data that the system generates. As previously mentioned, the company's success is based on the B2B side of the business, and that means services and products provided to the event organizers.

The main product that organizers have access to is the platform called **NFCtron Hub**. My personal goal and personal motivation to work on this thesis is to discover new ways to improve the platform and provide even more valuable insights to the organizers.

Problem Statement

Even though the system provides a lot of data, it still has a lot of potential to provide even more valuable insights to the organizers. Currently, the previously mentioned B2B platform, **NFCtron Hub**, provides a real-time data analytics dashboard presenting the most important KPIs and metrics to the organizers.

These KPIs and metrics summarize:

- **Total sales**: the total amount spent on online ticket sales and on-site payments.
- **Total sales in time**: the total amount split into time intervals.
- **Total refunds**: the total number of sale reversals or refunds made including refunds from online tickets, refunds of on-site payments, and chip credit refunds.
- **Chip balances**: the current balance topped-up on the NFC chip bracelets on-site or pre-topped-up online.

- **Customer orders rating**: customers can rate their orders via an NFCtron mobile application, which provides the organizers with feedback on the vendor's performance and the quality of the products sold.

Moreover, it provides less clear data such as

- **List of vendors**: the list of vendors presents at the event with their sales and rating.
- **List of products**: the list of products sold at the event with their sales and rating.
- **List of sale places**: the list of sale places at the event with their sales and rating.
- **List of top-up places**: the list of top-up places at the event with the number of top-ups made.
- **List of customer chips**: the list of unique individual NFC chips issued to the customers with their balance, spending, and security status.
- **List of customer ratings**: the list of customer ratings with their feedback from points of sale.

And finally, it also provides unstructured data in the form of data exports in tabular format that can be used for further analysis:

- **Product exports** – list of all products sold with summarized metrics.
- **Points of sale exports** – list of all selling points with summarized metrics.
- **Vendor exports** – list of all vendors with summarized metrics.
- **Deal exports** – list of summarized sales made under a deal[1] between the organizer and the vendor.
- **Transaction exports** – a heavy export of all transactions made at the event.
- **Ticket redeems exports** – list of all tickets redeemed at the event.

[1] A deal is an arrangment between the organizer and the vendor that states the terms of the vendor's presence at the event, the products they are allowed to sell, the price of the products, the commission the vendor pays to the organizer and other terms of the deal.

- **And other exports regarding the online sales** – list of all online ticket sales, top-ups, receipts, customers, and other data.

With the above giving some initial picture about the capabilities of the system, the platform (and thus the organizers) still face several problems or challenges that need to be addressed:

Problem 1: The main KPI metrics may provide some core insights, but the other less or unstructured data is not used to its potential.

Problem 2: The organizers are not data scientists and need a simple and clear way to understand the data.

Problem 3: Even with all this data, there is still a lot that can be done to dig deeper and provide more valuable insights.

Objectives of the Work

With the problems stated above, the main goal of this thesis is to analyze, answer, and present results to important questions about the available data and the potential insights that can be extracted from it.

But to achieve this, it was a prerequisite to find a willing event organizer that would provide the data and would be willing to cooperate on the project. Cooperate in terms of providing valuable insights into what they would like to know more about their event.

In Cooperation with the Event Organizer

For this purpose, I chose an undisclosed event organizer who has been a close and helpful partner of NFCtron for many seasons. Together with the organizer in the first step, we have stated the following requirements to perform the analysis:

- **Requirement 1**: The event and organizer should be kept undisclosed.
- **Requirement 2**: The data should be anonymized to not leak any possible sensitive information about vendors or customers.

The next step was to choose an event from which the data will be used. As it cannot be disclosed any further, we will refer to the event as **The Event** and the organizer as **The Organizer**.

Now the important information about **The Event** for this study is the following:

- **The Event** is a music festival that has been organized for several years now.
- **The Event** takes place in the Czech Republic at the beginning of July 2025.
- **The Event** is a 3-day event with multiple stages and multiple vendors.
- **The Event** uses NFCtron system for cashless payments and access control.
- **The Event** had around 7,000 attendees in 2024 and had a roughly 43% increase in 2025 to around 10,000 attendees.

Data Analysis Objectives

The final step was to define questions or data analysis objectives that should be answered or achieved by the end of the thesis.

Together in the cooperation with **The Organizer** and several internal colleagues in NFCtron, we have defined the following questions for the data analysis:

Cashflow and Revenue

- ***RQ1***: *What was the organizer's total revenue and what did it consist of?*
- ***RQ2***: *How much and how was the balance topped up on the chips?*
- ***RQ3***: *What is the remaining balance on all chips after the event and refunds?*
- ***RQ4***: *What was the total sales of the event, how much of it was sold by the organizer, and how many external vendors?*

These questions should shed light on the event's cash flow and revenue sources, which the platform does not currently cover in detail. Possible answers to these questions may provide the organizer with valuable insights into the event's economic aspects, allowing them to optimize cashflow and revenue sources for the following event.

Performance

- ***RQ5***: *How many transactions were processed in total, and when was the system's largest "peak" in transaction volume?*
- ***RQ6***: *What was the average transaction processing time during peak hours?*
- ***RQ7***: *Were there any significant delays or downtimes in processing transactions?*
- ***RQ8***: *What were the best sale places?*
- ***RQ9***: *What were the best top-up points?*
- ***RQ10***: *Who were the best vendors?*
- ***RQ11***: *What were the best products?*

The current platform already provides some performance metrics, but these questions should provide more detailed insights into the performance of the event.

Beverage Consumption

- ***RQ12***: *What was the total amount of beverage consumed?*
- ***RQ13***: *How many returnable cups were issued and returned/not returned?*
- ***RQ14***: *What was the most popular beverage category?*
- ***RQ15***: *What was the top beer brand, and how much was consumed and sold?*
- ***RQ16***: *What was the most popular brand of other alcoholic beverages, and how much was consumed and sold?*

- ***RQ17***: *What was the most popular non-alcoholic beverage brand, and how much was sold?*

Currently, a more in-depth product analysis is missing in the platform, and the most important part of the product sales analysis at festivals is the beverage consumption. These questions should try to answer and give detailed insights into the beverage consumption, preferences, and sales at the event.

Customers

- ***RQ18***: *What was the total attendance at the event, and how many active customers were there each day?*
- ***RQ19***: *How many customers topped up their credit in advance online?*
- ***RQ20***: *What was the distribution of customers by type (on-site, online, staff, guest, VIP)?*
- ***RQ21***: *How many customers used the mobile app?*
- ***RQ22***: *What is the distribution of bank accounts used to refund credit?*
- ***RQ23***: *What is the distribution of card schemes used to top up credit both on-site and online?*
- ***RQ24***: *What was the course of the event in terms of new visitors? And when were the largest "peaks"?*
- ***RQ25***: *What is the average time of a visitor from arrival to first transaction?*
- ***RQ26***: *What was the course of the event in terms of topping up credit on-site? And when were the largest "peaks"?*
- ***RQ27***: *What was the customer's onsite credit top-up frequency?*
- ***RQ28***: *What were the beverage preferences throughout the day?*
- ***RQ29***: *What were the most common product combinations?*

The customer analysis is the most important part of the data analysis. It is crucial for the festival organizers to know their customer base and their behavior to optimize the event and make it more attractive for the customers.

Currently, no customer analysis, other than the customer ratings and list of customer chips, is available on the platform.

Answering these questions will possibly lead to the most valuable insights about the event's customer base and their behavior that no other platform or system currently provides.

Making these questions crucial and most valuable for the organizer and for the platform itself.

Technical Objectives

Answering the above questions will require a technical solution that will be able to process the data and provide the answers.

The scope of this study is not to implement a new system or a new platform and not even to implement any new changes to the existing NFCtron Hub platform. It is to find answers to the above questions and present them in a clear and understandable way in the form of a simple internal dashboard.

The technical goals of this study are:

- Prepare, process, and analyze the data from **The Event**.
- Find answers to the above questions.
- Implement a simple internal dashboard that will present the answers to the questions.

Scope of the Study

To ensure the feasibility and focus of the study, certain boundaries have been defined in terms of what is included and excluded from the scope of the study.

Included in the Scope

- The study will focus on transactional, customer, and operational data from a specific event, referred to as **The Event**.
- Key areas of analysis include cashflow, revenue sources, performance indicators, beverage consumption, and customer segmentation and behavior.
- The data analyzed includes pre-event (e.g., online top-ups), during-event (e.g., chip transactions, sales), and post-event data (e.g., credit refunds).
- A prototype dashboard will be developed using Python's Dash and Plotly libraries to present the key insights.
- The dashboard is intended for internal use and post-event analysis by the event organizer.

Excluded from the Scope

- **Real-Time Monitoring**: While the dashboard may be designed with real-time data potential, this study will focus solely on post-event analysis.
- **Multiple Event Comparisons**: This study is limited to the analysis of a single event (**The Event**) and does not involve comparative studies.
- **Data Collection**: The study does not involve the collection of new data and relies on data provided by **The Organizer** and the NFCtron system.
- **Implementation in NFCtron Hub**: The thesis focuses on analyzing data and developing a standalone prototype dashboard, not on direct integration into the NFCtron Hub platform.

Limitations

- **Anonymized Data**: To protect privacy, all customer and vendor data has been anonymized, which may restrict the analysis in some ways.
- **Single Event Focus**: Insights and recommendations are based only on data from a single event (**The Event**), which may limit broader applicability.

- **Time Constraints**: Due to the thesis's timeline, advanced features (e.g., predictive analytics) and technical implementations were deprioritized but will be considered for future work.

1 Data and Methodology

This chapter addresses the process and challenges of local environment setup, obtaining, preparing and anonymizing the data. Most importantly, this chapter describes and explains the data used for this research. It also briefly describes the tools, technologies, and methods employed to answer the research questions.

1.1 Environment and local setup

To start off, we needed to set up some kind of environment where we would later work with the data. The data we would be working with was stored in a PostgreSQL database.

Having direct access to the production database to perform the analysis was not a secure and ethical way to go. Exporting only the necessary and raw data from the production database was an initial thought, but we initially did not know what data we would need, and by exporting we would lose all the relations between the tables.

Therefore, we decided to set up a local database with the same structure as the production database where we can query and analyze the data safely. The next step was to import the data from the production database to the local database. Importing or simple cloning the full database was also not an option because only a small fraction of its subset was required.

So a deep internal analysis of the tables that were relevant to our study was performed. This resulted in a list of total 21 tables that held the necessary data for the study and were necessary to be imported.

1.1.1 Data Obtaining and Preparation

Almost every table was easily queried for the event and exported from the production database to a local CSV file. But some tables (for example, and not surprisingly, the *transaction* table with over 140k rows) were too large to be exported in one piece, so we had to split them into smaller parts. Later these parts were joined together to a single CSV file using a simple Python script.

Since no direct access to the production database was used for the export but rather a database management tool, the export was not as fast as it could be and took a significant amount of time. Moreover, the exported data, most importantly the timestamps, were in a different format than we needed. And also, all numeric values were exported as a formatted string with a comma as a decimal separator. So a data-preprocessing Python script was written to convert such invalid columns to the correct format.

1.1.2 Local Database Setup

Then a step to set up the local PostgreSQL database was needed. Due to the nature of this study, we wanted to keep the setup as simple as possible, so we used the default PostgreSQL installation without using any special environment using Docker or similar. However, during this process we made a mistake and forgot that a PostgreSQL with a PostGIS extension[1] was needed. This, unfortunately, required re-setting up the database with the PostGIS extension.

The next step was to import the data from the CSV files to the local database. For further database handling, analysis, and visualization, we used DataSpell, a Python IDE with a built-in database explorer and data visualization tools. DataSpell was then used for the local database import, which prior to it required some necessary database relations and constraints modifications, since the data was exported without them and was not relevant for the study.

This whole process resulted in approximately 387k rows of data in the local database that were ready to be queried and analyzed.

[1] PostGIS is a spatial database extender for PostgreSQL object-relational database[6]

1.1.3 Local Database Modifications

Before any analysis was performed, some modifications to the local database were needed due to some known limitations and missing data.

Beverage Volumes

The first necessary limitation that the Beverage Consumption Analysis section heavily relied on was the missing information about beverage products volume in milliliters. This information was crucial for the analysis, so a new column was added to the relevant product information tables. However, the next step was to back-fill this information, which was not easily automated.

The First approach was to write a Python script that would try to find the volume information from the product name. This worked for some products but not for all since the naming convention was not consistent.

After several attempts to automate this process, it was decided to manually fill in the missing information since only 425 products were present in the database. Only 159 of them were of the beverage type and thus eligible for the volume information.

Returnable Products

Since one of the research questions was to analyze the returnable cups and this information was not easily available in the database, a new column was added to the product information tables.

This was a simple binary column that indicated whether the product was returnable or not. Back-filling this information was also pretty straightforward since only one product was a returnable cup.

Venue Map Visualization

One of the initial ideas was to visualize the venue map with the locations of the selling places, top-up service points, stages, and other important places.

This would be invaluable, the database was partially ready for this, but the data would be significantly time-consuming to back-fill and the later analysis and visualization would require more time.

Since these facts and the fast that this process of preparing the data took place before completing the list of data analysis questions, this idea has been later abandoned.

Event Program

To present some time-related data and its correlation with the event program, it would require having the event program in the database.

Again, the database was ready for this, but no event program was set up since it was unnecessary for the event. Therefore, this required getting the event program from the festival website and manually inserting it into the database.

This was manually a very time-consuming process, but it was necessary for the analysis. For some simplification of the process, an AI tool was used to extract the data from the program schedule screenshots and instructed to prepare an SQL script that would insert the data into the database.

This seemed like a good idea, but the AI tool was initially hallucinating and made up some incorrect data. But after several iterations, it successfully extracted the data and prepared the SQL script which was used, and the event program was successfully inserted into the database.

In the end, one can doubt that this process was faster than manual data entry, but it was a good exercise and a good example of how AI can be used to automate some processes.

1.2 Data Anonymization

The Data Anonymization process was necessary due to requirements initially set by the data provider and later by the ethical considerations. This step was performed for the already imported data in the local database. It required identifying the sensitive data and replacing them with anonymized values.

> **What is Data Anonymization?**
>
> **Data anonymization** involves removing or encrypting sensitive data, including personally identifiable information (PII), protected health information (PHI), and other non-personal commercial sensitive data such as revenue or IP, from a data set. Its intent is to protect data subjects' privacy and confidentiality while still allowing data to be retained and used[7].

In this case, the most sensitive data were:

- **Vendor names**: Since it included the legal names of the vendors, it was necessary to anonymize them.
- **Selling places**: Some selling places were named after the vendors, so it was necessary to anonymize them as well.
- **Customer information**: Some tables included customer information like names, emails, phone numbers, etc.

The process could have been done various ways, but the fact that this study will not be exposing internal database structure, it was decided to perform the anonymization directly in the database.

However, if one-way anonymization were to be performed, it would permanently overwrite the original data, losing the possibility to switch from anonymized to original data. Therefore, a two-way anonymization process was chosen and performed.

This was particularly useful during the analysis phase, where the results would contain the original data for better understanding, fact-checking and for the internal presentation and consultations with the organizer.

It was done on the database level, where two new internal tables were introduced – *public.anonymization_config* and *public.original_values*.

Where the *public.anonymization_config* table held the configuration about which schema, table, and column should be anonymized and how. For the usage, a simple SQL function was created to define the anonymization configuration in a simple JSON format that looked like in the **Source Code** 1.

The *public.original_values* table was used to store the original values of the anonymized columns. Again, using a simple SQL function *anonymize_database()*,

```sql
SELECT configure_anonymization('[
  { "table": "schema.seller", "columns": ["legal_name", "name"] },
  ...
  { "table": "schema.user_account", "columns": ["email", "last_name",
  ↪  "first_name", "phone"] },
  ]'::JSONB
);
```

Source Code 1: Anonymization configuration example

it would store the original values and anonymize the configured table columns.

One particular challenge was anonymizing the values smartly. It could have been easily done by replacing the values with random strings, hashes, or encrypted values. But working with data where a vendor is named *fa65165b923e9cc* is not very convenient.

Therefore, a simple SQL function was written to anonymize the value depending on the configuration. This allowed configuring the anonymization to:

- replace vendor names with values like *Vendor 1*,
- customer emails with *03b09592-d0eb-43a3-9941-30d38ade6bce@gmail.com* keeping the original email domain,
- selling places with values like *Place 1* where the original name contained sensitive information, but keep original values for places like *BAR L2*, etc.

In the end, it resulted in a database with anonymized values per stated configuration that could have been used for the analysis and results presentation safely. However, the original values were still present in the database, and the database could have been anytime easily restored to the original state if needed and vice versa.

1.3 Data Structure

Without exposing the internal database structure, the abstract data structure this study was working with can be described as follows:

1.3.1 Financial Data

Transactions (approx. 300k rows): Transactional data, that holds the information about the type, amount, timestamps, and links to products, places, and other related entities. This analysis relies on and works with several transaction types, including **top-up charge** and **refund** transactions [2], **order sales** and **refund** transactions [3] and **chip registrations**[4].

Credit Refunds (approx. 15k rows): Post-event credit refund requested by the customers with the information about the amount, timestamp, customer, and the bank account to which the refund was sent. This data enables our analysis to correctly handle disposable credit balances, more information about the anonymous customers [5] and their behavior.

> **Why is it important?**
>
> These records also form the backbone of the analysis, enabling insights into revenue, sales trends, and customer behavior.

1.3.2 Customer Data

User Accounts (approx. 5k rows): Registered customer accounts with the information about the user and its potential online order history and other related information.

Tickets and Orders (approx. 30k rows): Information about the online sold tickets, its types, prices, timestamps, online order-related information.

> **Why is it important?**
>
> With the above data, this analysis can work with more customer information supporting the customer behavior analysis, customer segmentation, and other related analysis.

[2] Top-up transactions mean funding or refunding chip credit balances.
[3] Order transactions mean spending the credit balance for products.
[4] Chip registrations mean records of when the system registered the chip bracelets
[5] Anonymous customer essentially means a user without a registered account

1.3.3 Event Data

Places (approx. 400 rows): Selling and top-up service points, zones for access control and its other relations.

Products (approx. 500 rows): Essentially a product catalog including the product name, price, category, volume, seller ownership, and seller-organizer deal-related links.

Important information about products is their supported categorization that will later be used for the sales analysis and can be seen in **Table** 1.1.

Category	Description
Nonalcoholic	Any non-alcoholic beverages (e.g., coffee, water, etc.)
Beer	Any kind of beer
Wine	Any kind of wine
Other Alcohol	Any other kind of alcoholic beverages (e.g., shots, cocktails, etc.)
Salty	Any salty snacks
Sweet	Any sweet snacks
Other	Any other products that do not fit into the above categories

Table 1.1: Product categories

Source: *Author's rendition*

Event Program (approx. 140 rows): Event program schedule with the information about the stages, performers, times and other related information.

Why is it important?

This data provides more context to the event when combined with the financial and customer data above. Enabling the analysis to work with the event program, its correlation with the sales, customer behavior, and other related analysis.

1.3.4 Data Processing Views

To efficiently query the studied data during the analysis, several SQL views and functions were created to simplify and speed up the process.

