Project Final

Dennis Wasser s2831791
Lars Dopmeijer s3083616
☎ Gracjan Chmienicki s3077489
Maria-Paulina Costea s2883139
Victor Cebotar s2836238

Tutor: Aniruddha Vaidya

Publication date: 04-11-2022
Contact person: +48 518 326 837

BIT 11

Content

Introduction

The management of theme park Picco Melodioso is not satisfied with the revenue of the past years that the theme park has been generating. They want to find out why this problem is occurring, and how it can be fixed. In order to do this, the managers decided to hire a consultant team.

The team has been tasked with tackling the problem of the low revenue. In order to find out how to solve it, there had to be a greater understanding of why the problem exists. After all, there could be a lot of different reasons causing the low revenue. So, the team had to properly analyze the problem, and most importantly, the causes behind it. By doing a problem analysis, the team found the underlying problems, which caused the lack of revenue. It turned out that there were many intertwined problems, such as a poorly cleaned park, which in turn leads to a low level of customer satisfaction, and this leads to lower revenue, because there will be fewer tickets sold.

At this point, the team had to decide which of the problems was the most useful to solve, in terms of effects on the park and in terms of costs. After a critical consideration, the team decided to solve one of the core problems, this being the lack of meal planning. The team landed on this problem, because solving it would increase the customer satisfaction, which leads to a higher revenue, as mentioned above.

With a chosen problem, the team had to think of possible solutions to solve it. After producing different solutions, there was an important decision to be made: which solution is the best? The team found that the best way to choose the solution, was by ranking the solutions on different criteria. Namely by giving each solution a score ranging from 1 to 10 on 4 attributes; costs, time to implement, amount of waste and the employees needed for the implementation. But this was not sufficient, because not all of the criteria had an equal amount of importance. So, the team simply gave each criteria a weight, which showed how important one was. Using this method, it was decided that creating an app that lets customers order their food before they arrive at the theme park was the best possible solution to tackle the problem.

In order to realize the solution, the team created a so-called Business Process Modeling Notation (BPMN). In this case the BPMN visualizes in which processes the app is involved. This allowed the team to get a clear image of how the app should function. After this, it was time to start the development of the actual application. The team started with designing the different pages and aspects of the app. After the app was properly designed, the internals had to be created. Using the web application 'Thunkable' the team was able to make the app work as envisioned. There were a few snags in the way of the finalization of the solution. It was at first proposed that the customers would be able to pay in-app using either PayPal or iDEAL. In reality, the team was not able to implement this. At this time, the solution was almost ready to be implemented, only the testing and debugging of the application were left.

In the end, the app was properly set up, and lets the customers order their food as they desire. The team is confident that the provided solution will be able to solve the core problem, and thus raising the revenue. In the upcoming pages a detailed description of each part int the solution development can be read and seen.

Problem Analysis

The amusement park “Picco Melodioso” struggles with several issues. An in-depth analysis was taken to see what causes them and a conclusion has been made. The poor meal planning turned out to be the core problem resulting in tickets being sold too expensive; meal ingredients that are not delivered on time or in sufficient quantities; huge amounts of food waste which is highly undesirable; low customers’ satisfaction level. The group considered this as a critical point and decided to solve it.

Customers can order one of three meals: fries, pizza, or pancakes. However, sometimes the meal of their choice is already sold out or the kitchen simply runs out of ingredients to prepare it, so they must choose something else. Furthermore, at the end of the day, the kitchen throws away a lot of food. The kitchen staff simply don’t know how many ingredients should be ordered in order to prepare enough meals while not ordering too much.

Data analysis is a key process that should not be overlooked, although Picco Melodioso seems to underestimate its power. This aspect helps the company to gain important information that will help in better planning of expenses, which means less capital wasted. Such a process will also contribute to the correct investments, which will allow the company to quickly develop and expand. Knowledge about customers’ most frequent choices lets the park’s staff provide the right amount of goods.

Poor meal planning is often the result of many neglects and oversight. Regarding Picco Melodioso’s case, this seems to happen unconsciously. The amusement park does not keep track of its orders and it leads to the consequences. This is where the app comes in and sets new rules. It keeps track how many meals of each kind have been ordered and stores it in a database. Moreover, it provides sufficient information and generates essential statistics.

To conclude, the problem that is trying to be solved is poor meal planning. The alternative approach that will be used is simple to implement, and the benefits will be well worth it.

Project Idea

Description of the core app idea and the main functionality

An app allows users to indicate their preferences about their meals. However, it is important for them to let it know in time (suggested 24 hours before arrival), that the warehousemen provide the kitchen with the right amount of ingredients. Obviously, the restaurant must provide the meals to everyone who wants to buy them. The group came up with an idea of encouraging parks' customers to download the app, by simply informing them about benefits coming from it. At purchasing ticket stage, users must be notified about 33% discount for all pre-ordered meals in application.

Explanation of how the app can contribute to solve the identified problem

Since the identified problem is the meal planning, the app provides a direct solution to it. With the possibility of stating meals inside the app, customers are willing to reserve it in advance. The employees will be able to order the right amount of ingredient supplies. However, the order should still be a bit more than the declared amount for people who did not use the app (the price will be 1,5 times higher, yet all customers will stay satisfied). Additionally, the feature of counting the average number of meals sold in a day will help to provide the statistics for future orders.

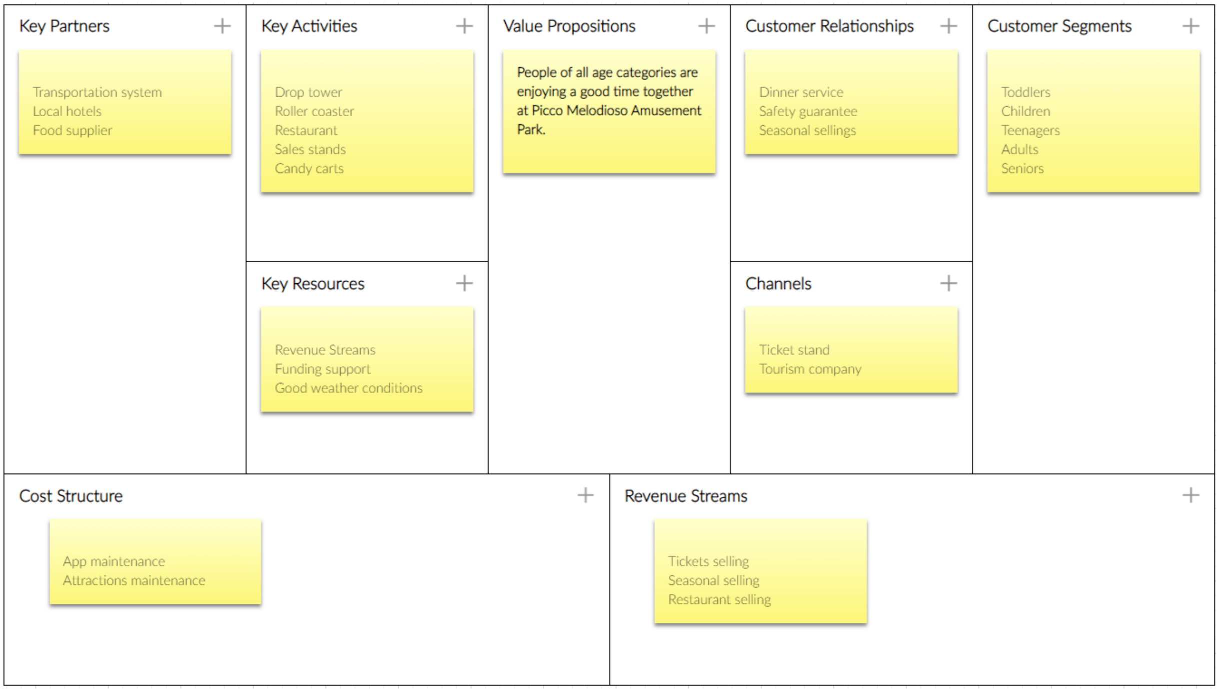
Declaring meals beforehand and planning supply deliveries wisely will result in having less food waste for the restaurant and will allow all customers to have the meal of their choice. These features will also lower the costs spent on the food by both customers buying it and by the staff ordering and preparing meals.