Transaction Commission Calculation: A function that calculates the commission for each transaction based on the product and the seller-organizer deal. This was a crucial method required to calculate and analyze the commission from event order sales contributing to the organizer's revenue.

Transaction Enrichment: Since the transactional data consisted of several transaction types which were not easily distinguishable, a view was created to enrich the transaction data with the transaction type information. It also benefited from the transaction commission calculation function mentioned above, which helped to easily calculate the commission for each transaction.

Chip Customers: Probably the most complex function that returns the customers at the event. Since the transactional data is architected using Event Sourcing, the customer information is not directly available and needs to be compiled from the transactional history. This function was constructed in a way where it supports time-based filtering and provides extensive insights into the customers, which is shown in **Table** 1.2.

For further understanding, an enumeration of chip types should be mentioned; that can be seen in **Table** 1.3.

1.4 Tools and Technologies

As mentioned earlier, this process drew from a variety of tools and technologies to handle the data and prepare for the analysis. These main tools and technologies included:

- **PostgreSQL**: An open-source relational database management system used to store and query the data[6].
- **PostGIS**: An extension of PostgreSQL that supports geospatial data, enabling spatial analysis and visualization[7].
- **DataSpell**: An integrated development environment (IDE) for data science and analytics, used for database exploration and data visualization[8].
- **Python**: Used for data preprocessing, querying, and analysis, along with libraries like Pandas and Matplotlib for data manipulation[9].
- **Claude AI**: Utilized for extracting data from unstructured sources (e.g., program schedules) and automating repetitive tasks like data entry[10].

[6] `https://www.postgresql.org/`
[7] `https://postgis.net/`
[8] `https://www.jetbrains.com/dataspell/`
[9] `https://www.python.org/`, `https://pandas.pydata.org/`, `https://matplotlib.org/`
[10] `https://claude.ai/`

Column Name	Description
CHIP_ID	Unique chip identifier
CHIP_TYPE	Type of chip (e.g., regular, VIP, online, staff)
REG_AT	Timestamp of chip registration
FIRST_TRX	First transaction timestamp associated with the chip
LAST_TRX	Last transaction timestamp associated with the chip
LAST_BALANCE	Last known balance at the specified time frame
ACTUAL_BALANCE	Balance at the specified time frame after credit refunds
IS_BLOCKED	Indicates if the chip is blocked due to suspicious activity
HOURS_ACTIVE	Total active hours of the chip (daily sum)
DAYS_ACTIVE	Total number of days the chip was active
T_COUNT	Total number of transactions associated with the chip
O_TOTAL_CNT	Total number of orders placed using the chip
O_TOTAL_AMT	Total amount spent through orders
O_MAX_AMT	Maximum amount spent in a single order
O_AVG_AMT	Average amount spent per order
O_MODE_AMT	Most common amount spent per order
OS_AVG_AMT	Average amount spent on sales orders (excluding refunds)
OS_MODE_AMT	Most common sale amount (excluding refunds)
TU_TOTAL_CNT	Total number of top-ups made to the chip
TU_TOTAL_AMT	Total amount credited to the chip via top-ups
TU_MAX_AMT	Maximum amount credited in a single top-up
TU_AVG_AMT	Average amount credited per top-up
TU_MODE_AMT	Most common amount credited per top-up
TU_CARD_BRAND	Used card brand for top-ups (e.g., Visa, Mastercard)
BR_AMT	Total amount refunded to the customer's bank account
BR_EMAIL_DOMAIN	Domain of the refund request email (e.g., gmail.com)
BR_COUNTRY	Country associated with the bank account for refund
BR_REQ_SOURCE	Source of the refund request (e.g., iOS, Android, Web)
BR_BANK_NAME	Name of the Czech bank used for the refund
BR_CREATED	Timestamp when the refund request was created
BR_APPROVED	Timestamp when the refund request was approved
A_EMAIL_DOMAIN	Email domain of the account
A_COUNTRY_NAME	Country associated with the account
A_REQ_SOURCE	Source of the account creation (e.g., iOS, Android, Web)
EO_PAYMENT_METHOD	Payment method for orders (e.g., card, bank transfer)
EO_CARD_BRAND	Used card brand for online order (Visa, Mastercard
EO_REQ_SOURCE	Source of the order request (e.g., iOS, Android, Web)

Table 1.2: Customer chips function return table

Source: *Author's rendition*

Chip Type	Description
Regular	Chip issued at the event without any prior credit
VIP	VIP chip issued with artificial credit on behalf of the organizer
Online	Chip issued via online purchase or top-up
Staff	Chip issued to the staff members
Guest	Chip issued to the guests (band members, etc.)

Table 1.3: Chip types enumeration

Source: *Author's rendition*

Using such tools during this process provided a convenient environment for data handling and processing, initial analysis and ensuring the accuracy and efficiency for further analysis and results presentation.

1.5 Conclusion

This chapter laid out the process of setting up the local environment, collecting and preparing the data, anonymizing sensitive information, and introducing the data structure.

Some key challenges, such as missing beverage volume information, data anonymization, and event program integration, were addressed through a combination of manual and automated processes.

In the end, the use of SQL views for data enrichment and functions for further data processing ensured efficient querying and analysis. And hopefully laid the groundwork for answering the research questions and achieving the study's goals, as detailed in the later chapters.

For better understanding and visualization of this initial comprehensive Knowledge Data Discover process[11], a simplified diagram, which can be seen in the **Figure** 1.1, was created to illustrate the workflow of the process.

[11] Knowledge Data Discovery (KDD) is the process of discovering useful knowledge from a collection of data[8]

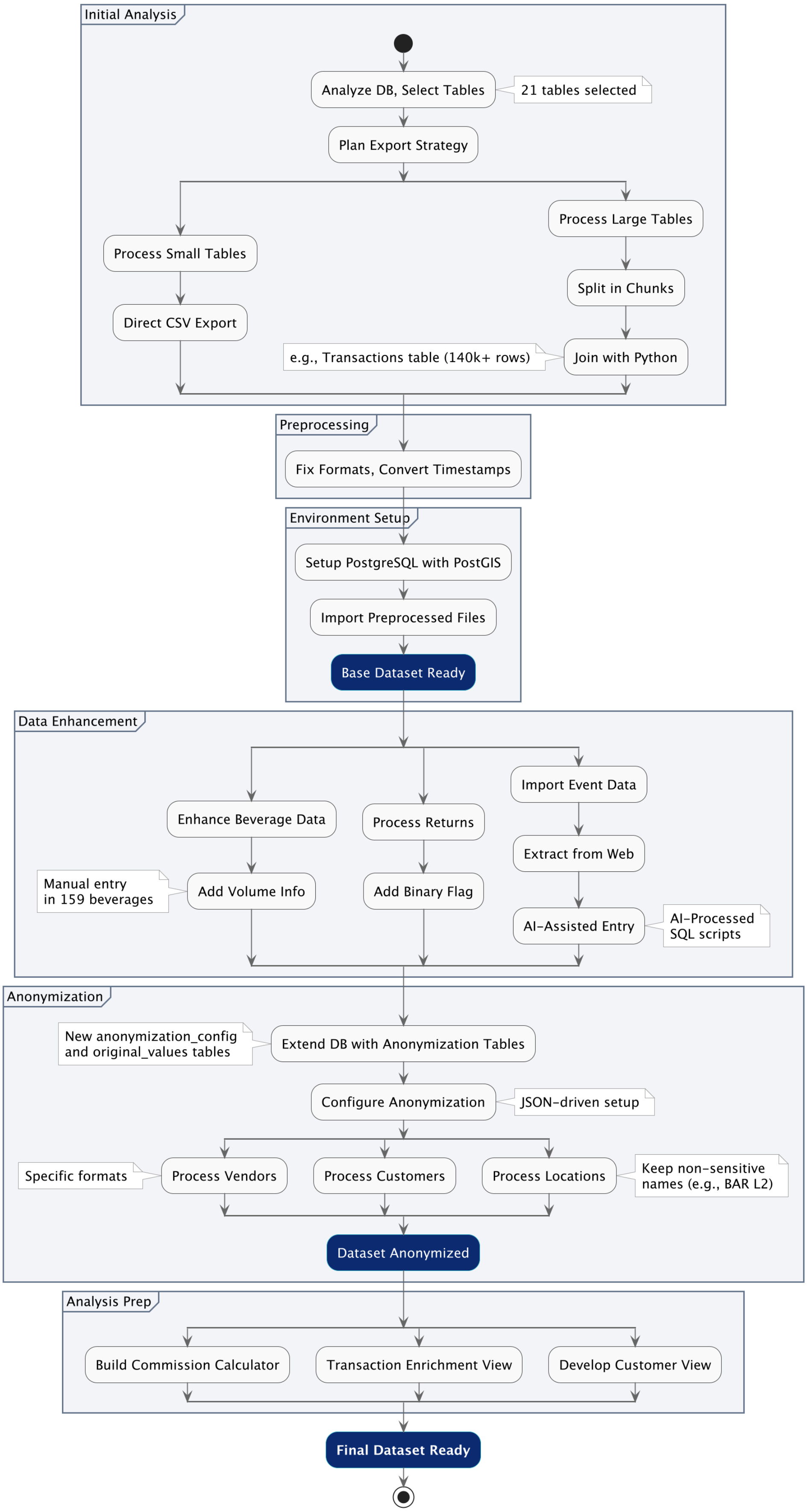


Figure 1.1: Knowledge Data Discovery workflow diagram

2 Data Analysis and Results

This chapter presents the data analysis and results of the research. The goal is to address the research questions outlined earlier, with a focus on providing actionable insights for the event organizer.

The chapter is divided into several sections corresponding to key analytical areas:

1. **Cashflow and Revenue Sources Analysis**
2. **Performance Indicators Analysis**
3. **Beverage Consumption Analysis**,
4. and **Customer Analysis**

Each section focuses on a different aspect of the data analysis trying to answer the research questions, present results, visualizations, and possible interpretations.

2.1 Cashflow and Revenue Sources Analysis

This section provides a comprehensive view of the festival's financial performance and cash flows. It should answer critical questions about how finances were funded into the system, how they were processed, and what were the outcomes.

For this analysis, four questions were previously formulated. However, they were reordered to better fit the narrative of the analysis and logical flow of the chapter.

In the end, this section should provide a clear picture of the financial flows during the event and an easy understanding of the generated revenue from various sources.

2.1.1 Chip Top-Up Analysis

Research Question

RQ2: How much and how was the balance topped up on the chips?

Attendees could top up their chip balances via online prepayments or on-site using cash or card. Additionally, the system allows top-up "artificial" credit for VIP-issued chips, which is also a mean of funding the system. However, these VIP credits are later not refundable, but this will be discussed in the next section.

This subsection quantifies these methods, highlighting their respective contributions to the overall top-up total.

To get the results, it was necessary to find all top-up transactions and their respective payment methods used. This resulted in **17,704** top-up transactions, with a total value of **14,520,973 CZK**.

When looking at the grouping by payment methods, the results in **Chart** 2.1 give a clear picture of the distribution.

Payment Method	Count	Total Value (CZK)
Card terminal	8,486	7,264,503 CZK
Cash	7,561	5,782,570 CZK
Online pre-top-up	1,634	1,436,400 CZK
VIP issued	23	37,500 CZK
Total	**17,704**	**14,520,973 CZK**

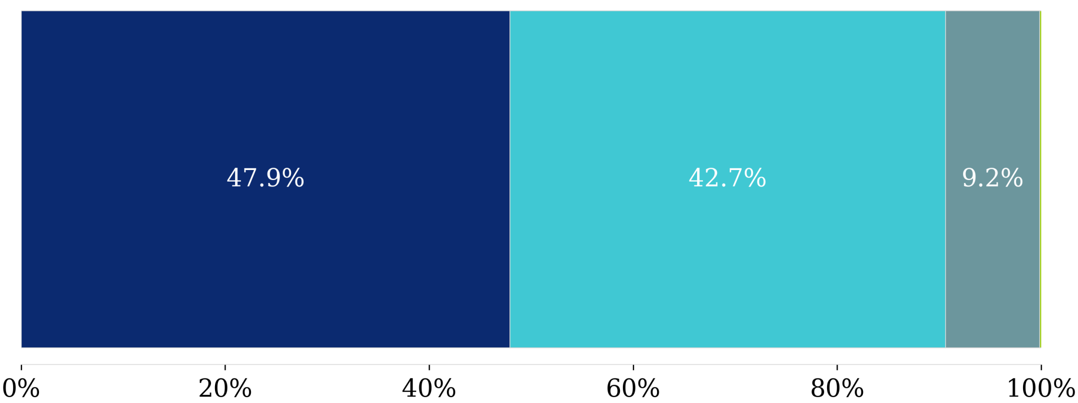


Chart 2.1: RQ2 Top-Up Transactions by Payment Method

Source: *Author's rendition*

Thanks to the results, it is clear how many funds the system received and by what means.

Key Takeaways

- The total top-up amount was **14,520,973 CZK**.
- The most used payment method was the card terminal at the event with 50% of all top-ups.
- Only around 10% of the top-ups were done online.

2.1.2 Sales Analysis

Research Question

RQ4: What was the total sales of the event, how much of it was sold by the organizer, and how many external vendors?

The sales analysis was crucial for understanding the overall sales behavior and served as a basis for further insights tightly connected to the revenue sources.

To answer the research question, it was necessary to find all sales transactions and their respective sellers and to divide them into two groups: the direct organizer's sales and external vendors' sales. And for better understanding, the sales were also grouped by the product categories (see **Table** 1.1 for the list of categories).

The results show that the total sales of the event were **11,711,807 CZK** with the organizer's sales being **8,240,264 CZK** and the external vendors' sales being **3,471,543 CZK**.

The organizer, most importantly, sold all the beer beverages and most of the non-alcoholic and alcoholic (spirits) beverages. Whereas the external vendors sold mainly the food, wine, beverages, and other uncategorized products. This can be seen in the **Chart** 2.2 below.

The organizer also sold not so few of the uncategorized products, which after further investigation turned out to be ticket sales at the event amounting to **684,700 CZK**.

In total, the organizer direct sales were **70%** of the total sales, which is a significant portion, and thus the organizer itself has even bigger influence on the event's financial performance.

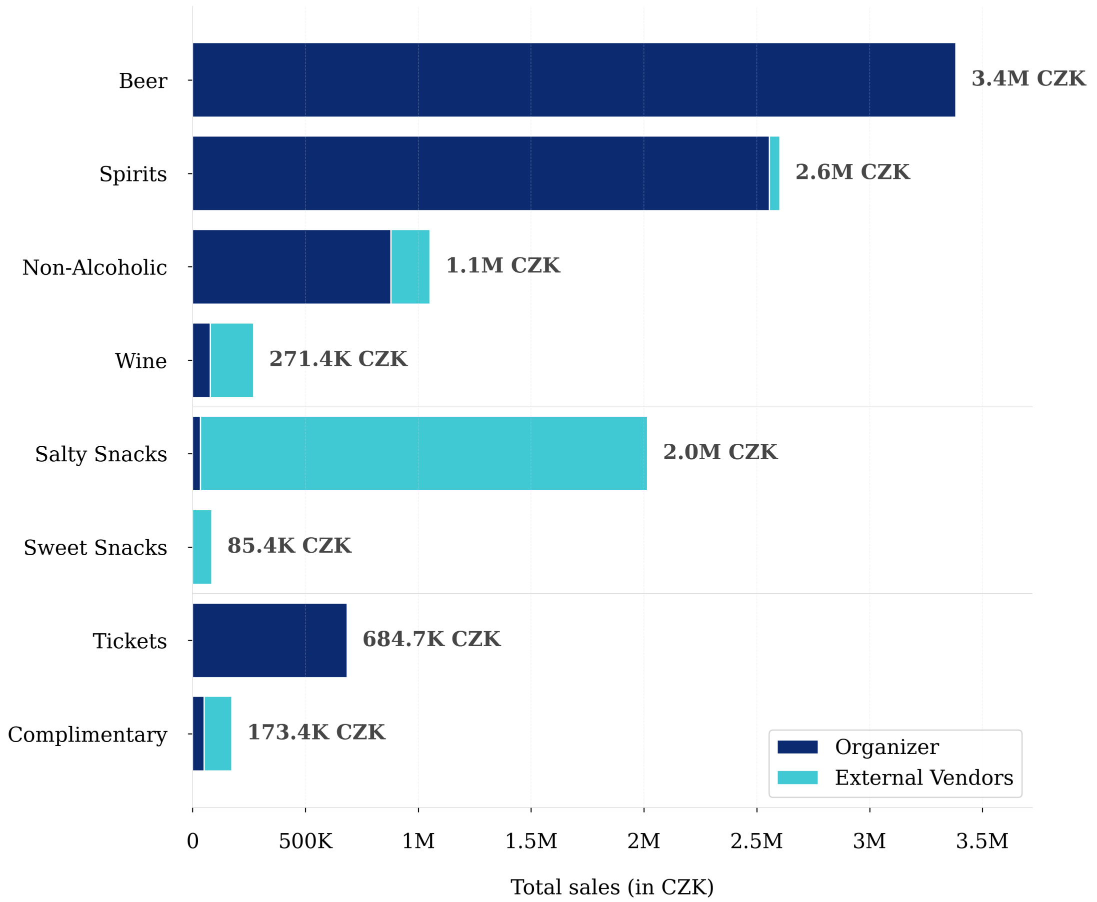


Chart 2.2: RQ4 Sales of the Organizer vs. External Vendors

Source: *Author's rendition*

Key Takeaways

- Total sales of the event were **11,711,807 CZK**, where organizer's sales were **70%** of the total.
- The organizer sold all beer beverages and the majority of the non-alcoholic and alcoholic beverages.
- The organizer also sold tickets at the event amounting to **684,700 CZK**.
- External vendors sold food, wine, and other uncategorized products.

2.1.3 Remaining Chip Balances

Research Question

RQ3: What is the remaining balance on all chips after the event and refunds?

The remaining chip balances are crucial for the event organizer as they represent the potential revenue that can be still claimed. Any unclaimed balances following a specified refund period, typically up to 14 days post-event, will be deemed taxable revenue for the organizer.

Out of the total top-up amount of **14,520,973 CZK**, the total spent credit amounted to **10,984,945 CZK**, which left a total of **3,536,028 CZK** on the chips before refunds. After refunds – done both at the event (**15,379 CZK**) and later via online bank refund requests (**3,163,567 CZK**) – the remaining balance was reduced to **357,082 CZK**.

However, this still included the artificially issued VIP credits with leftover balance of **12,405 CZK**. The system also reported integrity errors in the data, which resulted in a total of **10,246 CZK** due to fraudulent activities performed by some attendees which were automatically suspended by the system.

This left the total unclaimed balance at **334,431 CZK**, which has been claimed by the organizer as taxable revenue.

Since these numbers can be quite abstract, the results in a form of sankey diagram in **Chart** 2.3 below provide an accurate picture of how the money flows.

Thanks to this breakdown, it is clear how the remaining balances were reduced and what was the outcome. These results are important for the last part of this section, which is the total revenue of the organizer.

Key Takeaways

- Total unused credit was **3,536,028 CZK**.
- Credit refunded to customers was **3,178,946 CZK**.
- After VIP issued credits and system integrity error, the unclaimed balance was **334,431 CZK**.