Description and impact analysis of the involved actors

The app described above impacts three major stakeholders. The first stakeholder are the managers of the theme park, because the app impacts their revenue. Most of the customers will indicate their food preference in the app. The restaurant can adjust their orders to the data of the app, which in turn leads to a lower waste of food, which leads to a higher revenue. This brings the second stakeholder, which is the restaurant staff. The staff can get detailed information about the number of orders and the amount of food which is needed. Because of this, the restaurant staff can prepare the orders in advance, which will result in lower waiting times for the customers. The last direct stakeholders are the customers. In the app the customers can order their food in advance, this will ensure that the food that they want is available. Another impact of the app for the customers is the fact that they can order their food at a discounted price. There is also a major indirect stakeholder, which is impacted by the app. This is the supplier of the ingredients for the restaurant. They get impacted by the app, because the restaurant will order different amounts of ingredients. Before the app the restaurant ordered the same amount of ingredients every time.

Osterwalder Business Model

This Osterwalder Business Model of our application The Key Partners are the ones on whom the company relies for constant support to use the application. For the application to run it needs the participation of the restaurant staff, the cashier and the kitchen staff. They will prepare the meals that the customers selected in the application and validate that the customer paid. The key activities show what the app provides for the theme park. In this case it is increasing the efficiency in the restaurant. The staff knows what meals people are going to order. The application also shows detailed information about what and when customers order. This data can be used in the future to predict what customers are going to order. In that way the restaurant can prepare on time and make sure every meal is available without much food waste. What is needed from the theme park are the key resources. For the app to run effectively it needs to get the data about what the customers order, the amount and also the prices of the meals. The app will show the amount of orders to the kitchen staff and will store how much money is being made for the financial department. The value propositions are the benefits the application delivers to the users. This app will help customers buy food, it is easier through the app and they get the food with a discount. By ordering food in advance the waiting times will decrease because the kitchen staff knows what the customer wants before he/she is even there. At the customer relationships is about what the customer expects from the application. In this case they expect to see useful information about the park and to order food with a discount. The customer segments is about who the app is creating a value for. The app is creating a value for the customers as they get food with a discount, and it is creating value for the park as the park is getting more revenue because more people are ordering food. The customer segments are being reached through the theme park, as the customer is in the park it will see about the app, and the app itself as the app is working as an advertisement for the park. The cost structure shows what the park is spending money on to get the app. It is on developing the app and updating/maintaining the app. The revenue streams show how the theme park will earn money from the app. Because the restaurant will waste less food as it mostly knows what the customers are going to order for the coming days it will have less inventory left at the end of the day. Everything that has been bought and is thrown away at the end of the day is wasted money. Because the restaurant will have less waste they also reduce the expenses. For customers, it is way easier to order food now because of the reduced waiting times and the discount. Because of this, more customers will order which will create more revenue for the theme park. It is also optional to put advertisements in the app for example the cotton candy carts or even for external companies.



Key Partners +

Transportation system
Local hotels
Food supplier

Key Activities +

Drop tower
Roller coaster
Restaurant
Sales stands
Candy carts

Key Resources +

Revenue Streams
Funding support
Good weather conditions

Value Propositions +

People of all age categories are enjoying a good time together at Picco Melodioso Amusement Park.