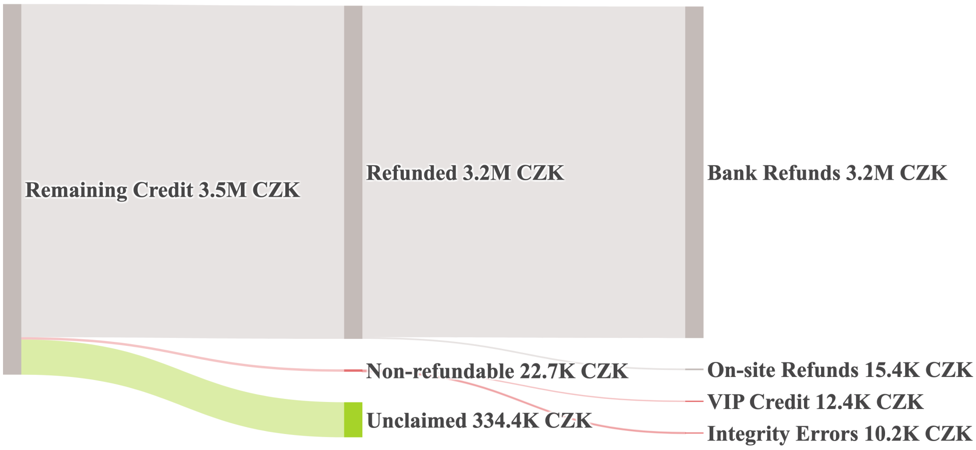


Chart 2.3: RQ3 Remaining Chip Balances Sankey Diagram

Source: *Author's rendition*

2.1.4 Total Revenue of the Organizer

Research Question

RQ1: What was the organizer's total revenue and what did it consist of?

The festival's financial model is based on a combination of revenue streams.

The most important stream is the **commission from the vendor sales**, which is arranged in advance between the organizer and the vendors. The commission is, in this case, a percentage (ranging from 15% to 30% depending on the deal) of the vendor sales amount without VAT.

Therefore, this required finding all sales transactions made at the external vendors' stands and calculating the commission based on the agreed percentage. However, this was not a straightforward task, since a transaction could contain multiple products even from different vendors.

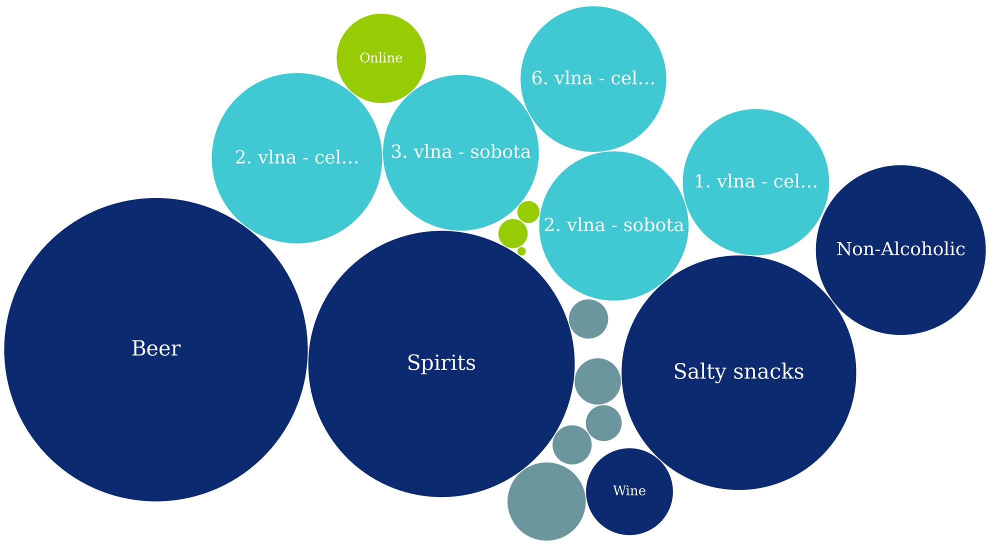
This required a more complex calculation, for which were used the previously mentioned data processing views which were designed for this purpose. In the end, the total revenue from sales commissions was **820,712.79 CZK**.

Another source of revenue is the **unclaimed chip balances**, which, after a credit refund period, are considered as taxable revenue for the organizer. This, thanks to the previous subsection, was found to be **334,431 CZK**.

Currently totalling **1,155,143.79 CZK** is the direct revenue of the organizer from the event. However, given the circumstances and setup of this event, there were also additional, but indirect revenue streams that were not included in the total revenue. These include **the online ticket sales**, which were sold by the organizer and **the direct sales of the organizer**. They were not included in the total direct revenue, as they may misinterpret the results since the analysis lacks expenses of the organizer.

If we were to include these, the total revenue would increase by **11,179,700 CZK** from the online ticket sales and **8,240,264 CZK** from the direct sales, which would result in a total revenue of **20,575,107.79 CZK**.

To better understand the revenue streams, the results are visualized in **Chart** 2.4 below.



Revenue Stream	Amount
Vendor Commissions	820,712.79 CZK
Unclaimed Chip Balances	334,431 CZK
Total Direct Revenue	**1,155,143.79 CZK**
Online Ticket Sales	11,179,700 CZK
Organizer Direct Sales	8,240,264 CZK
Total Revenue (All Streams)	**20,575,107.79 CZK**

Chart 2.4: RQ1 Breakdown of All Revenue Streams

Source: *Author's rendition*

Key Takeaways

- Total direct revenue of the organizer was **1,155,143.79 CZK**.
- Vendor sale commission contributed to approximately 71% of the total direct revenue.
- With other indirect revenue streams, the total revenue would be **20,575,107.79 CZK**.

2.1.5 Summary

This section provided a comprehensive view of the festival's financial performance and cash flows. The results covered the top-up transactions, sales analysis, remaining chip balances, and the total revenue of the organizer and contributed to a better understanding from the financial perspective of the festival.

Nevertheless, results covered in these subsections are only a part of the whole picture and can be interpreted in various ways.

For this particular challenge, a summarized cash flow diagram of evnet payments was created, excluding thus the online ticket sales from the total indirect revenue. This diagram can be seen in the **Chart** 2.5 below.

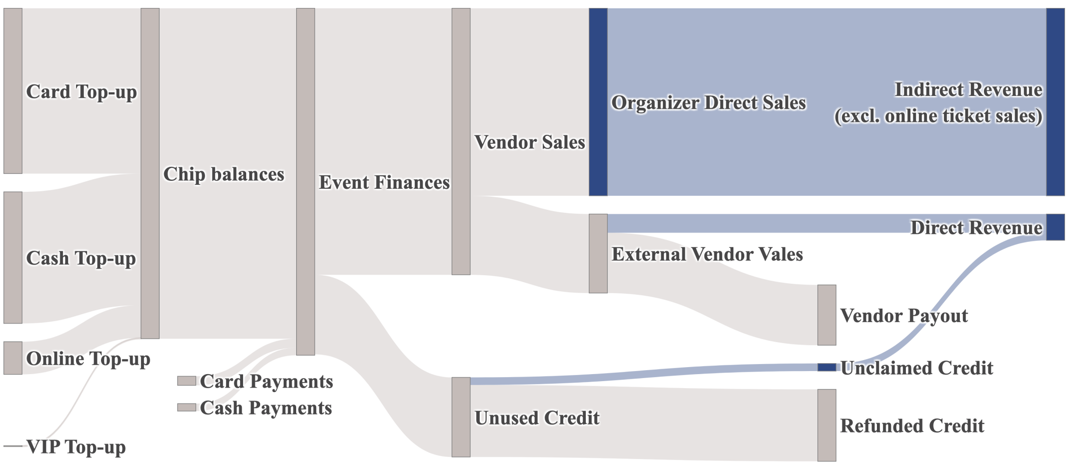


Chart 2.5: Overall Cash Flow Diagram

Source: *Author's rendition*

This diagram provides a clear overview of the financial flows during the festival and nicely summarizes the results of this analysis.

Key Takeaways

- The total incoming money flow was **14,520,973 CZK** from top-up transactions and **726,862 CZK** from non-chip sales.
- Total sales amounted to **11,711,807 CZK**.
- This left a total of **3,536,028 CZK** in unused credit before refunds.
- After refunds and non-refundable chips, the remaining balance left was **334,431 CZK** claimed as taxable revenue.
- Commission from external vendor sales contributed to **820,712.79 CZK** of the total direct revenue.
- Together, the total direct revenue of the organizer was **1,155,143.79 CZK**.

2.2 Performance Indicators Analysis

This section emphasizes the event's performance metrics. The goal is to identify key metrics that can be used to further evaluate the event and its success. The potential of this analysis is to measure the "greatness" and the size of the event in terms of performance.

This analysis aims to provide answers to previously defined research questions about the event's performance. For this analysis, they were slightly reordered and grouped into the two subsections listed below:

1. **Transactions Processing Analysis**
2. and **Best Sale & Top-Up Points, Vendors, and Products Analysis**

The results of this analysis should provide insights into the event's performance and help the organizer to understand the key metrics that can be used to evaluate the event's success.

2.2.1 Transactions Processing Analysis

This subsection will focus on the processing of transactions during the event in pursuit of answering the three first research questions of this section.

> **Research Question**
>
> **RQ5**: How many transactions were processed in total, and when was the system's largest "peak" in transaction volume?

This question actually consists of two sub-questions, which will be addressed separately.

The first part questions the total number of transactions processed during the event. This was actually quite simple to answer, as the system was designed to track all transactions and their types. The resulting total number of transactions was **141,381** consisting of **110,854** sales transactions, **17,726** top-up transactions and **12,801** chip register transactions.

The second part focuses on rather time-related metrics and asks about the processing peak times during the festival. For this part, it was necessary to spread out the above transactions over time and find the peaks.

The results in the **Chart** 2.6 below show the distribution of the processed transactions over time. It clearly identifies the peak on the last day of the festival at 18:00, amounting to **8,986** transactions.

> **Key Takeaways**
>
> - Total number of transactions processed was **141,381** consisting mainly (**78**%) of order transactions.
> - The peak of transactions was on the last day of the festival at 18:00 with **8,986** transactions processed at that hour.

The following two questions (**RQ6** and **RQ7**) focus on processing times and potential delays during the event.

> **Research Question**
>
> **RQ6**: What was the average transaction processing time during peak hours?

The answer to this question is closely related to the previous one, as it requires the identification of the processing times during the peak times, which were already

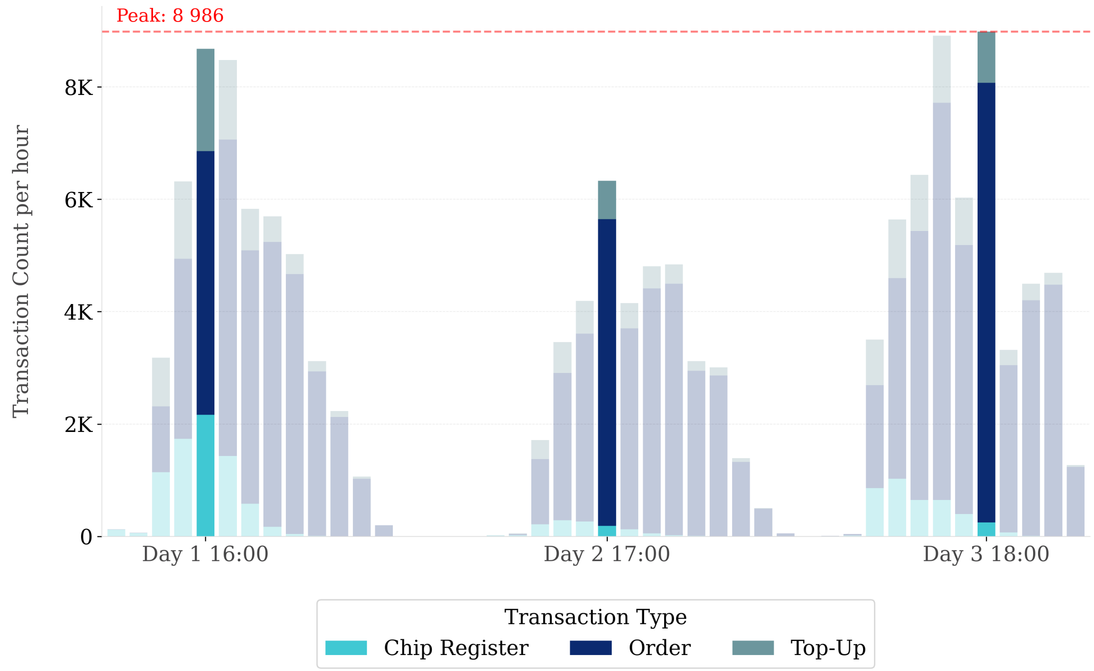


Chart 2.6: RQ5 Transactions Peaks

Source: *Author's rendition*

identified.

It required finding the average processing time, meaning the difference between the transaction creation and its completion times.

> ⓘ What causes the processing time?
>
> The time when the transaction is created is the time when the in-place offline-supported system created the transaction, and the processed time is later when the central system receives the transaction and processes it.
> The delays can be caused by various factors, such as network latency, offline mode active, system load, or even the transaction type.

The results show that the average processing time during the peak times was approximately **47** seconds.

When slightly changing the displayed data, we also get the answer to the **RQ7** about the potential delays and downtimes during the event.

> Research Question
>
> **RQ7**: Were there any significant delays or downtimes in processing transactions?

The chart in the **Chart** 2.7 shows the distribution of the processing times over the time and identifies one high processing peak of approximately **11** minutes. This is highly unusual and indicates a vendor's misuse of the system or accidentally put the system into offline mode.

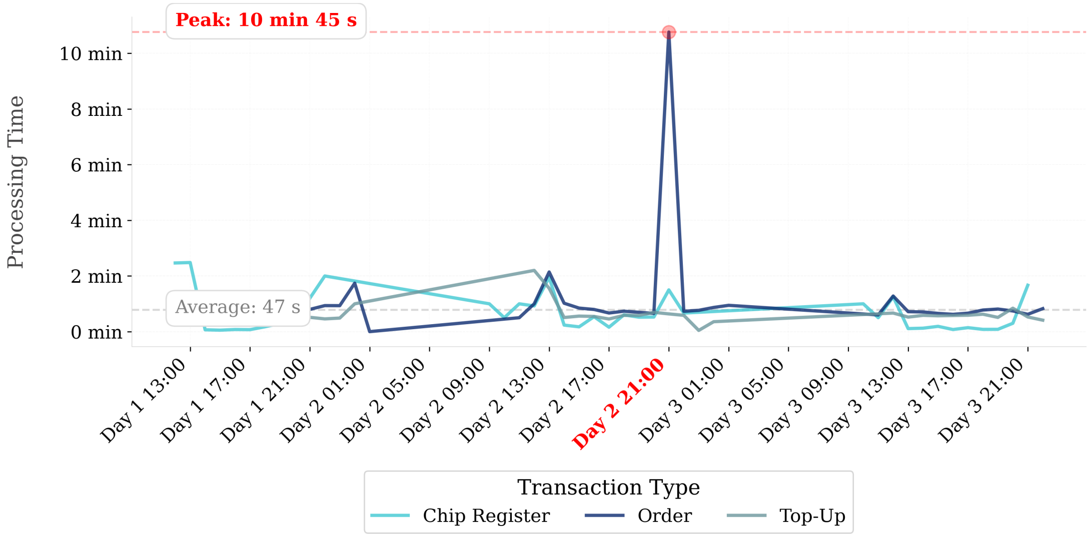


Chart 2.7: RQ7 Transaction Processing Times

Source: *Author's rendition*

Other high peaks are visible on the second day in the afternoon with approximately **3** minutes processing time, which was probably caused by the initial load on that day.

Key Takeaways

- The average processing time during the peak times was **47** seconds.
- The highest processing peak was approximately **11** minutes, indicating a potential misuse of the system.
- Other high peaks were visible on the second day in the afternoon with approx.˜3 minutes processing time.

These results provide insights into the system's performance during the event, its reliability, and potential bottlenecks. It also shows the festival's popularity and the system's ability to handle the load.

2.2.2 Best Sale & Top-Up Points, Vendors, and Products Analysis

In this subsection, the main goal will be to address the last four research questions of this section and provide insights into the best: selling points, top-up points, vendors, and products.

The problem with these question statements is that they are quite broad and can be interpreted in various ways. What does a "best" mean in this context?

It can be the most profitable, the best rated, the most visited, etc. But since we are exploring the performance indicators, the best should be understood as the "busiest". In terms of the system and this analysis, it should mean **the most transactions created** and possibly the point's ability to handle the load.

Best Top-Up Points

The first focus will be on the best top-up points since unlike the selling points, vendors, and products, the top-up points are not linked to any specific product or vendor.

> Research Question
>
> **RQ9**: What were the best top-up points?

To find these results, it required finding all top-up transactions, aggregating them in a bucket-like time frame and finally calculating their total counts, max peaks, and averages over time.

This resulted in the following findings in the **Table** 2.1 below.

This indicates that the most busy top-up points were somehow evenly distributed with approximately around **1000 transactions** processed during the event with average peaks of around **100 transactions/hour**. The least busy top-up points were the specific ones, such as the Support tent, VIP, and Accreditation points, which were not used so much for top-ups.

The overall distribution, shown in the **Chart** 2.8, also shows that Top-up points were more busy than Check-in points. That makes sense because the top-ups were done more frequently than the initial check-ins, but the check-ins were done

	Top-Up point	Customers	Transactions	Max trx./h
1	Pokladna 16	1,119	1,139	99
2	Pokladna 2	1,103	1,108	106
3	Pokladna 15	1,035	1,043	82
4	Pokladna 4	1,007	1,017	107
		...		
15	Pokladna 7	740	744	73
16	Pokladna 9	699	711	72
17	Pokladna 10	629	640	69
18	Odbavení	529	529	125
		...		
25	Odbavení 5	198	198	46
26	Odbavení 6	191	191	32
27	Support	64	85	11
28	VIP	23	23	5
29	Akreditace	17	17	8

Table 2.1: RQ9 Best Top-Up Points

Source: *Author's rendition*

in a more concentrated time frame.

Especially **Odbavení** point processed only **529** transactions but peaked at **125** transactions per hour, which was much higher than the other check-in points and even higher than the best top-up points.

Key Takeaways

- The most busy top-up points are processed around **1,000** transactions during the event.
- The average peak of the top-up points was around **100** transactions per hour.
- The least busy top-up points were the specific ones, such as the Support tent, VIP, and Accreditation points.
- Check-in points were less busy than the top-up points, but the **Odbavení** point peaked at **125** transactions per hour.

Best Sale Points

The next focus will be on the best sale points, which are the points where the most orders were created.