Customer Relationships +

Dinner service
Safety guarantee
Seasonal sellings

Channels +

Ticket stand
Tourism company

Customer Segments +

Toddlers
Children
Teenagers
Adults
Seniors

Cost Structure +

App maintenance
Attractions maintenance

Revenue Streams +

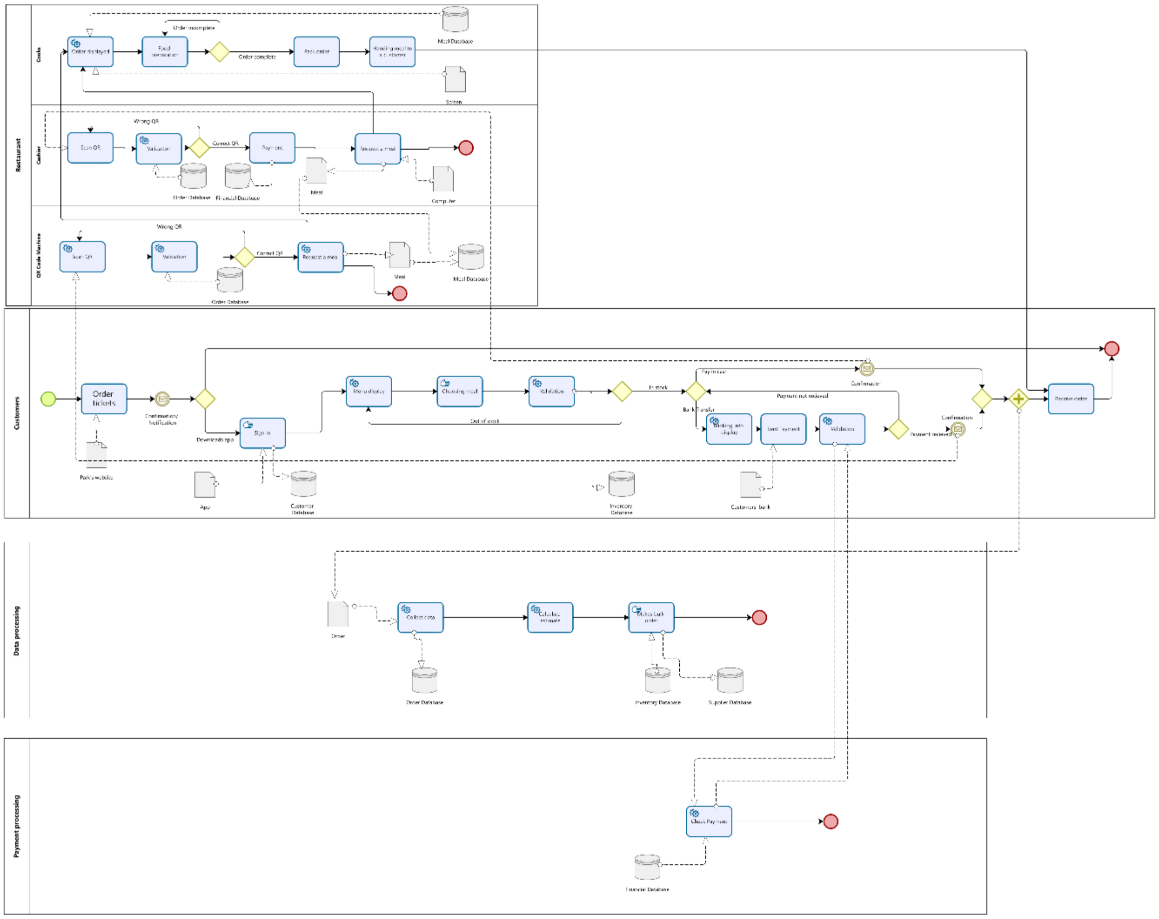
Tickets selling
Seasonal selling
Restaurant selling

Osterwalder Business Model (see appendix C1 for enlarged picture)

IT Landscape

Business Process Modeling Notation

A Business Process Modeling Notation (BPMN) has been made in order to map out the different functions of the to be developed application. This was developed by putting the general idea of the app into smaller sections of the processes in which they belong. First the team had to think of the processes happening in the restaurant, and then select the processes where the potential app would be involved. After this the individual tasks of the app had to be specified. This resulted in the BPMN as seen below. Using the BPMN as a blueprint, it is clear in which order the application should handle its' functions and tasks.



Business Process Model and Notation (see appendix C2 for enlarged picture)

Discussion BPMN

The BPMN consists of four different intertwined pools. As a customer after you bought tickets, you can choose to reserve meal for the day you will be visiting the theme park. When a customer logs in, the data will automatically be transferred into the customer database. After validation of their chosen meal, that information will directly go into the order database. If the chosen meal is not in stock, the customer will be sent back to the meal choice page in the app. The customer can either pay in person, or through the app. If the customer decides to pay in-app, then the payment can be done by sending a bank transfer to the bank details provided by the app. When the app has validated that the payment is completed, then the customer will receive an confirmation mail. The customer will receive a custom QR code which can be either scanned by the staff or at the QR-code machine once they are in the restaurant. The QR-code will link to a meal in the order database to look up what meal they ordered. After payment confirmation, the meal order will be transferred to the meal database so that the cooks can see what they have to prepare. After ensuring that the order is completed, it is being packed and handed to the customer.

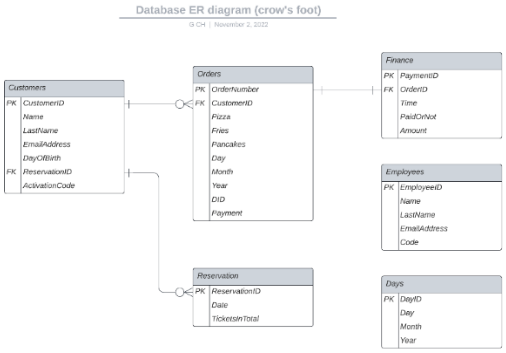
When the customer receives their order confirmation, simultaneously their order data will be collected and stored in the order database. The inventory management system will be using the order database to estimate how many orders will be ordered in the near future. After seeing what the estimated orders will be and checking the inventory database, a staff member will order the ingredients at the supplier. This information will be stored in the supplier database.

Entity-Relationship diagram

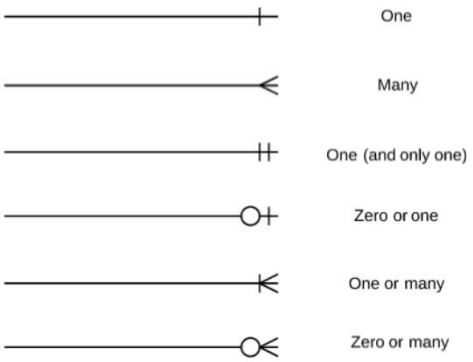
In an Entity-Relationship diagram (ERD), the different databases and its' relations to other databases can be easily visualized. This is done by linking the databases in the diagram by using 'Primary Keys' (PK) and 'Foreign Keys' (FK). Primary key: A candidate key chosen by the database designer to uniquely identify the entity set. Foreign key: Identifies the relationship between entities. The ERD is made using the 'crows-foot notation', these denote the cardinality between the entities. The cardinality defines the numerical attributes of the relationship between two entities or entity sets, also see the meanings of the links below (enlarged pictures are available in the appendix).

To work properly, the app uses six different entities. Customers database stores essential information including 'ActivationCode' that lets the app validate user's credentials. Using the Reservation database the app can easily access the date of reservation. Moreover, by 'CustomerID' the Orders database gets access to its properties and stores more useful information regarding order size and payment. Then the Finance database gets access to the Orders database and stores great details.

The app also uses Employees and Days databases. These are not connected to any others, but they play a key role in the properly working app. Namely, the first one is responsible for storing employees' credentials including their 'code'. Days on the other hand, assigns 'DayID' to every day in order to count days more easily.



Entity-relationship diagram (see appendix C3 for enlarged picture)

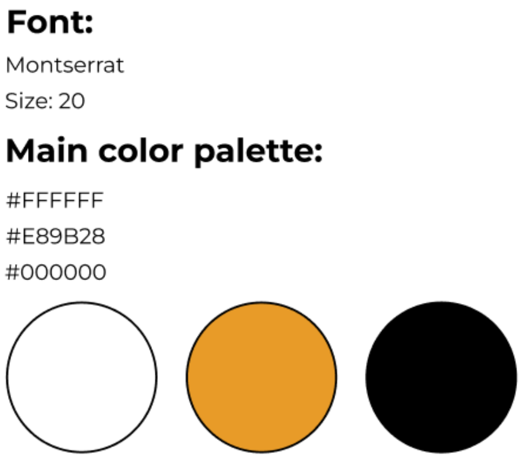


Crow's-Foot notation (see appendix C4 for enlarged picture)

Application Design

First, the color palette was chosen to reflect the right colors of both the amusement park and restaurant. After concluding small research, it was decided to choose orange color as a base for the user interface, since orange represents joy, fun, action and energy. It is also used in an app’s title to draw user’s attention. The two combining colors of choice are black and white, that contrast with each other, and do not distract the user from the main color – orange.

The next step was to choose an appropriate font. “Montserrat” happened to be the chosen one, since it is mostly used for creating a simple and clean-looking design structure and provides nine different options of font thickness.



Font:

Montserrat

Size: 20

Main color palette:

#FFFFFF

#E89B28

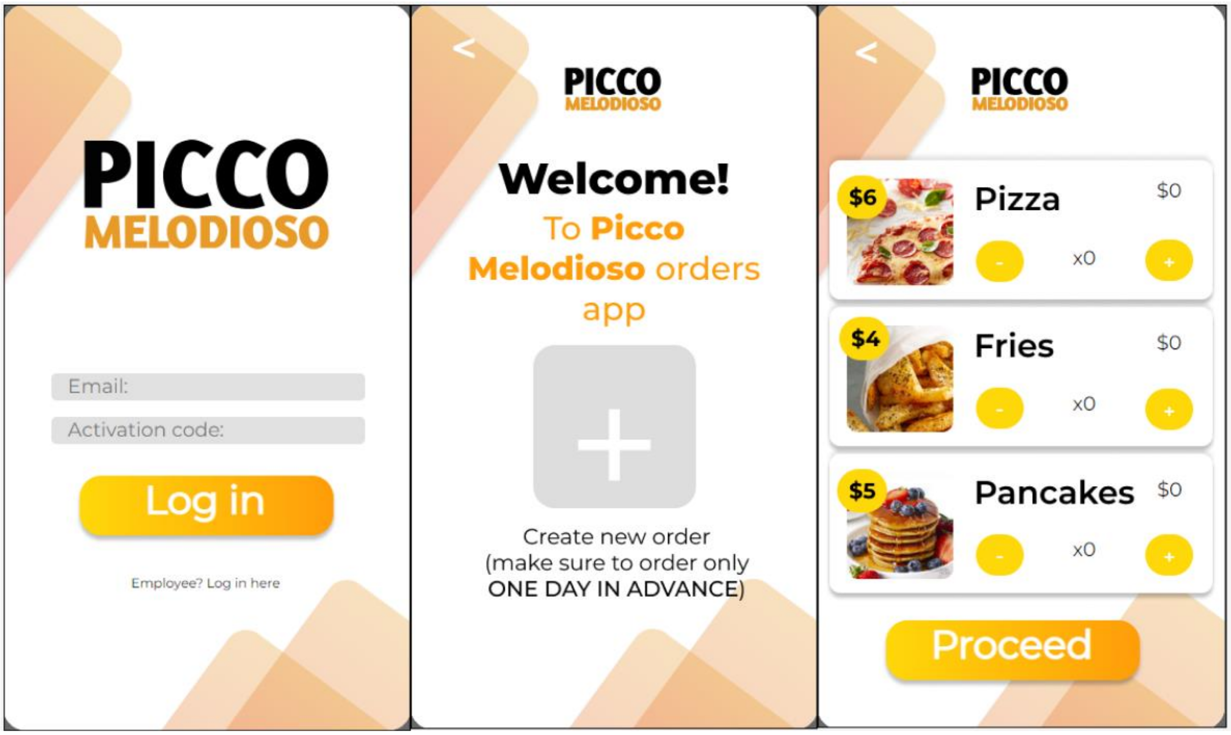
#000000

The application design was done using Figma and all prototyping was already implemented in there. The positioning of buttons and frames on the screen were chosen in order to make a good-looking interface and not complicated user experience. Colors and gradients are highlighting the important and interactive parts.

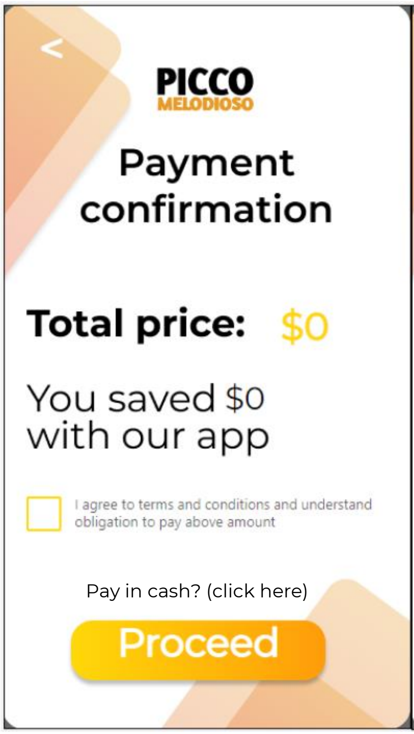
The initial versions of the app design included the feature of choosing the language and the ability to pay using IDEAL (which would be the most convenient in the Netherlands). During the development process it was decided that these features rather not be implemented.

Below, the final design results of the screens are shown.

1. On the first login screen, user is able to log in with their email address and the activation code they get on their email address after purchasing a ticket. This screen also has an employee login button.
2. After successfully completing the login, the next screen is displayed to the user, that allows them to place the order. The condition that the order should be placed one day in advance is also visible.
3. The next screen lets the customer see the prices and add the meals of choice. The user-friendly interface lets simply add or remove meal units, where the total prices for each are displayed.



4. After completing the order, the screen with the total price, how much money is being saved by using the app and term and conditions confirmation are shown.
5. The next screen provides the user with their order number and basic additional information.
6. Proceeding to the bank details page, the user gets all required information in order to place and pay for the order.