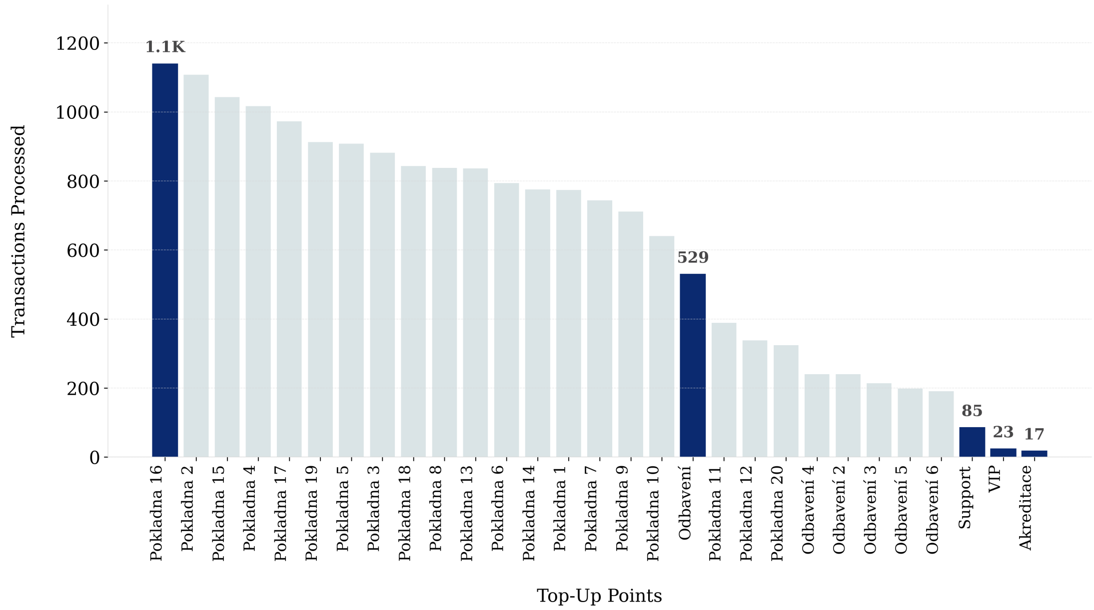


Chart 2.8: RQ9 Best Top-Up Points

Source: *Author's rendition*

Research Question
RQ8: What were the best sale places?

The process of finding the best sale points was similar to the previous one, but this time it required finding all sales transactions and their respective points.

Out of the total of **145** sale points, the best place was undeniable the **L20 PIVNÍ STAN 1** with total **10,114** orders processed and the maximum peak of **840** orders per hour.

Another interesting fact is the number of unique users processed at the places. The **L20 PIVNÍ STAN 1** processed **9,159** unique customers, accounting for a significant portion (**91.50**%) of the total.

Based on this particular finding, we can assume that in the following analysis - the best vendors and products - the product preferences will be highly in favor of the beer beverages. And thus the best vendor will probably be the organizer, as they sold all the beer beverages at the festival.

	Sale Point	Customers	Orders	Max orders./h
1	L20 PIVNÍ STAN 1	9,159	10,114	840
2	Place 78	4,914	5,113	479
3	A19 PIVO 1	4,615	5,073	608
4	L16 FRISCO	3,420	3,646	235
5	Place 11	3,078	3,226	265
6	L22 BEEFEATER/ HAVANA 1	2,613	2,857	317
7	A19 VÝKUP KELÍMKŮ	2,683	2,743	316
...				
132	Support	23	23	4
133	Taxi 51	11	21	4
134	Taxi 50	5	8	4

Table 2.2: RQ8 Best Sale Points

Source: *Author's rendition*

Key Takeaways

- The best sale point was the **L20 PIVNÍ STAN 1** with total **10,114** orders processed, which was **9.12**% of the total orders created.
- The best sale point peaked at **840** orders per hour.
- The **L20 PIVNÍ STAN 1** processed **9,159** unique customers, which was **91.50**% of all active customers.

In conclusion to these two questions, the results show clearly the busiest points of the festival and their ability to handle the load. However, the results can be visualized in a more interactive way, which would provide a better understanding of the data.

One especially interesting visualization of the best sale and top-up points would be a heatmap of the festival area with the points and their respective transaction counts. As this was initially intended to be part of the analysis, it was unfortunately not possible to create it due to the lack of the necessary data.

Best Vendors

To analyze the best vendors, in terms of performance, the same approach as with the sale points is used.

Research Question

RQ10: Who were the best vendors?

The results in the **Table** 2.3 below show the distribution of the processed orders over the vendors.

	Vendor	Customers	Orders
1	Organizer	9,831	89,217
2	Seller 05	3,632	5,113
3	Seller 15	1,571	2,282
4	Seller 13	1,533	1,812
5	Seller 29	915	1,623
6	Seller 28	530	1,468
7	Seller 08	1,118	1,359
...			
25	Seller 02	30	31
26	Seller 09	26	27
27	Seller 10	4	5

Table 2.3: RQ10 Best Vendors

Source: *Author's rendition*

As predicted in the previous section, the best vendor was the organizer, which processed the most orders and customers. The total of **89,217** orders was processed by the organizer, which was **80.04**% of the total orders created and served **9,831** unique customers, which was **98.22**% of all active customers.

The second-best vendor, out of **27** total, was some **Seller 05** with about **84,104** orders less than the organizer.

In these results, we did not go after the hourly maximums, as in the previous questions, as the vendors were not time-bound and could have been at multiple places simultaneously. The results would not be so relevant and would not provide any additional insights.

Key Takeaways

- The best vendor was the organizer, which processed **89,217** orders (**80.04**% of the total orders).
- The best vendor served **9,831** unique customers (**98.22**% of all active customers).
- The second-best vendor was behind around **84,104** orders less than the best vendor.

Best Products

The last focus will be on the best products, which are the products that were sold the most during the event.

Research Question

RQ11: What were the best products?

Previously, the best products were predicted to be the beer beverages, as the organizer sold all the beer beverages at the festival.

The results in the **Table** 2.4 below somehow confirm this prediction as the very best product was a returnable cup – **Kelímek – záloha** and the next four best products were beer beverages:

	Product	Sales	Refunds	Customers
1	Kelímek - záloha	17,933	11,215	8,729
2	Radegast 12°	12,844	26	3,760
3	Hladinka - Prazdroj	9,106	21	2,958
4	Radegast 10°	7,662	12	2,846
6	Birell	4,367	6	2,321
...				
326	Ponožky	1	0	1
327	Churros přímé	0	0	4
328	Radegast 10	0	0	9

Table 2.4: RQ11 Best Products

Source: *Author's rendition*

As there were more than **300** unique products sold during the event, a better visualization of the results would be a bar chart showing the best product categories instead of individual products. This can be seen in the **Chart** 2.9 below.

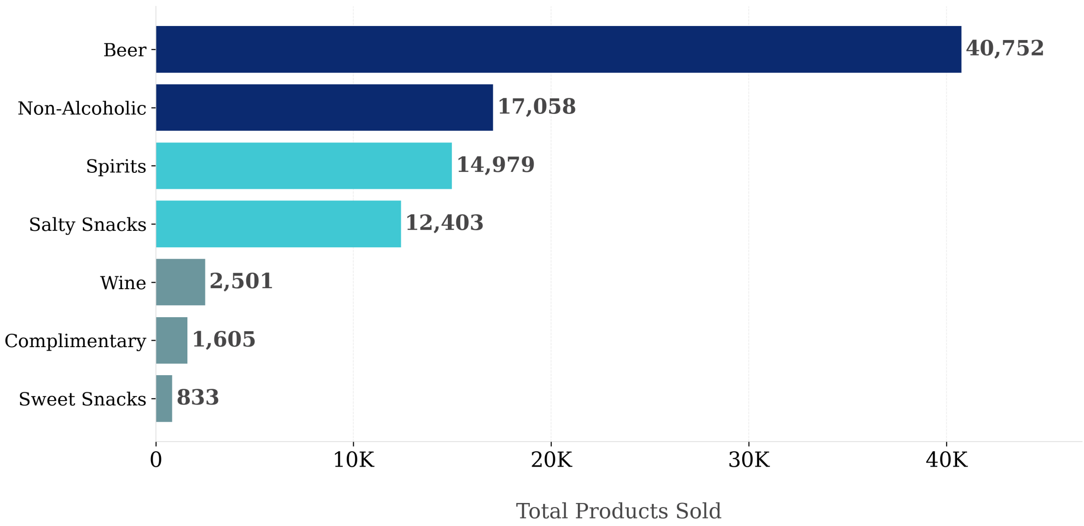


Chart 2.9: RQ11 Best Products by Category

Source: *Author's rendition*

This chart now confirms the prediction about the beer beverages, as the **Beer** (Pivo) category was the most sold during the event with a little more than **40,000** orders processed.

Key Takeaways

- The best product was a returnable cup followed by beer beverages.
- Prediction about the beer beverages was confirmed, as the **Beer** (Pivo) category was the most sold during the event.

This last analysis provided insights into the best products, which confirmed the previous predictions and will now serve as a basis for the next section, where the beverage consumption will be analyzed.

2.2.3 Summary

Thanks to this section, the performance indicators of the event were analyzed, and the key metrics were identified. The results analyzed the transactional processing performance, identified several peak points during the event, and provided insights into the best sale points, top-up points, vendors, and products. Providing a better understanding of the event's performance and giving more context for the next analysis dealing with the beverage consumption.

2.3 Beverage Consumption Analysis

This section provides a detailed analysis of beverage consumption, which was previously determined to be the most important aspect of the event.

It should provide insights into overall consumption, as well as detailed information on returnable cups, the most popular beverages, and the leading beverage brands, while also answering the previously stated questions.

This section will address the next six research questions, after a minor logical reorganization into three groups:

1. **Returnable Cups Analysis**
2. **Total Consumption Analysis**
3. and **Popular Brands Analysis**.

2.3.1 Returnable Cups Analysis

Due to the little alteration of the local database and the previously referenced data in **Subsection** 1.1.3, it became possible to monitor the returnable cups and their associated transactions.

This capability, previously absent, should enhance comprehension of product sales and the utilization of returnable cups.

Research Question

RQ13: How many returnable cups were issued and returned/not returned?

To chart out the results, it was essential to analyze the actual contents of the transactions rather than merely identifying transactions including returnable cups, as a single transaction may encompass many products and hence multiple returnable cups. The organizer sold the cups for a price of 70 CZK with a 21% VAT.

Upon calculating the total number of issued and returned cups, the results depicted in **Chart** 2.10 below illustrate the distribution of returnable cups throughout the event.

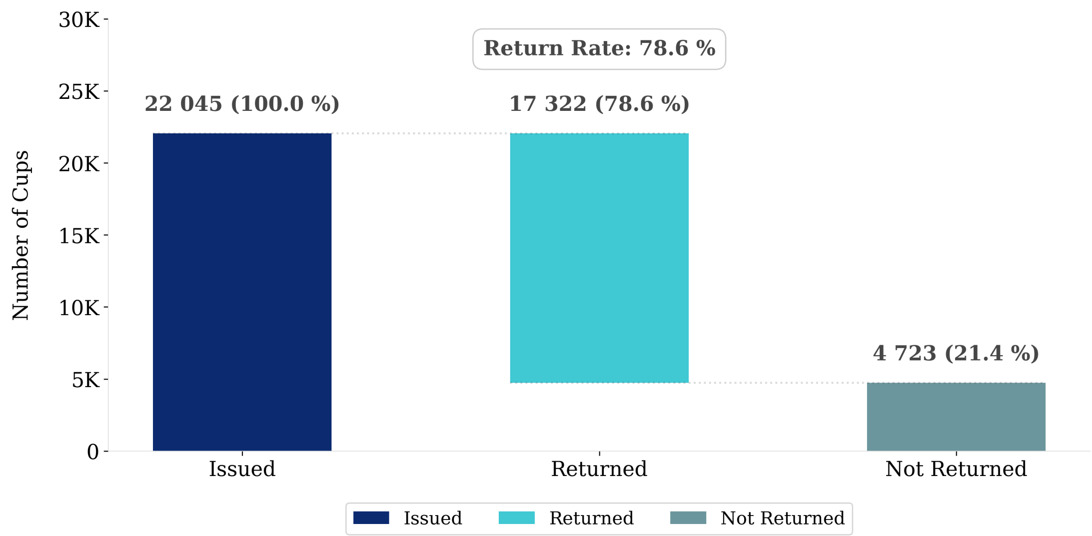


Chart 2.10: RQ13 Returnable Cups

Source: *Author's rendition*

The data indicate that a total of **20,045** cups was distributed during the event, with only **17,322** cups returned, yielding a return rate of **78.60**%.

This, however, does not imply that the remaining **4,723** cups were lost or regarded as a loss, as the cups were purchased. Most unreturned cups can be presumed to have been retained as souvenirs.

Key Takeaways

- The total of **20,045** cups were issued during the festival.
- Only **17,322** cups returned resulted in a **78.60**% return rate.

2.3.2 Total Consumption Analysis

This focuses on the overall beverage consumption during the event. Again, thanks to local database modifications, it was possible to track the beverage consumption easily, as each product now had a volume attribute in milliliters.

Research Question

RQ12: What was the total amount of beverage consumed?

To find the results was quite straightforward, as it only required summing up the volumes of all sold products.

However, to present the results, it was convenient to group the products into categories and show the total consumption of each category and thus answering also the next question **RQ14** .

Research Question

RQ14: What was the most popular beverage category?

These results are shown in the **Table** 2.5 below and show a total of **29,342** liters of beverages consumed during the event, with the beer category being the most consumed.

Beverage Category	Volume	Ratio
Beer	19,797 l	67.46 %
Non-alcoholic	6,987 l	23.81 %
Spirits / other alcohol	1,992 l	6.78 %
Wine	575 l	1.95 %
Total volume	**29,342** l	100 %

Table 2.5: RQ12 Total Beverage Consumption

Source: *Author's rendition*

These results serve as a basis for the next questions, which focuses on the most consumed beverage brands rather than product categories.

Key Takeaways

- A total of **29,342** liters of beverages were consumed during the event.
- The beer category was the most consumed with **19,797** liters consumed.

2.3.3 Popular Brands Analysis

This section explores beverage preferences, concentrating on the most popular brands within the leading categories:

- **Beer Brands Analysis**
- **Non-alcoholic Brands Analysis**,
- and **Alcoholic Brands Analysis**.

Answering these issues required the identification and categorization of all beverage products, followed by the computation of their overall consumption. However, this was not entirely easy, as the products were not uniformly labeled. This indicated that beers labeled as “Radegast 10” and “Radegast 12” were not categorized together, resulting in biased outcomes.

To address this issue, the products ought to be categorized by their brand rather than by their name. This methodology seemed logical; nevertheless, the data lacked brand information, containing simply the product name.

A systematic approach would be to extend the database with brands and back-fill the products with a link to the brand. This would require a significant amount of work and time, which was not available at the time of this analysis.

A more straightforward and manual method was used, wherein the product's “brand” was identified by extracting the essential parts of the product name, eliminating redundant elements such as volume, beer grade, or other details. This helped produce better results, however, still lack complete accuracy.

Beer Brands Analysis

Starting with the most consumed and most popular category – beer.

> Research Question
>
> **RQ15**: What was the top beer brand, and how much was consumed and sold?

The results in the **Chart** 2.11 below illustrate the distribution of the most frequently consumed beer brands during the event.

This distribution indicates that the most consumed beer brand was **Radegast**, with a total consumption of **10,366** liters and **27,329** units sold.

The second was **Pilsner Urquell**, which had half the consumption of **Radegast**, followed by the **Frisco** and **Proud** brands. A complete list of all ten brands from the festival is shown in the **Table** 2.6 below.

The data demonstrate a strong preference for Radegast beer among festival attendees. The price may have influenced this, as the Radegast was generally slightly less expensive than, for instance, the Pilsner Urquell.

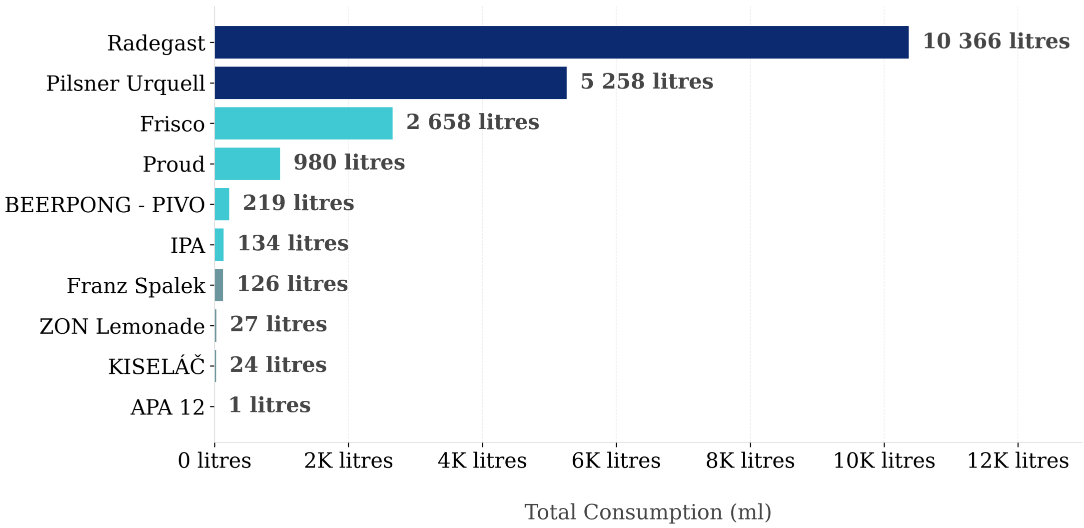


Chart 2.11: RQ15 Most Consumed Beer Brands

Source: *Author's rendition*

Key Takeaways

- The most consumed beer brand was **Radegast** with a total of **10,366** liters consumed.
- The second most consumed beer brand was the **Pilsner Urquell** with half the consumption of the Radegast.

Non-alcoholic Brands Analysis

This next section is bound to the non-alcoholic beverages, which were the second most consumed category.

Research Question

RQ17: What was the most popular non-alcoholic beverage brand, and how much was sold?

Aggregated similarly as in the previous analysis, the results in the **Table** 2.7 below show the distribution of the most consumed non-alcoholic brands during the event.

The results indicate a close rivalry between the **Birell** and **ZON Lemonade** brands, with Birell emerging as the most consumed non-alcoholic beverage at the event. A total of **6,971** liters drank and **20,487** units sold established it as the most popular non-alcoholic beverage, commanding a **31.35**% share of the total

	Beer brand	Total consumption	Units sold
1	Radegast	10,366 l	27,329
2	Pilsner Urquell	5,258 l	14,137
3	Frisco	2,658 l	8,209
4	Proud	980 l	2,908
5	BEERPONG - PIVO	219 l	681
6	IPA	134 l	332
7	Franz Spalek	126 l	285
8	KISELÁČ	24 l	55
9	ZON Lemonade	27 l	51
10	APA 12	1 l	3
	Total	**19,793** l	**53,990**

Table 2.6: RQ15 Most Consumed Beer Brands

Source: *Author's rendition*

	Non-alcoholic brand	Total consumption	Units sold
1	Birell	2,186 l	4,893
2	ZON Lemonade	1,878 l	4,508
3	Water	984 l	2,855
4	Redbull	862 l	3,973
5	Lemonade	791 l	2,063
6	Coffee	204 l	1,855
7	Tea	54 l	153
8	Others	12 l	187
	Total	**6,971** l	**20,487**

Table 2.7: RQ17 Most Consumed Non-alcoholic Brands

Source: *Author's rendition*

consumption.

Key Takeaways

- The most consumed non-alcoholic brand was **Birell** with a total of **2,186** liters consumed.
- The second most consumed non-alcoholic brand was the **ZON Lemonade**.

Alcoholic Brands Analysis

The last part of this section focuses on other alcoholic beverages, such as spirits, shots, cocktails, and other alcoholic beverages.

Research Question

RQ16: What was the most popular brand of other alcoholic beverages, and how much was consumed and sold?

This analysis consisted of a total of 23 different brands consisting mostly of spirits and shots. The results presented in this section may be more biased than the previous ones, as the brand determination was more challenging due to the lack of consistent labeling. Moreover, previously it was necessary to classify the products with the volume information, which is not so straightforward for shots and spirits.

The results in the **Table** 2.8 indicate that the most consumed alcoholic brand was the **Beefeater** with a total of **966** liters consumed and **4,628** units sold. However, the second most consumed brand was **Absolut Vodka** with only **335** liters consumed, but with a much higher units sold count of **9,177**.

Key Takeaways

- The most consumed alcoholic brand was the **Beefeater** with a total of **966** liters consumed.
- The second most consumed alcoholic brand was the **Absolut Vodka** with a total of **335** liters consumed.

	Alcoholic brand	Total consumption	Units sold
1	Beefeater	966 l	4,628
2	Absolut Vodka	335 l	9,177
3	Jack Daniels	312 l	2,391
4	Becherovka Lemond	96 l	4,941
5	Havana	67 l	1,847
6	Malibu + Pineapple	55 l	707
7	Malfy Rosa + Tonic	45 l	217
8	Fire Štrúdl	28 l	770
9	Aperol	17 l	414
10	Others	59 l	1,872
	Total	**1,980** l	**26,964**

Table 2.8: RQ16 Most Consumed Alcoholic Brands

Source: *Author's rendition*

2.3.4 Summary

Insights into beverage consumption, covered in this section, should provide a better understanding of the overall preferences and total consumptions during the festival. As this data was not previously available, it was a valuable addition to the analysis and will play a significant role when presenting the results to the festival organizers.

The results showed the total consumption of all beverages, the insightful view on returnable cups usage, and the most consumed beverage brands in the most popular categories.

With the data available, there are still many more questions that could be answered, particularly in combination with customer data. However, clear bounds were set for this analysis to answer the most important questions asked by the festival organizers.

2.4 Customer Analysis

This section will focus on customer analysis, which should provide interesting insights into festival attendees' behavior and segmentation.

Because the organizers lacked deeper insights into festival attendees, this analysis should contribute significantly to a better understanding of the event.

The analysis addresses the remaining research questions, which have been reorganized into four logical groups to improve the narrative flow:

1. **Event Attendance and Timeline Analysis**
2. **Customer Segmentation**,
3. **Payment Behavior Analysis**
4. and **Purchase Pattern Analysis**.

2.4.1 Event Attendance and Timeline Analysis

This section should provide information about attendees' initial behavior, total attendance throughout the festival, and later top-up behavior.

Total Attendance and Daily Activity

Research Question

RQ18: What was the total attendance at the event, and how many active customers were there each day?

The analysis shows that the festival drew total **10,009** unique customers over the three-day period. However, the daily attendance figures show interesting patterns in how these customers were distributed throughout the festival days.

Looking at the daily active customers:

- Day 1 (Thursday): **6,214** active customers
- Day 2 (Friday): **5,832** active customers
- Day 3 (Saturday): **8,066** active customers

These numbers indicate that many customers attended multiple days of the festival, as the sum of daily attendees (**20,112**) is significantly higher than the unique customer count. This was expected, as the festival was designed to attract visitors for multiple days.

The attendance peaked on the final day of the festival, with Day 2 showing slightly lower attendance than the opening day. The significant increase in attendance on Day 3 suggests that the festival successfully attracted weekend visitors.

Key Takeaways

- Total unique customers over the three days: **10,009**.
- The highest attendance was on Day 3 (Saturday) with **8,066** active customers.
- The festival maintained consistent attendance on weekdays (Days 1–2).
- Significant increase in attendance (38% higher than Day 2) for the weekend (Day 3).

Visitor Arrival Patterns

Research Question

RQ24: What was the course of the event in terms of new visitors? And when were the largest "peaks"?

The analysis of visitor arrival patterns reveals distinct peak periods across the three festival days, with significant variations in the rate of new visitor registrations throughout each day.

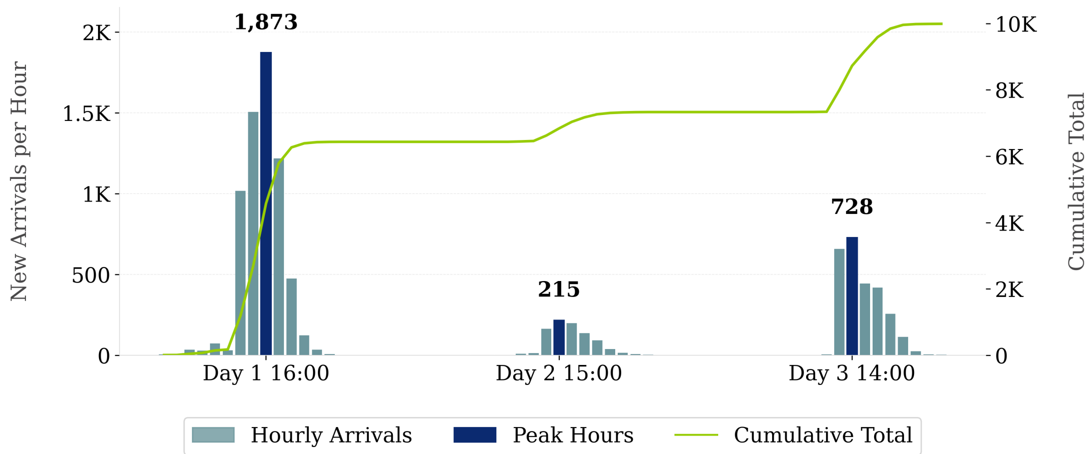


Chart 2.12: RQ24 Visitor Arrival Patterns

Source: *Author's rendition*

The data reveals three distinct patterns across the festival days:

- **Day 1 (Thursday)**: A rapid peak of approximately **1,873** new visitors was observed during the 16:00 hour, which was the time of the highest arrival activity. The day showed a clear pattern of increasing arrivals from 14:00 to 16:00, followed by a gradual decline resulting in **6,433** new visitors.
- **Day 2 (Friday)**: A modest peak of **215** visitors at 15:00, but significantly lower new arrivals than on Day 1. This decrease in the number of new arrivals was expected, as a significant number of visitors had already completed the registration process on Day 1.
- **Day 3 (Saturday)**: The weekend attendees' influx was clear in the resurgence of new arrivals, which reached a peak of **728** new visitors at 14:00.

The data indicates that the majority of visitor arrivals occurred during the afternoon hours (14:00–17:00) on all days. Minimal new arrivals were consistently observed during the early morning hours (before 10:00) and late evening hours (after 20:00).

Key Takeaways

- Highest single-hour registration peak: **1,873** visitors (Day 1, 16:00).
- The registration window that was most effective was from 14:00 to 17:00 on all days.
- Day 1 accounted for approximately **64.39**% of total arrivals.
- Weekend (Day 3) saw renewed registration activity with the peak of **728** new visitors.

Time to First Transaction

Research Question

RQ25: What is the average time of a visitor from arrival to first transaction?

The analysis reveals significant variations in how quickly different types of visitors made their first transaction after arrival. Overall, the average time to the first transaction was **80** minutes. However, this number alone is misleading, as evidenced by the much lower median of **7** minutes and mode of **3** minutes.

Visitor Type	Average	Mode	Median	Max
Guest	**2** h **21** min	**0** min	**10** min	**52** h **56** min
Regular	**45** min	**0** min	**6** min	**47** h **32** min
Online	**67** min	**3** min	**7** min	**44** h **51** min
Staff	**9** h **44** min	**7** min	**3** h **54** min	**58** h **12** min
Overall	**80** min	**3** min	**7** min	**58** h **12** min

Table 2.9: RQ25 Time to First Transaction

Source: *Author's rendition*

Breaking down the analysis by visitor type[1], reveals distinct patterns:

- **Regular**: Quickest to transact, with an average of **45** minutes and a median of only **6** minutes
- **Online**: Similar efficiency with an average of **67** minutes and a median of **7** minutes
- **Guests**: Took longer, averaging **141** minutes with a median of **10** minutes
- **Staff**: Showed significantly different behavior, with an average of **584** minutes and a median of **234** minutes

The significant difference between average and median times across all categories suggests a right-skewed distribution, implying that while most visitors completed their first transaction quickly, others took much longer. This was especially true for regular attendees, who would sometimes arrive early to check in and then return later to make their first purchase.

Staff members who were not there primarily to consume, on the other hand, exhibited a different behavior pattern, with their first transaction occurring later. This was most likely due to their responsibilities at the event, which may have prevented them from making purchases during working hours.

[1] Visitor or rather, chip types are described in the **Chip types enumeration** table.

Key Takeaways

- Most visitors (as indicated by the median) made their first transaction within **7** minutes of arrival.
- Regular visitors were the most efficient, typically transacting within **6** minutes.
- Staff members showed distinctly different behavior, likely due to their different roles at the event.
- The large gap between mean and median times suggests some visitors waited significantly longer than others before their first transaction.

Credit Top-up Patterns

Research Question

RQ26: What was the course of the event in terms of topping up credit on-site? And when were the largest "peaks"?

The examination of credit top-up patterns reveals distinct peak periods and daily patterns throughout the event. The system processed **17,233** top–up transactions, with activity varying significantly throughout the day.

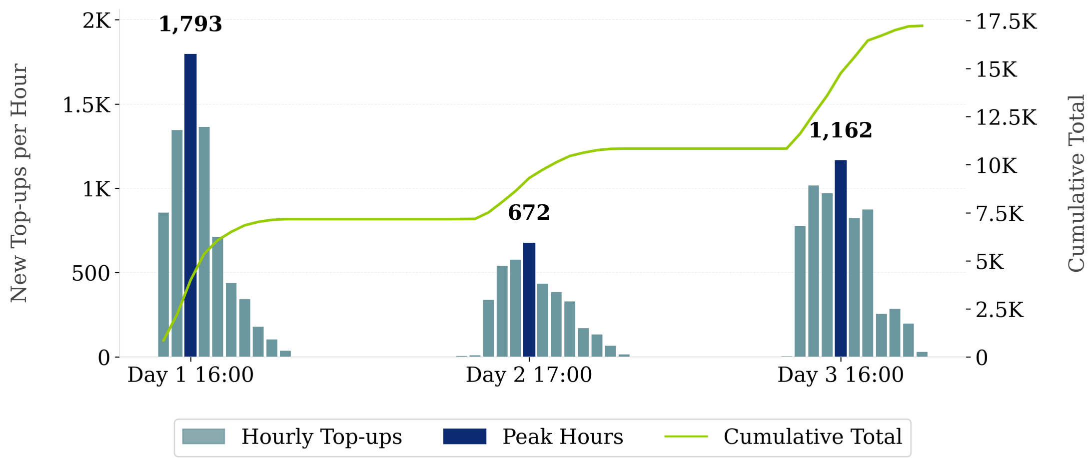


Chart 2.13: RQ26 Credit Top-up Patterns During Event

Source: *Author's rendition*

The data shows three major peak periods:

- **Day 1 Peak**: The highest single-hour volume occurred at 16:00 with

1,793 top-ups

- **Day 2 Peak**: A moderate peak at 17:00 with **672** top-ups
- **Day 3 Peak**: A strong resurgence with **1,162** top-ups at 16:00

Each day had a similar pattern, with activity increasing in the early afternoon, peaking in the late afternoon (between 16:00 and 18:00), and gradually decreasing into the evening. The overnight hours (00:00–12:00) saw minimal top-up activity.

Key Takeaways

- Peak top-up activity consistently occurred during late afternoon hours.
- Day 1 saw the highest single-hour volume with **1,793** top-ups.
- Clear daily pattern of afternoon peaks and overnight lulls.
- Day 3 showed sustained high activity with multiple hours exceeding **800** top-ups.

2.4.2 Customer Segmentation

This section analyzes the distribution of customer types and their digital service adoption patterns, providing insights into how different groups of attendees interacted with the system.

Research Question

RQ19: How many customers topped up their credit in advance online?

Out of **8,974** customers who got their chips through online ticket purchases, only **1,630** (**18.20**%) took advantage of the advance credit top-up option. This indicates that while many customers purchased tickets online, most preferred to top up their credit on-site.

Research Question

RQ20: What was the distribution of customers by type (on-site, online, staff, guest, VIP)?

The festival attracted a total of 10,009 attendees across different categories. This distribution is shown in the **Chart** 2.14 below.

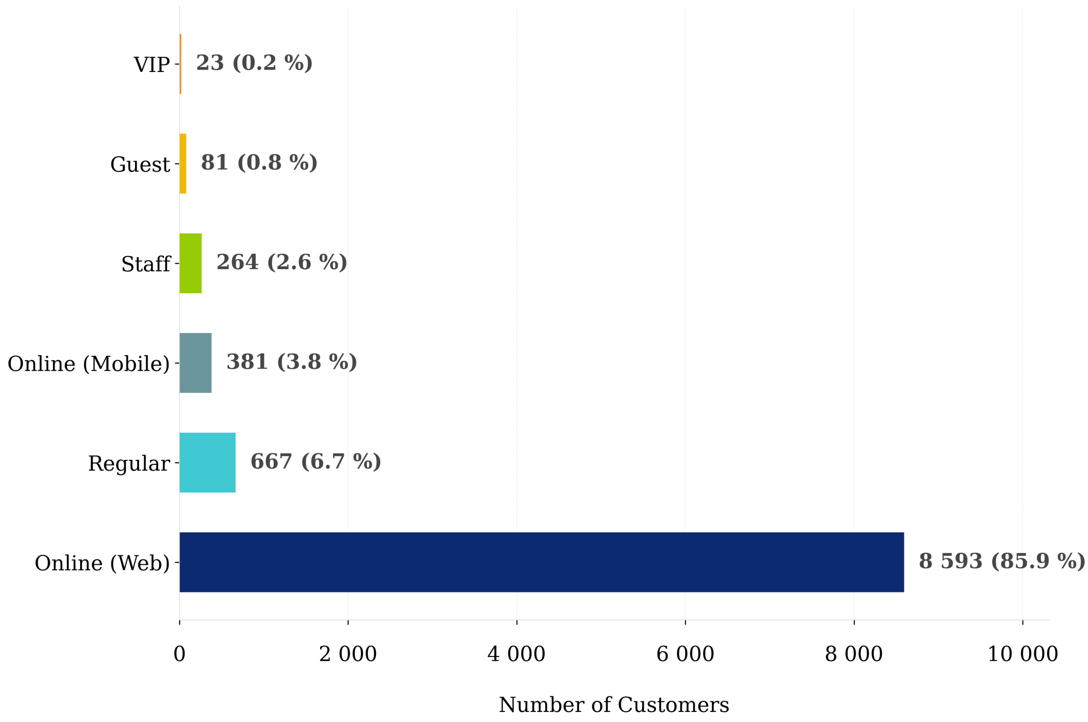


Chart 2.14: RQ20 Customer Distribution by Type

Source: *Author's rendition*

The data shows that the vast majority of attendees (**89.66**%) came through online ticket purchases. Regular on-site customers made up **6.66**% of attendees, while staff, guests, and VIPs collectively accounted for less than **4**% of the total attendance.

> Research Question
>
> **RQ21**: How many customers used the mobile app?

The system provides several ways for customers to interact with the festival, including mobile apps and web platform. These can be used for ticket purchases, credit top-ups, credit refunds, and other services. In the **Chart** 2.15 below is shown this breakdown.

The analysis reveals a significant adoption of mobile apps, with **40.69**% of customers using either iOS (**25.36**%) or Android (**15.33**%) app. The web platform remained the most popular choice, used by **55.17**% of customers. A small portion (**4.14**%) of customers had no recorded platform preference whatsoever.

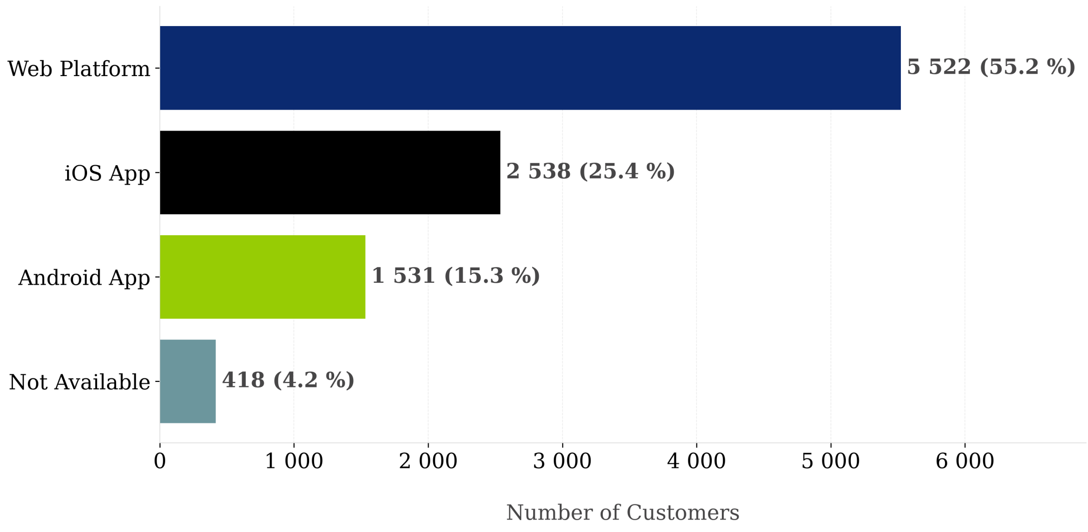


Chart 2.15: RQ21 Customer App Usage

Source: *Author's rendition*

Key Takeaways

- Only **18.20**% of online ticket holders preloaded their credit before the event
- Online ticket purchases dominated attendance (**89.66**% of all customers)
- Mobile app adoption was significant, with **40.69**% of customers using mobile platforms
- iOS app usage (**25.36**%) was notably higher than Android (**15.33**%)
- The web platform remained the preferred choice for most customers (**55.17**%)

2.4.3 Payment Behavior Analysis

This subsection focuses mainly on the payment segmentation and behavior of festival attendees.

By segmenting the customers based on their payment data, such as their card-issuing banks [2] and used payment card schemes[3] schemes, the analysis should

[2]Card Issuing bank is the bank responsible for issuing payment cards to customers; Issuers manage cardholder accounts, authorize transactions, and handle billing[9]

[3]Card Scheme, such as Mastercard or Visa, are a payment network provider who processes card payments globally[9]

find interesting patterns and answer the defined research questions.

Research Question

RQ22: What is the distribution of bank accounts used to refund credit?

The analysis of bank refunds reveals the distribution among Czech banking institutions shown in the **Chart** 2.16 below.