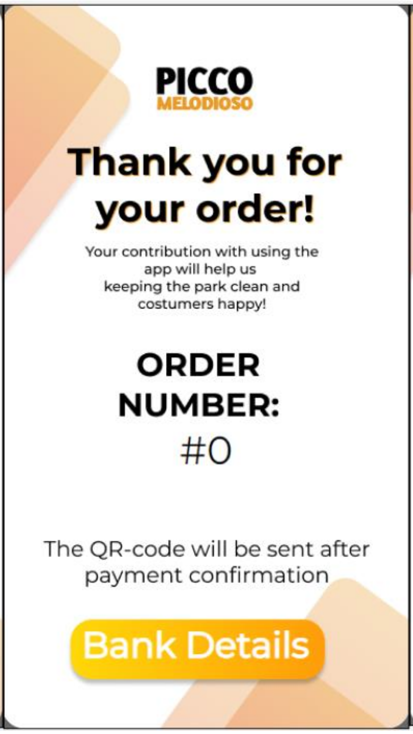
Payment confirmation

Total price: $0

You saved $0 with our app

□ I agree to terms and conditions and understand obligation to pay above amount

Pay in cash? (click here)

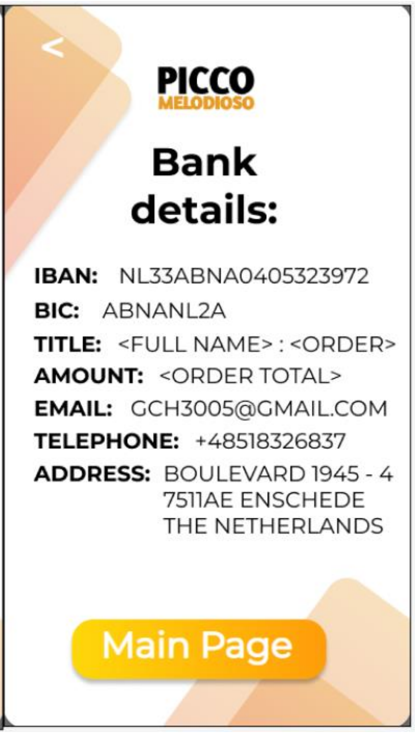


Thank you for your order!

Your contribution with using the app will help us keeping the park clean and costumers happy!

ORDER NUMBER:
#0

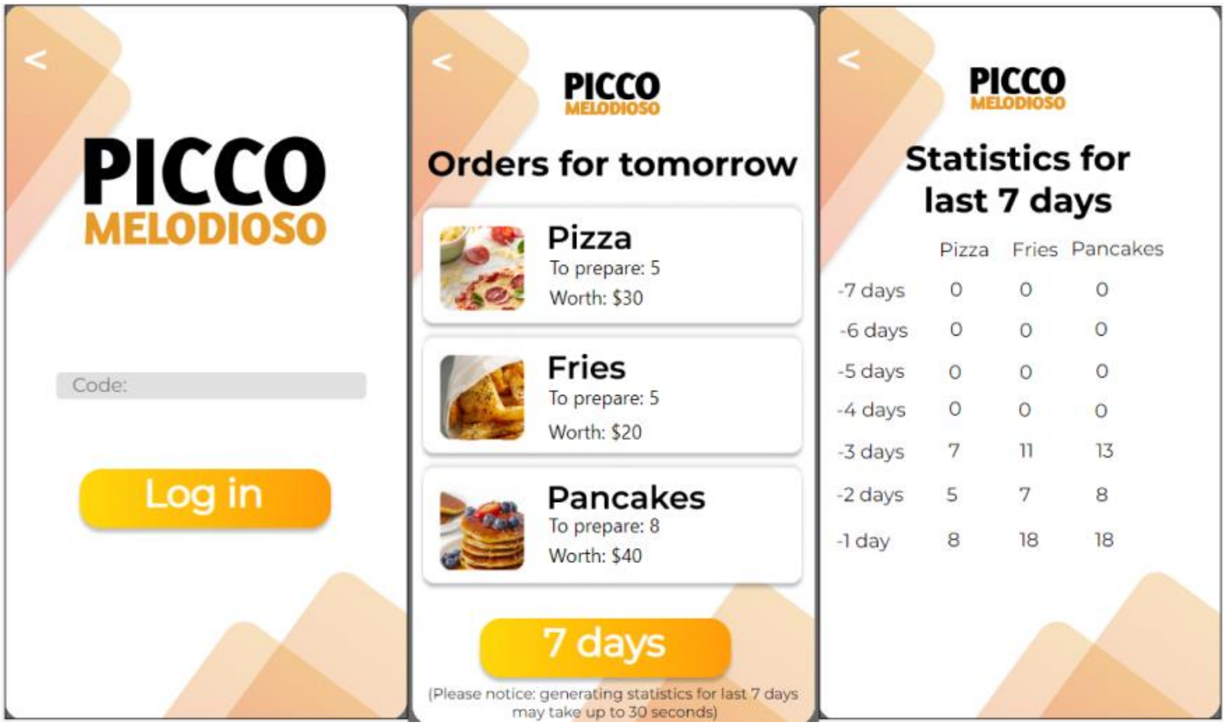
The QR-code will be sent after payment confirmation