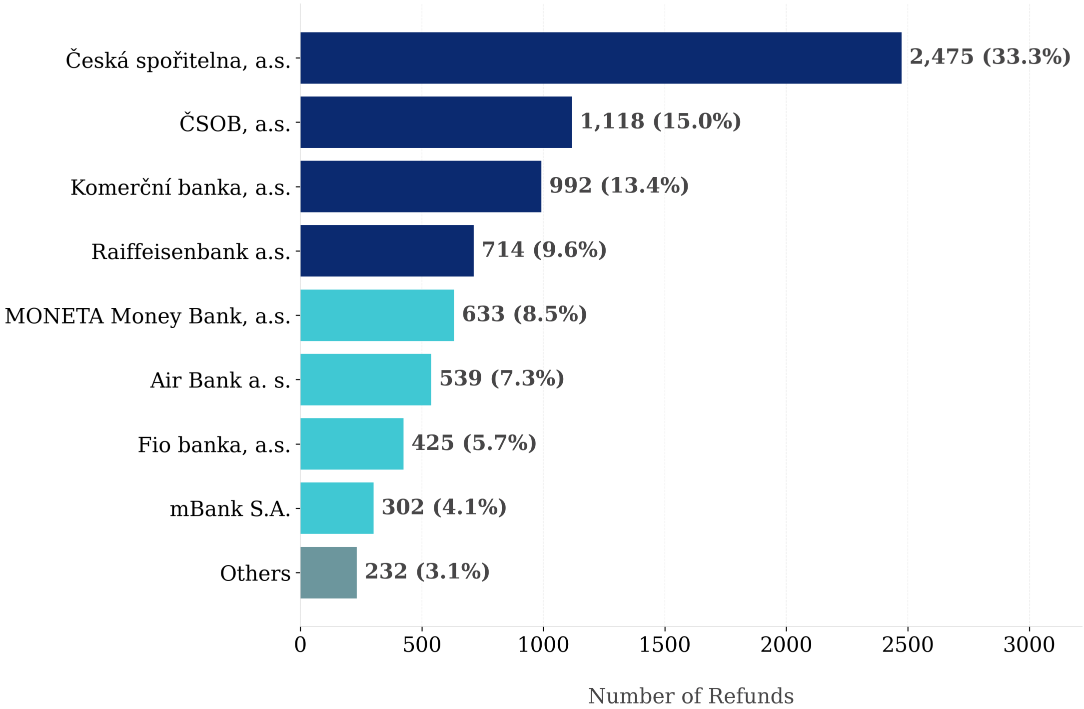


Chart 2.16: RQ22 Bank Refunds Distribution

Source: *Author's rendition*

Out of the **7,430** online credit refunds that were requested, the data shows that **Česká spořitelna** was the leading bank, accounting for **33.3**% of all refunds. Followed by **ČSOB** and **Komerční banka** with **15.0**% and **13.4**% share respectively.

Research Question

RQ23: What is the distribution of card schemes used to top up credit both on-site and online?

Out of total **12,626** card transactions done both online (online ticket purchases, credit top-ups in advance) and onsite (onsite top-ups at festival, onsite physical ticket purchases), only two card schemes were identified:

- **VISA** cards were used in **62.4**% of transactions, making them the dominant

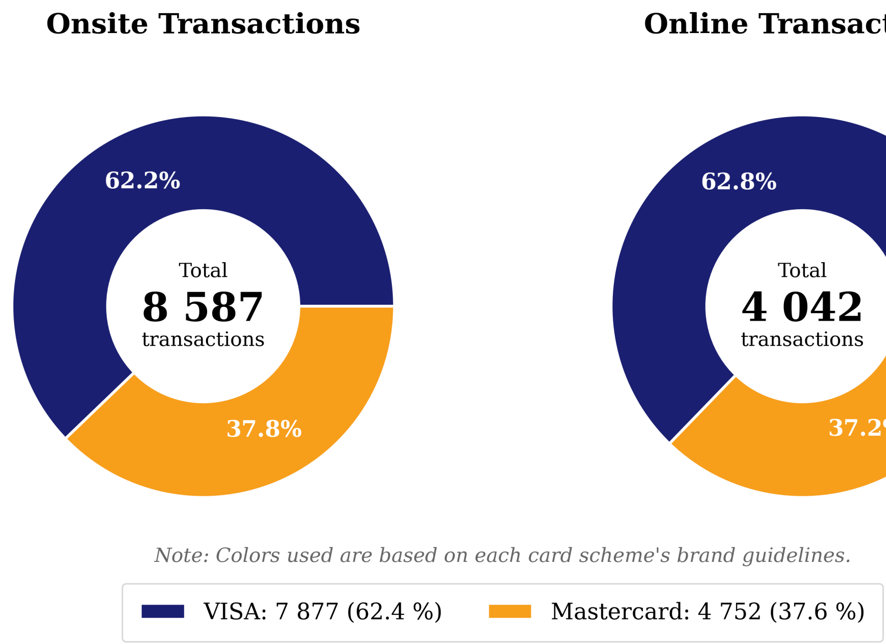


Chart 2.17: RQ23 Card Schemes Distribution

Source: *Author's rendition*

card scheme.

- **Mastercard** cards followed at **37.6**% of transactions.

This distribution rather nicely follows the local trend, where VISA is the most used card scheme with **65**% market share and Mastercard with **35**% market share[10].

> Research Question
>
> **RQ27**: What was the customer's onsite credit top-up frequency?

Analyzing onsite top-up frequency behavior at the festival reveals the distribution of top-up counts among customers, as shown in the **Chart** 2.18 below.

The analysis of top-up behavior reveals that most customers (**44.9**%) topped up only once during the festival. A significant portion (**29.8**%) topped up twice, while only a small fraction of customers (**5**%) topped up more than three times.

An interesting finding is that around **9**% of customers did not top up at onsite at all. This meant for Online customers that they preloaded their credit before the event and did not need nor required to top up during the festival.

For other on-site registered customers, it meant they did not use their chip

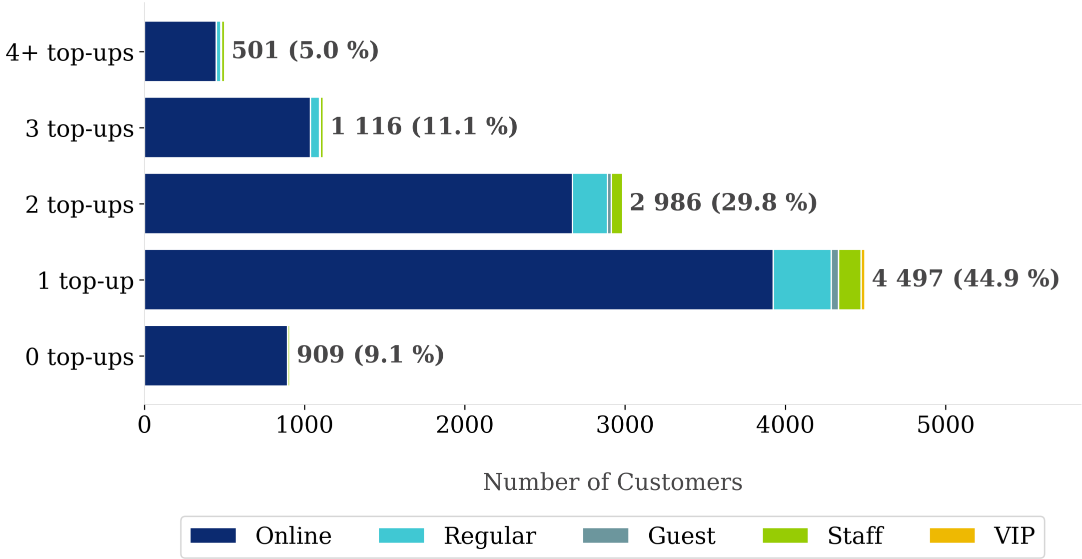


Chart 2.18: RQ27 Top-up Behavior Analysis

Source: *Author's rendition*

bracelet for payments at all, but only as an entry ticket or for access control purposes.

Key Takeaways

- The most common bank for refunds was **Česká spořitelna** (**33.3**%)
- **VISA** cards were predominantly used (**62**%) compared to **Mastercard** (**37**%)
- The majority of customers (**74.7**%) topped up their credit 1 or 2 times during the event
- Very few customers (**5**%) needed four or more top-ups

2.4.4 Purchase Pattern Analysis

This section examines customer purchase behaviors throughout the festival, focusing on beverage preferences across different times of day and common product combinations.

This section examines customer purchase behaviors throughout the festival, focusing on beverage preferences across different times of day and common product combinations.

Beverage Preferences Throughout the Day

> Research Question
>
> **RQ28**: What were the beverage preferences throughout the day?

To analyze beverage preferences throughout the day, sales data was aggregated by hour and beverage category. The **Chart** 2.19 shows the distribution of alcoholic and non-alcoholic beverage sales over the course of the day.

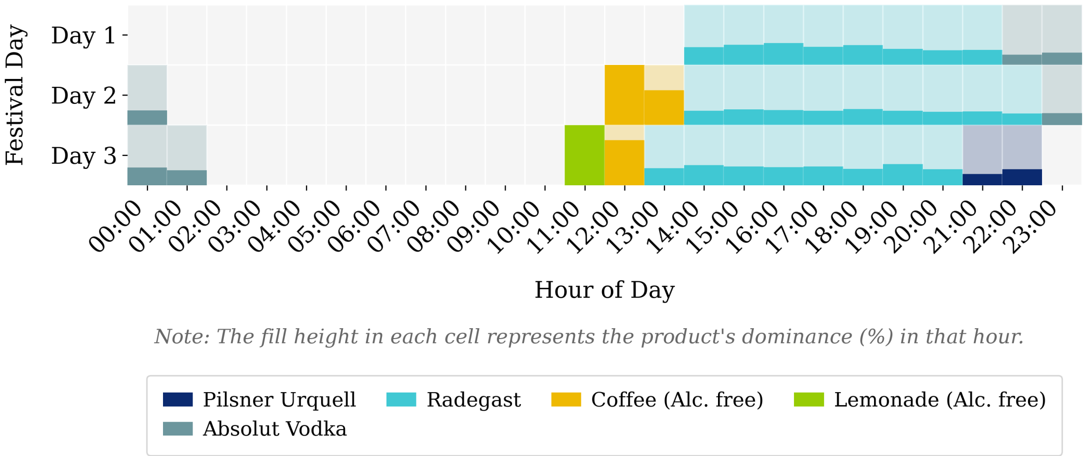


Chart 2.19: RQ28 Beverage Preferences Throughout the Day

Source: *Author's rendition*

These results reveal that non-alcoholic beverages, such as Coffee and Lemonades, were most popular around noon. Afternoon hours saw a shift towards light alcoholic beverages, with beer sales peaking in the late afternoon and early evening.

The beer preferences also show and confirm previous findings that Radegast was the most popular beer brand, with Pilsner Urquell following closely behind, especially at the end of the festival on Day 3.

Late evening hours saw a shift towards spirits and shots, with Absolut Vodka emerging as the most popular choice.

Hours without any preferences indicate that no sales were recorded during those times, or that the sales were too low ($\leq$ **10** products sold in an hour) to be considered significant.

This showed no sales data throughout the night and early morning, which is expected. However, one interesting observation is the lack of sales at the end of the festival on Day 3 after 22:00, which would indicate the festival's closing time.

Why would a festival close at 22:00 on a Saturday?

The festival ended prematurely on Day 3 due to an unexpected change in weather, which later resulted in evacuation of the festival grounds.

This information was not possible to extract from the data alone, but it was a known fact from the festival organizer.

Key Takeaways

- Non-alcoholic beverages were most popular around noon.
- Beer sales peaked in the late afternoon and early evening.
- Radegast was the most popular beer brand throughout the festival.
- Absolut Vodka was the most popular spirit in the late evening.
- The festival ended prematurely on Day 3 due to an unexpected change in weather.

Common Product Combinations

Research Question

RQ29: What were the most common product combinations?

Identifying frequently purchased product combinations can provide valuable insights into customer preferences and behavior. Two analyses were performed: one with returnable cups and one without, as the high frequency of cup purchases can obscure other meaningful patterns.

The initial results are shown in the **Chart** 2.20 below and ss expected, returnable cups appear prominently in the most popular combinations as it was necessary to purchase them for drinks.

To gain additional insights, the analysis was repeated while excluding returnable cups, as shown in the **Chart** 2.21 below.

This revealed the most popular product combination, without returnable cups included, was the **Absolut Vodka** and **Redbull** accounting to total of **2.48**% of all transactions.

There was also a notable high combination of non-beverage products, such as

Product A	Product B	Count	% of total
Kelímek - záloha	Radegast	5,202	4.69%
Absolut Vodka	Redbull	2,457	2.22%
Frisco	Kelímek - záloha	1,793	1.62%
Birell	Kelímek - záloha	1,728	1.56%
Kelímek - záloha	Pilsner Urquell	1,627	1.47%
Kelímek - záloha	ZON Lemonade	1,509	1.36%
Absolut Vodka	Kelímek - záloha	1,174	1.06%

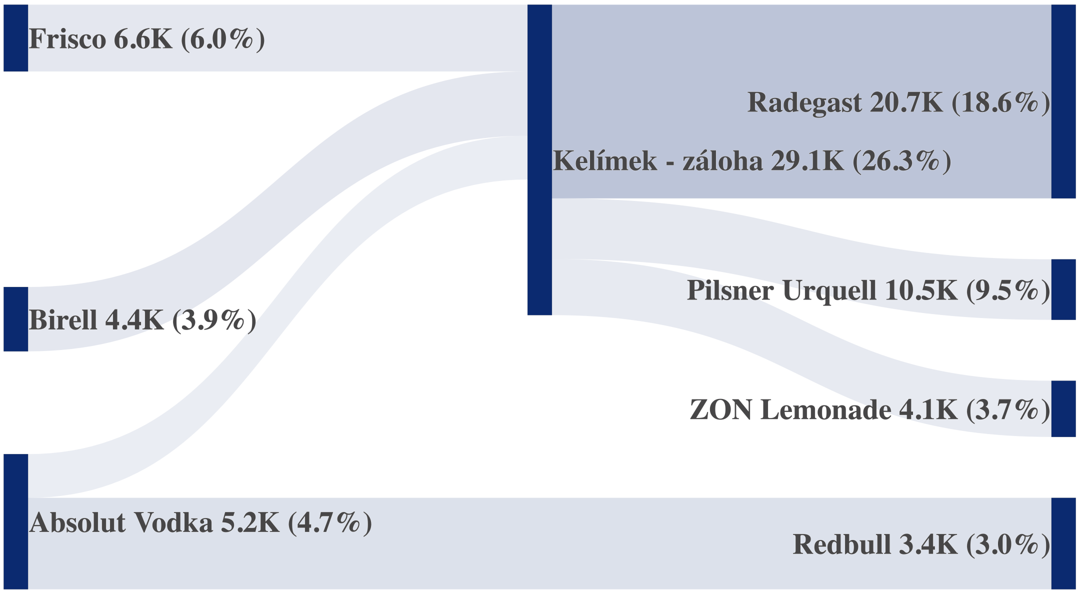


Chart 2.20: RQ29 Most Common Product Combinations with Cups

Source: *Author's rendition*

Klobása, **Hermelín**, or **Oštěpok** in combination with **Příloha**, making it a seemingly popular food choice.

Interpretation of the results could have been done in many ways, for example, using a network graph to visualize the relationships between products. This, as exciting as it sounded, was not in the end the most suitable way to present the results in a clear and concise manner.

Key Takeaways

- The most common product combination was **Absolut Vodka** and **Redbull** (**2.48**% of all transactions)
- The most common non-beverage combination was **Klobása** and **Příloha**
- Returnable cups were, not surprisingly, a significant part of the most common combinations

Product A	Product B	Count	% of total
Absolut Vodka	Redbull	2,457	2.48%
Klobása	Příloha	759	0.76%
BEERPONG - PIVO	BEERPONG - START	422	0.43%
Chléb	Ořez prasate	299	0.30%
Hermelín	Příloha	198	0.20%
Pilsner Urquell	ZON Lemonade	167	0.17%
Oštěpok	Příloha	159	0.16%

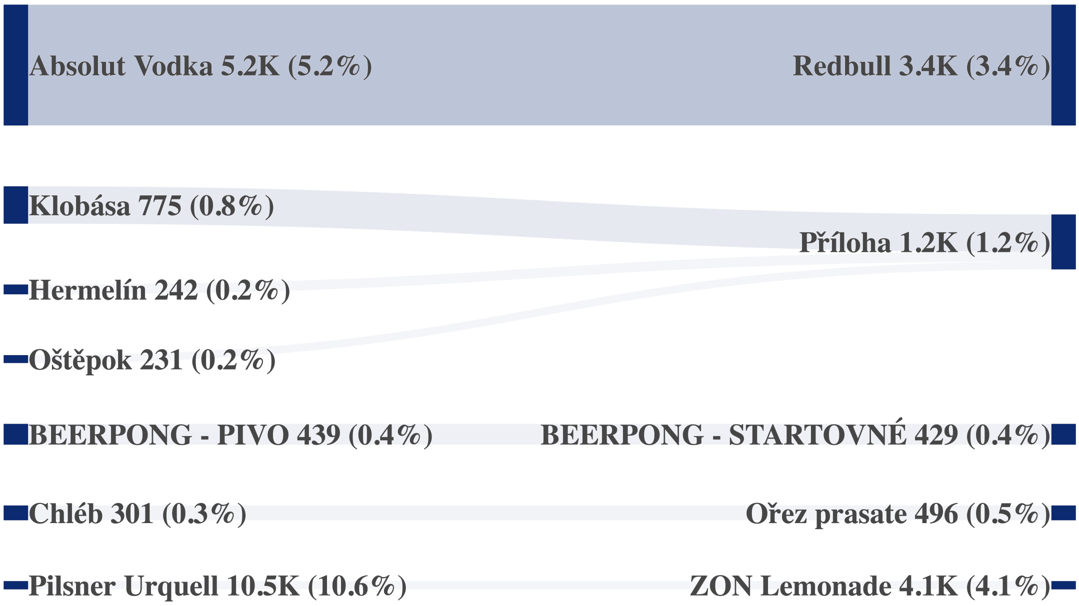


Chart 2.21: RQ29 Most Common Product Combinations without Cups

Source: *Author's rendition*

2.4.5 Summary

This section provided a thorough evaluation of festival attendees' habits, preferences, and segmentation. It provided practical information about visitor interactions with the festival by analyzing trends in attendance, payment methods, purchase preferences, and event product combinations that contributed most to overall sales.

All previously defined questions were answered and presented in a simple and understandable way, providing hopefully valuable insights for the festival organizer.

2.5 Conclusion

The analyses in this chapter provide a comprehensive look at the festival's behavior, including data-oriented insights on operational effectiveness, financial performance, and customer behavior. By fully addressing every research question, the results provide an in-depth knowledge of the festival dynamics and offer practical suggestions for upcoming seasons.

Analysis of cash flow dynamics and income sources revealed important revenue-generating systems and their interactions. Concurrent with this, performance indicators showed the system's ability to control transactional spikes and identified possible bottlenecks.

Examination of consumer behavior revealed notable trends in attendance and buying preferences.

Furthermore, the study of beverage consumption highlighted how important brands and product categories are for determining sales. The distinctive trends in beverage preferences and their correlation with specific times of day highlight the possibility to adapt offers to satisfy consumer needs.

The foundation for the upcoming dashboard development—which will show consolidated results in an interactive and user-centric form—is laid in this chapter.

3 Dashboard Implementation

This chapter describes the implementation of a prototype analytical dashboard that visualizes the key findings from the analysis. Using Dash and Plotly[1], we developed a local development prototype that demonstrates how the analytical insights could be presented in an interactive format. The implementation focused on efficient data querying, caching strategies for development, and handling asynchronous operations in Python.

The chapter details the technical approach, the challenges encountered during implementation (particularly with callback caching and async handling), and our solutions to these challenges. While not intended for production deployment, the prototype successfully demonstrates the potential of the analytical findings in an interactive format.

3.1 Development Approach

The implementation phase focused on transforming the analysis results from the **Chapter** 2 into an interactive dashboard prototype. Our approach prioritized rapid development and effective visualization of our analytical findings.

3.1.1 Development Goals

The primary goal was to create a functional prototype dashboard that could effectively present our analysis results in an interactive format. Specifically, we aimed to:

[1]Dash is a Python web application framework that enables the creation of interactive web applications using Plotly visualizations[11].

- Demonstrate key findings through interactive visualizations
- Implement basic filtering capabilities for data exploration
- Create a responsive interface that handles data operations efficiently
- Establish a foundation for visualizing festival transaction data

3.1.2 Technology Selection

The dashboard was built using Dash and Plotly, chosen primarily for their familiarity from prior personal experience in the academic field and their great capability for relatively fast data app prototyping[11].

The technology stack included:

- Dash & Plotly[2] using Python for the core dashboard framework
- Dash Mantine Components[3] for enhanced UI elements
- PostgreSQL[4] for local data storage

This stack was chosen specifically for prototyping, with Dash and Plotly providing a good balance of functionality and development speed based on previous personal experience using these tools in university projects. Mantine Components were added to enhance the visual presentation without significantly increasing additional development time.

3.1.3 Local Development Focus

The dashboard was developed exclusively for the local environment, focusing on demonstrating analytical capabilities rather than production readiness.

This decision was influenced by several factors:

- The prototype nature of the implementation, focusing on demonstrating analytical insights rather than production features

[2] https://dash.plotly.com/
[3] https://www.dash-mantine-components.com
[4] https://www.postgresql.org/

- Complexity of Python async handling, production requirements and deployment challenges with Python environments
- Implementing a fully featured production-ready dashboard would require significant additional development time and would not bring additional value to the project

Moreover, as stated previously, the personal motivation for this thesis was to learn more about the data and demonstrate the findings in an interactive way, leading to more insights and knowledge for future updates of the NFCtron Hub dashboard.

3.2 Core Architecture

Since it was developed for local use only, it did not require any complex environmental setups for the app nor the database. This significantly simplified and sped up the development process, allowing us to focus on the core functionality and visual presentation of the dashboard.

The dashboard's architecture was designed to efficiently manage data during development, with a strong emphasis on query management and caching strategies.

3.2.1 Query Management System

To manage the SQL database interactions efficiently, a simple **Query Management System** has been implemented to handle loading data from our SQL database.

It consisted of several key components:

- **QueryDefinition**: Defines query structure and parameters
- **QueryRegistry**: Maintains registered queries
- **QueryManager**: Handles query execution and caching
- **QueryParameter**: Defines parameter types and validation

The following example in **Source Code** 2 demonstrates the query registration process:

```
query_manager.registry.register_query(
    QueryDefinition(
        name="sankey_diagram",
        sql=QueryManager.process_sql_query("""
            SELECT * FROM get_sankey_diagram_data(:date_from$1, :date_to$2)
        """),
        parameters=[
            QueryParameter("date_from", datetime.datetime),
            QueryParameter("date_to", datetime.datetime)
        ],
        default_data="FSCacheDefault"  # Enables local CSV caching
    )
)
```

Source Code 2: Query Management Example

This class has been the core of the dashboard's data management, allowing multiple approaches to data loading and caching strategies.

It allowed defining raw SQL queries with `QueryManagerprocess_sql_query`, which safely handled parameter substitution and query formatting. Or it could have loaded an SQL query from an external file (due to the complexity of the query) and then executed it with parameters.

On top of that, it allowed for defining default data loading strategies, such as loading data from a local CSV cache, which significantly sped up the development process. More of this is described in the following section.

3.2.2 Dual Caching Strategy

To optimize the development process, the implementation uses two caching mechanisms that are complementary to one another.

Query Result Caching

Since the dashboard relies on several complex SQL queries to load data, it was crucial to optimize query execution during development. Running these queries repeatedly during development significantly slowed down the development process.

Thus, a basic file-based caching system was implemented to store query results as CSV files, preventing redundant database queries during development. It, simply by defining a `default_data` parameter in the `QueryDefinition` class, allowed turning on the caching for specific queries which did not need to be reloaded every time.

The query execution handler shown in the **Source Code** 3 below [5], demonstrates the simple implementation of this caching strategy.

```python
def execute_query(self, query_name: str, parameters: Dict[str, Any],
↪  or_query_def: Optional[QueryDefinition]):
    query_key = self.get_query_key(query_name, parameters)

  # Load query from cache if enabled
    if query_def.default_data == "FSCacheDefault":
        try:
            return pd._read_csv_cache(query_key)
        except Exception as e:
            print(f"Failed to load query {query_name} from cache: {e}")

    # Execute database query directly
    result = self._execute_db_query(query_def, parameters)

    # Cache result for future use
    try:
        self._save_csv_cache(query_key, result)
    except Exception as e:
        print(f"Failed to save query {query_name} to cache: {e}")

    return result
```

Source Code 3: Query Result Caching Implementation

[5] The implementation is rather a simplified pseudo-implementation, as the `execute_query` method is more complex and contains additional logic for results formatting, database connection retries and error handling

It firstly generates a unique query key based on the query name and its parameters in a form of hash, which then later was used as a directory and file path to the cached CSV data.

Then, if the caching is enabled for the query, it tries to read the CSV file from the cache directory and return it as a result. Otherwise, it executes the DB query directly as usual and then saves the result to the cache directory for future use.

Background Callback Caching

Since the dashboard, when not using cached data, relied on long-running data processing operations, it was crucial to use Dash's background callbacks feature. These background callbacks run in a separate background process in a queue, allowing the main Dash application to remain responsive while the callback processes the data and then returns the result to the main application[12].

Dash provides two options for background callback backends:

DiskCache: Runs the callback logic in a separate process and stores the results to disk using the diskcache library. This is the easiest backend to use for local development but is not recommended for production.

Celery: Runs the callback logic in a Celery worker[13] and returns results to the Dash app through a Celery broker like Redis. This is recommended for production as it queues the background callbacks, running them one-by-one in the order that they were received by dedicated Celery worker(s).

For our development, we used the DiskCache backend, as it was easier to set up and use for local development:

```python
CACHE_DIR = os.path.join(os.path.dirname(__file__), 'dash_cache')
os.makedirs(CACHE_DIR, exist_ok=True)
# cache configuration
cache = diskcache.Cache(
  directory=CACHE_DIR,
  size_limit=3e9,
  eviction_policy='least-recently-used',
)

# unique launch_uid to refresh cache when the app restarts
launch_uid = uuid4()
background_callback_manager = dash.DiskcacheManager(
  cache,
  cache_by=[lambda: launch_uid],
  expire=300
)
```

Source Code 4: DiskCache Background Callback Manager Setup

This implementation evolved through several iterations as we encountered scaling challenges. More on this is later described in the **Section** 3.3.

3.2.3 Custom Callback Management System

While developing the dashboard, we encountered a mismatch between our preferred asynchronous programming patterns and Dash's synchronous callback system.

Having extensive experience with asynchronous programming in the JavaScript ecosystem, we wanted to leverage similar patterns in our Python implementation. This led to a custom callback management system implementation that somehow bridged this gap.

The implementation addressed several development needs:

- **Async/Await Pattern Support**: The system allowed using familiar async/await syntax in our Python callbacks, maintaining consistency with the query execution manager which was already implemented using async patterns.

- **Unified Error Handling**: Instead of implementing error handling in each callback separately, the decorator provided centralized, but simple, error management and logging, reducing code duplication and ensuring consistent error handling across all callbacks.

- **Flexible Processing Modes**: The system seamlessly supported both synchronous and asynchronous operations through a single interface, automatically managing asyncio event loops [6] when needed.

The simplified pseudo-implementation [7], shown in **Source Code** 5, demonstrates this approach to the custom callback registration.

```python
def register_callback(self, output, inputs, background=False):
  def decorator(func):
    async def async_wrapper(_self, *args):
      try:
        return await func(_self, *args)
      except Exception as e:
        print(f"Error in {func.__name__}: {str(e)}")
        raise dash.exceptions.PreventUpdate

    if background:
      return self.__app.callback(
        output=output,
        inputs=inputs,
        background=True,
        *args, **kwargs
      )(async_wrapper)
    return self.__app.callback(output=output, inputs=inputs, *args,
    ↪  **kwargs)(func)

  return decorator
```

Source Code 5: Custom Callback Decorator Implementation

This unified approach simplified the possibly complex and boiler-plated, callback implementations, as shown in **Source Code** 6:

[6] Asyncio is a Python library to write concurrent code using the async/await syntax[14]

[7] The actual implementation includes additional features such as proper asyncio event loop management, argument parsing, and integration with the caching system. This example demonstrates the core pattern while omitting implementation details.

```python
@register_callback(
    output=Output("sankey-diagram", "figure"),
    inputs=[
        Input("date-range", "start_date"),
        Input("date-range", "end_date")
    ],
    background=True
)
async def update_sankey_diagram(self, start_date, end_date):
    data = await self.query_manager.execute_async(
        "sankey_diagram",
        parameters={"date_from": start_date, "date_to": end_date}
    )
    return create_sankey_figure(data)
```

Source Code 6: Callback Registration Example

This method worked seamlessly with the query management system and offered flexibility for simple and complex data processing. The combination of background processing and proper error handling proved essential for managing the complex data flows in the dashboard.

It allowed a better focus on callback business logic because the decorator pattern encapsulated the complexity of bridging sync and async environments.

3.2.4 Dashboard Structure

The dashboard has been structured to reflect the analysis sections from the **Chapter** 2, with dedicated sections for:

- **Cashflow and Revenue Analysis**,
- **Performance Analysis**,
- **Beverage Consumption Analysis**,
- and **Customer Analysis**.

Each section implements background callbacks for data loading and filtering, using the dual caching strategy to maintain responsiveness during development.

For each section, a separate file was created, containing the layout and callback definitions for the section. The main dashboard file then imports these sections and combines them into a single layout.

This approach allowed for better code organization and easier development of individual sections.

3.3 Technical Challenges and Solutions

The implementation process encountered several significant technical challenges, primarily centered around asynchronous operations and callback caching in Python. These challenges and their solutions significantly influenced the overall development approach and the outcome of the dashboard prototype.

3.3.1 Asynchronous Handling Challenges

Coming from a JavaScript background, where asynchronous programming is relatively straightforward[15], Python's async implementation presented several challenges. The main difficulties arose in implementing background tasks for the dashboard's filtering capabilities and data loading operations whilst already having implemented the query management system using async patterns.

However, thanks to the **Custom Callback Management System** implementation, it was possible to leverage Python's async/await patterns effectively.

3.3.2 Callback Caching Evolution

The evolution of the callback caching solution went through several iterations, each addressing specific challenges, as described below.

Initial Implementation

Following Dash documentation[12], we initially implemented a single `DiskcacheManager` with a simple cache directory setup:

```python
cache = diskcache.Cache(directory=CACHE_DIR)
background_callback_manager = dash.DiskcacheManager(
    cache,
    expire=300
)
```

Source Code 7: Initial Cache Manager Setup

However, since our dashboard had several callbacks registered, and many had the same inputs, due to the modular structure of the dashboard[8], this approach led to cache key collisions.

> **Why Cache Key Collisions Occurred**
>
> **Cache Key Collisions** occurred because the DiskcacheManager used the same cache key for callbacks with identical inputs. We did not know that the callbacks should ideally have unique inputs to prevent cache key collisions. As a result, the cached data was overwritten, leading to incorrect data being returned.

First Solution Attempt

To address cache key collisions, we tried to implement a unique DiskacheManager instance for each unique callback, based on it's `callback_id`:

```python
def create_callback_manager(callback_id):
    return dash.DiskcacheManager(
        cache,
        cache_by=[
            lambda: launch_uid,
            lambda: callback_id
        ]
    )
```

Source Code 8: Unique Cache Managers Per Callback

This solution worked well for individual dashboard sections but revealed issues when scaling and combining all dashboard sections together.

[8]Each dashboard section had its own callbacks; but many of them shared the same inputs, such as date range filters

Scaling Challenges

As the number of callbacks increased, we encountered SQLite concurrent connection issues. The local SQLite database used by DiskcacheManager[9] could not handle multiple concurrent write connections effectively, causing the dashboard to crash during loading. It was caused because each callback had its own DiskcacheManager instance, which created a separate SQLite connection for each callback, and all of them were trying to write to the same SQLite database file.

A more robust solution, recommended for production setup in Dash documentation – Celery & Redis [10], was considered as a solution to this issue.

Such a solution looked promising since it would allow for each callback to define `cache_by` parameter. This would be unique for each callback, and then the Celery worker would process the tasks in the order they were received, effectively preventing concurrent SQLite connection issues.

Due to the complexity of the setup, especially the Celery, and relatively a lack of instructions and functional examples, it was not easy to make it work in the local development environment.

We successfully set up the Redis server and Celery worker, but the integration with Dash was not successful. The Celery worker was not registering the tasks from the Dash app, despite following all available guides and examples. This resulted in no tasks being processed by the Celery worker, and the dashboard was not able to load any data.

Eventually, after an unsuccessful 12–hour overnight journey, we decided to look for a simpler solution as a result of the frustration, lack of progress, and time constraints.

Then a simple and dumb, yet effective solution was found, which was to create a separate cache directory for each callback, effectively isolating the SQLite database for each callback:

[9] `https://grantjenks.com/docs/diskcache/`

[10] Celery is a distributed task queue that allows for background task execution, while Redis is an in-memory data structure store used as a database, cache, and message broker[16][17][18]

```python
def create_isolated_cache_manager(callback_id):
    # Create separate SQLite database per callback
    cache_dir = os.path.join(
        os.path.dirname(__file__),
        f'dash_cache_{callback_id}'
    )
    cache = diskcache.Cache(
        directory=cache_dir,
        size_limit=3e9,
        eviction_policy='least-recently-used'
    )

    return dash.DiskcacheManager(
        cache,
        cache_by=[
            lambda: launch_uid,
            lambda: callback_id
        ],
        expire=300
    )
```

Source Code 9: Final Cache Implementation

While this solution is not production-ready and nearly on the edge of being a hack, it effectively solved the local development needs and allowed us to continue development without further issues.

3.4 Implementation Results

The dashboard prototype successfully demonstrates key findings from our festival analysis through interactive visualizations. Drawing from our analytical framework, we organized the interface into four main sections that provide different perspectives on the event data.

A global date-range filter, shown in the **Figure** 3.1, sits at the top of the dashboard, allowing users to analyze any time period during the festival. Preconfigured options for each festival day make it easy to quickly compare different phases of the event.

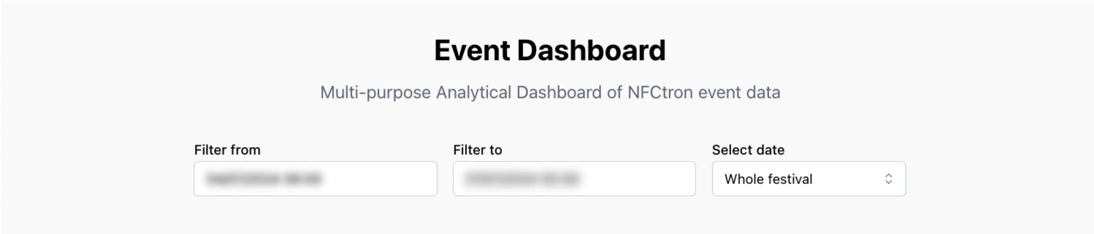


Figure 3.1: Dashboard Header with Date Range Filter

Source: *Author's rendition*

3.4.1 Cashflow and Revenue Analysis Section

The financial section gives an immediate overview of money flows through the festival system. Three key metrics at the top show the total revenue (**1,155,162 CZK**), available balances (**334,450 CZK**), and total chip top-ups (**14,520,973 CZK**). A visual representation of this section is shown in **Figure** 3.2.

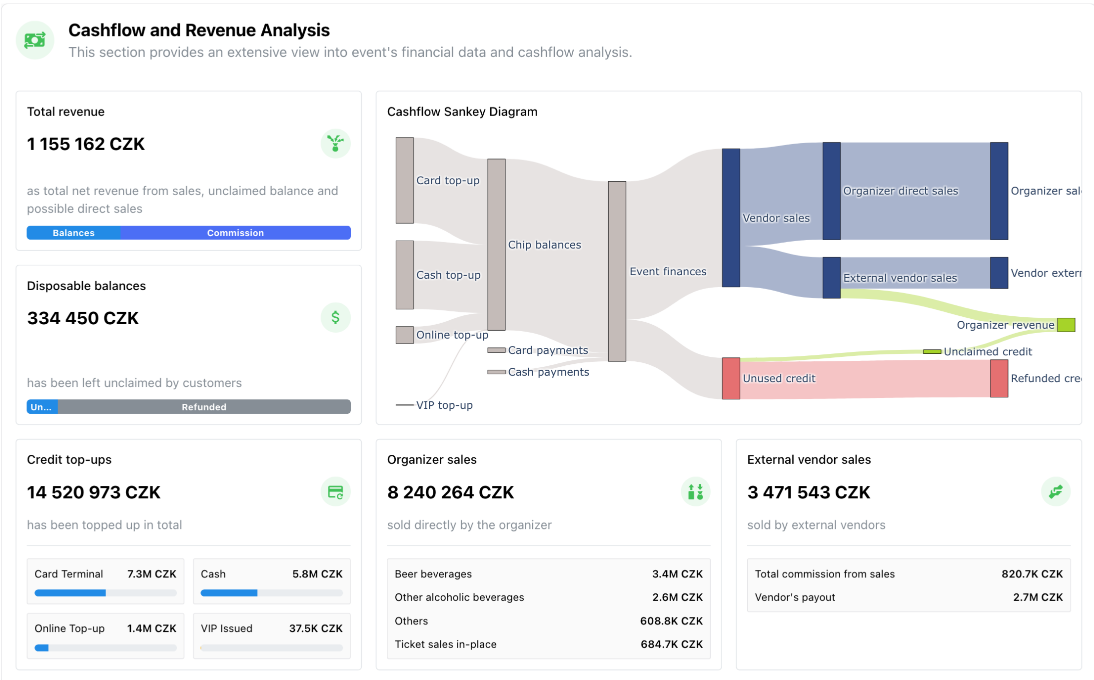


Figure 3.2: Cashflow and Revenue Analysis Section

Source: *Author's rendition*

The centerpiece Sankey diagram visualizes how money moves through the system—from initial top-ups through various payment channels to final settlements. This helps to track the conversion of chip credits into sales and monitor refund volumes.

Below, sales data splits between organizer (**8,240,264 CZK**) and vendor (**3,471,543 CZK**) categories, with detailed breakdowns for beverages, food, and other revenue streams.

3.4.2 Performance Analysis Section

The performance section helps to understand operational dynamics throughout the festival. Key metrics prominently display the festival's scale: **10,009** active customers, **141,378** processed transactions, and a peak volume of **8,986** transactions per hour. A visual representation of this section is shown in **Figure** 3.3.

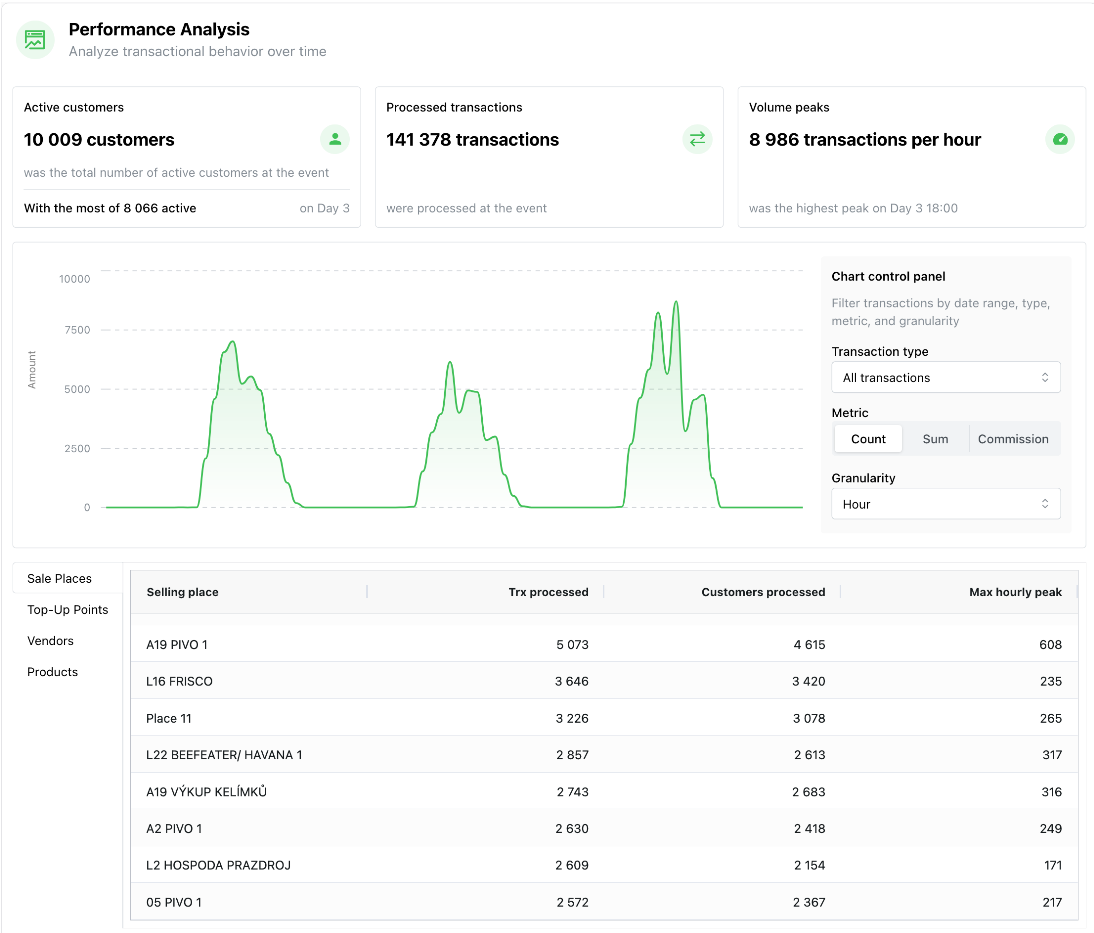


Figure 3.3: Performance Analysis Section

Source: *Author's rendition*

The main graph shows transaction volumes across all three days, with the highest peak visible on Day 3 at 18:00.

The control panel can be used to filter by transaction types and adjust time granularity for detailed analysis.

A table below highlights the busiest operational points, showing how many transactions and customers each location handled, as well as for individual vendors or products.

3.4.3 Beverage Consumption Analysis Section

The beverage section tracks consumption patterns and returnable cup management. At a glance, a total consumption (**29,343** liters), returnable cups issued (**22,045**), and the return rate of **78.40**% can immediately be seen. A visual representation of this section is shown in **Figure** 3.4.

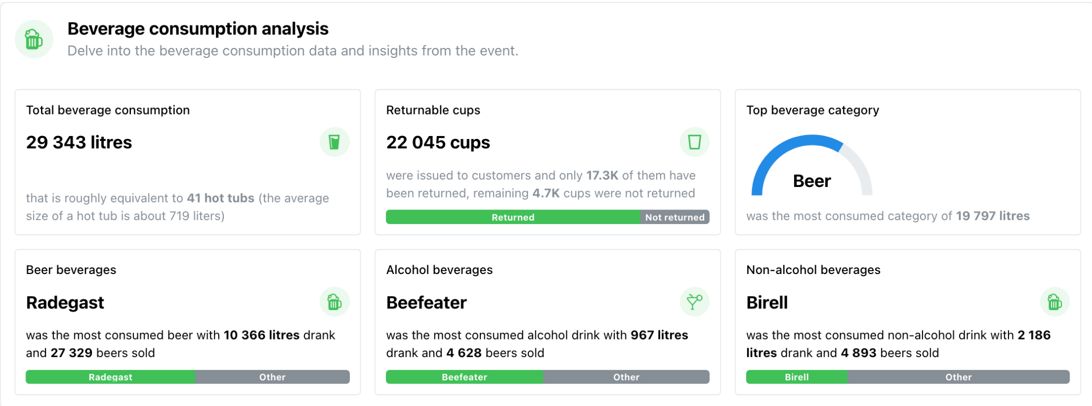


Figure 3.4: Beverage Consumption Analysis Section

Source: *Author's rendition*

Breaking down by category, **Radegast** led beer sales with **10,525** liters and **27,752** units sold. **Birell** dominated non-alcoholic beverages with **21,846** liters, while **Beefeater** led spirits with **966** liters consumed.

3.4.4 Customer Analysis Section

While currently still in development, the customer analysis section aims to provide insights into attendee behavior patterns. The planned implementation focuses on four key areas of visitor analysis.

The attendance timeline will show how **10,009** unique visitors were distributed across the festival, with Day 3 seeing peak attendance of **8,066** visitors. Customer segmentation will break down the **89.66**% online ticket purchasers and track platform preferences, including the **40.69**% mobile app usage rate. Payment analysis will visualize card scheme preferences (VISA **62.40**%, Mastercard **37.60**%) and top-up patterns.

This section represents a key area for future development, particularly for exploring customer behavior patterns. Unfortunately, due to the time constraints, it was not possible to finish the implementation of this section during the thesis project. However, the key findings from the analysis were already prepared and ready to be implemented.

3.5 Missing Features and Future Development

While the current implementation successfully demonstrates our analysis results, there are several directions for future development and improvement.

3.5.1 Technical Improvements

From a technical perspective, several key improvements would be needed for production deployment:

Infrastructure and Deployment: The current implementation is limited to local setup with a local database. Significant architectural changes would be needed to enable external access and proper production deployment with a more robust database solution and data layer separation.

Performance Optimization: SQL queries would require optimization for larger datasets. Implementation of materialized views and better query planning would improve response times with production-scale data. [11]

Caching Solution: A production-grade caching system using Redis would replace the current development-focused implementation. This would improve dashboard responsiveness and enable handling multiple concurrent users.

Data Formatting: Enhanced date formatting and chart configurations would improve data presentation. The current implementation uses basic formatting which would need refinement for production use.

[11] Materialized views are essentially database tables that store the results of a query and can be used to improve query performance[19]

3.5.2 Analytical Enhancements

From an analytical perspective, several key features would enhance the dashboard's utility:

Customer Analysis: Complete implementation of the customer analysis section would provide valuable insights into attendee behavior. This would enable better understanding of customer segments and their preferences.

Performance Metrics: Expansion of the performance section with detailed time processing metrics would provide better operational insights. This would help identify bottlenecks and optimization opportunities.

Timeline Navigation: A dynamic timeline player would allow users to observe festival data evolution over time. This would enable to replay the festival's progression and identify critical moments or patterns.

Interactive Elements: Additional interactive features would allow users to switch between different metrics and apply more granular filters. This flexibility would help to discover deeper insights and test different hypotheses about festival operations.

4 Conclusion

4.1 Summary of Work

This thesis has been built upon two main pillars: **the analysis of the data collected by NFCtron's system** and **the development of an interactive dashboard** to visualize the findings.

The analysis process started with the definition of the main research questions in cooperation with **The Organizer**, resulting in a total of **29 questions** to be answered. These questions were from a wide range of topics covering overall cashflow and revenue sources, system's performance, beverage consumption and customer behavior. A complete list of the research questions, together with the goals of the analysis, is covered in the **Introduction** chapter.

Before conducting the analysis, exploration of the data, followed by the identification of the main data quality issues and the data cleaning process, was necessary. This process covered also the local environment setup, the data import process, and the data model of the system, all thoroughly described in **Chapter** 1. Followed by a simple data anonymization process, the data was ready for the analysis.

The main part of the thesis, the data analysis itself, resulted in the most voluminous part of the thesis – that being **Chapter** 2. It was divided into the four research areas: **Cashflow and Revenue Sources Analysis**, **Performance Indicators Analysis**, **Beverage Consumption Analysis**, and **Customer Analysis**. Each of these sections contained a detailed analysis of the data, answering the research questions in detail, presenting the findings in rich charts, diagrams, and tables, providing the required insights.

The final part of the thesis, development of the interactive dashboard, was based on the findings of the analysis. This process took significant time and effort but resulted in an interactive dashboard that visualizes the most important findings of the analysis. Unfortunately, due to the time constraints, the dashboard was not fully completed, and some of the planned features were not implemented. The overall process of the implementation, from the initial approach, through core architecture, technical challenges faced, to the final implementation results and presentation of the dashboard, is described in **Chapter** 3.

The thesis now concludes with the reflections on the project, the professional impact and outcomes, and the future impact of the thesis on the NFCtron Hub system, as well as the final thoughts on the project.

4.2 Reflections

Working directly with festival data revealed both challenges and opportunities that may impact future development in the NFCtron system.

From a data perspective, the analysis revealed several opportunities for improving data collection. The need to manually add beverage volume information highlighted a gap in the current system's data model, which can unlock great insights with minimal effort. Similarly, the value of combining transaction data with event scheduling information demonstrated the importance of comprehensive data integration for meaningful insights.

The necessity of data anonymization for privacy reasons highlighted the importance of data protection and possible future challenges in this area. However, a simple data anonymization process was enough for the current analysis, and the data was successfully anonymized without losing its analytical value, providing a lesson for future data handling.

From the analytical side, the analysis process revealed the importance of defining clear research questions and objectives before starting the analysis. Initially, without having clear research questions, the analysis process was challenging, ineffective, and unstructured. However, this unstructured exploration phase was necessary to understand the data and to prepare multiple processes for the later data inspection. It also taught us various SQL techniques and optimization methods, from which we benefited in the later stages of the analysis and surely will benefit in the future.

After defining specific research questions, the analysis process became more focused, efficient, and productive, leading to straightforward insights and findings. The presentation of the results was crucial for this analysis, but initially the data presentation was not optimal as preparing the data for the visualization is a meticulous process. If not done correctly, the results may be misleading or incorrect. However, with the right tools, techniques, and adequate presentations, the results can be easily understood and interpreted. Thanks to the necessity to present data in a clear and understandable way, various better visualization charts were learned and used in the analysis. This included charts such as the waterfall chart, packed bubble chart or a simple pie chart alternative – the waffle chart.

Finally, the development of the interactive dashboard was a challenging but rewarding process, and the technical challenges faced provided valuable lessons. While the initial impulse often was to implement complex "proper" solutions, simpler approaches frequently proved to be more effective and efficient. This was particularly obvious in the caching system evolution, where a straightforward file-based solution ultimately outperformed more sophisticated alternatives for development purposes.

Many situations require a balance between the complexity of the solution and the time available for implementation. As a result, the dashboard was not fully completed, and some of the planned features were not implemented, which leaves room for future development and improvements.

Overall, the thesis provided required insights into the data, the analysis process, and the development of interactive dashboards. It showed the importance of clear research questions, data quality and processing and an understandable data presentation for effective analysis results.

4.3 Professional Impact and Outcomes

The main personal motivation for this thesis was to learn more about the data and demonstrate the findings in an interactive way, leading to more insights and knowledge for future updates of the NFCtron Hub dashboard.

The research and analysis process has already given new understanding and knowledge about the data, the SQL techniques, and optimization methods. Thanks to this, we were already able to manage and deliver, together with our team, two significant updates to NFCtron Hub's analytical capabilities in late 2024.

These updates resulted in a new time-based charts for ticket sales breakdowns and a comprehensive timeline view of festival activities, including its sales, refunds, customer arrivals, and customer ratings. A simple preview of this new timeline component can be seen in the **Figure** 4.1 below.

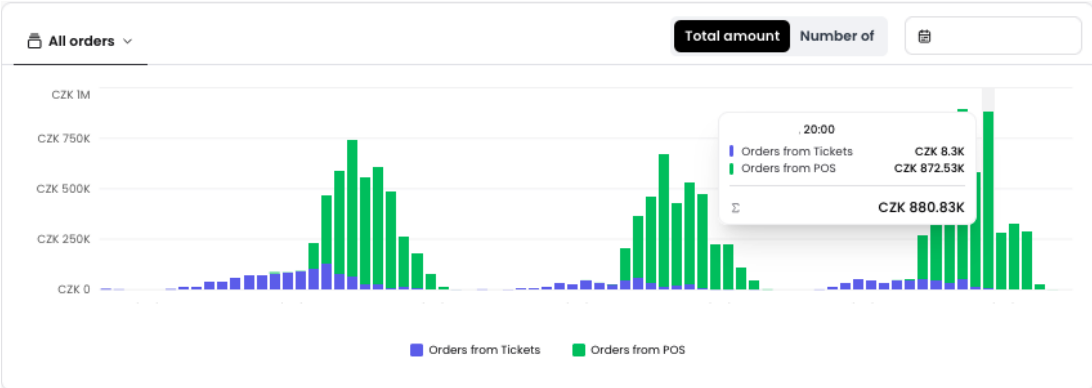


Figure 4.1: New Timeline Analysis in NFCtron Hub (December 2024 Update)

Source: *NFCtron Hub[20]*

Also, thanks to the analytical findings and problems encountered during the analysis, we were able to provide valuable feedback to the development team. This will lead to improvements in the data collection process and the data model of the system, which will make future analysis easier and more efficient.

Finally, the knowledge acquired by this thesis still shapes our product development and enables us to provide more advanced analytical capabilities to our clients – festival organizers. Future development will build upon these foundations, further expanding the analytical capabilities of NFCtron's system.

Bibliography

[1] Bas Leunissen. Why festivals are switching to cashless payments: six reasons. Online. Available from: https://www.tactile.events/en/waarom-festivals-overstappen-op-cashless-betalen, 2024. [cit. 2024-12-31].

[2] NFCtron. Seamless payment is no longer the music of future. Online. Available from: https://www.nfctron.com/en/company, 2024. [cit. 2024-12-31].

[3] NFCtron. About us - nfctron. Online. Available from: https://www.nfctron.com/en/about-us, 2024. [cit. 2024-12-31].

[4] NFCtron. Seamless festival payments arrive in austria with nfctron - nfctron. Online. Available from: https://www.nfctron.com/en/blog/nfctron-austria-events-festivals-september-visitors-payments, 2024. [cit. 2024-12-31].

[5] NFCtron. At the nfctron keynote 2023, we presented novelties for the 2023 season as well as the nebula and nautilus projects - nfctron. Online. Available from: https://www.nfctron.com/en/blog/nfctron-keynote-mastercard-cashless-event, 2023. [cit. 2024-12-31].

[6] PostGIS. Postgis. Online. Available from: https://postgis.net/, 2024. [cit. 2024-12-31].

[7] Heather Devane. What are the top data anonymization techniques? Online. Available from: https://www.immuta.com/blog/data-anonymization-techniques/, 2024. [cit. 2024-12-31].

[8] University of Regina DBD. Overview of the kdd process. Online. Available from:

https://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html, 2024. [cit. 2024-12-31].

[9] John Clark. What are card schemes and how do they work? a guide for small businesses. Online. Available from: https://www.takepayments.com/blog/online-and-tech/what-are-card-schemes-and-how-do-they-work/, 2024. [cit. 2024-12-31].

[10] Sdružení pro bankovní karty. sbk website. Online. Available from: https://www.bankovnikarty.cz/pages/czech/profil_karty.html, 2024. [cit. 2024-12-31].

[11] Plotly. Dash documentation user guide — plotly. Online. Available from: https://dash.plotly.com/, 2024. [cit. 2024-12-31].

[12] Plotly. Background callbacks — dash for python documentation — plotly. Online. Available from: https://dash.plotly.com/background-callbacks, 2024. [cit. 2024-12-31].

[13] Celery. Workers guide — celery 5.4.0 documentation. Online. Available from: https://docs.celeryq.dev/en/v5.4.0/userguide/workers.html, 2024. [cit. 2024-12-31].

[14] Python Software Foundation. asyncio — asynchronous i/o — python 3.13.1 documentation. Online. Available from: https://docs.python.org/3/library/asyncio.html, 2024. [cit. 2024-12-31].

[15] Node.js. Javascript asynchronous programming and callbacks. Online. Available from: https://nodejs.org/en/learn/asynchronous-work/javascript-asynchronous-programming-and-callbacks, 2024. [cit. 2024-12-31].

[16] Reintech. Celery and redis: Combining caching with task queuing. Online. Available from: https://reintech.io/blog/combining-celery-redis-caching-task-queuing, 2024. [cit. 2024-12-31].

[17] Celery. getting-started introduction. Online. Available from: https://docs.celeryq.dev/en/stable/getting-started/introduction.html, 2024. [cit. 2024-12-31].

[18] Redis. Docs. Online. Available from: https://redis.io/docs/latest/, 2024. [cit. 2024-12-31].

[19] The PostgreSQL Global Development Group. Postgresql: Documentation: 17: 39.3. materialized views. Online. Available from: https://www.postgresql.org/docs/current/rules-materializedviews.html, 2024. [cit. 2024-12-31].

[20] NFCtron. Nfctron hub. Online. Available from: https://hub.nfctron.com/, 2024. [cit. 2024-12-31].

List of Figures

Note: Unless stated otherwise, all figures are the author's rendition.

List of Tables

Note: Unless stated otherwise, all tables are the author's rendition.

List of Charts

Note: Unless stated otherwise, all charts are the author's rendition.

List of Source Codes

List of Appendices

Appendix A Source code of the application

The source code for the dashboard application is available on the attached CD in the `dashboard_app.zip` file.

The application is built using Python 3.9[1], and its requirements are listed in the `requirements.txt` file.

The application is structured as follows:

- `app.py`: the main Dash application implementation
- `queries`: directory with SQL queries
- `sections`: directory with implementation of dashboard sections
- `assets`: directory with CSS stylesheets
- `_dash_utils.py`: utility functions for Dash
- `_db_utils.py`: common database utility functions with query manager implementation
- `_query_manager.py`: registered SQL queries
- `_format_utils.py`: utility functions for data formatting
- `_chart_utils.py`: utility functions for chart generation, SankeyDiagram implementation

[1] `https://www.python.org/downloads/release/python-390/`