Bank details:

IBAN: NL33ABNA0405323972
BIC: ABNANL2A
TITLE: <FULL NAME> : <ORDER>
AMOUNT: <ORDER TOTAL>
EMAIL: GCH3005@GMAIL.COM
TELEPHONE: +48518326837
ADDRESS: BOULEVARD 1945 - 4
7511AE ENSCHEDE
THE NETHERLANDS

7. If the employee login option is chosen, a unique code is required.
8. Next screen provides employee with gathered data regarding next day's orders. The amount of orders placed and the total worth for the ingredients is shown. There is also an option to proceed with the last 7 days' statistics.
9. The last screen displays the last 7 days' statistics for each kind of meal.



Orders for tomorrow

Pizza
To prepare: 5
Worth: $30

Fries
To prepare: 5
Worth: $20

Pancakes
To prepare: 8
Worth: $40

(Please notice: generating statistics for last 7 days may take up to 30 seconds)

Statistics for last 7 days

	Pizza	Fries	Pancakes
-7 days	0	0	0
-6 days	0	0	0
-5 days	0	0	0
-4 days	0	0	0
-3 days	7	11	13
-2 days	5	7	8
-1 day	8	18	18

Application Development

Description of how you used Thunkable

The app development started with in-depth research. Each screen development usually starts in the same way. Firstly, the final design is being made in Figma with the right proportions and attention to detail. Then, components such as buttons and labels are being added. All of them need adjustment and proper placement. After that, developers head into the "Blocks" section where all algorithmic design takes place and other functionalities are being implemented.

Description and motivation of the main technical features of your application

The app uses labels, buttons, input blocks, checkbox, databases and alerts.

Labels

This component is usually used in an extraordinary way. Namely, its original usage is to add text to the screen. In the "Picco Melodioso" app it serves a button function most often. This is due to the team's willingness to implement custom buttons with gradient which makes it more readable and user-friendly.

Buttons

Buttons used in app appear only on Screen 3, since plain yellow was originally there in the design. Furthermore, they have a more precise click area, and it is essential in such small components.

Input blocks

Input blocks are used to get an input from the customer on Screen 1 and Screen 6. They are responsible for taking customers' email or employees' code number, respectively.

Checkbox

Checkbox is used only once on Screen 4 and plays an essential role checking if user agreed to terms and conditions. It makes the user obliged to pay for his order.

Databases

Database is an essential component in the application. It allows to store data there, regardless of app usage. Using this "Picco Melodioso" keeps track of orders, customers and other useful information.

Alerts

This component displays an alert whenever it is needed. It sometimes happens that the user does something unexpected, and alerts let him know what is wrong and needs to be done.

Detailed description of the algorithms behind the most important functions of your app

The “Picco Melodioso” app uses several algorithms, mostly related to databases. Below are descriptions of the most relevant of them:

Sign in

An algorithm iterates through customers/employees database ‘number of rows’ times. It starts with setting index value to 1 and loops through it searching for the value given by user input. If it was successfully found, the algorithm sets app value ‘isValid’ to 1, otherwise it stays 0. It depends on whether the user can go further, or an error will be displayed.

Setting up DayID

This algorithm iterates through days database ‘number of rows’ times. It starts with setting index value to 1 and loops through it searching for the current date. As long as the database is always up to date, there should be no error and app variable ‘dayid’ should be always assigned to corresponding number.

Calculating statistics for tomorrow

In order to calculate statistics for tomorrow, an algorithm iterates through orders database ‘number of rows’ times. It starts with setting the index value and amount value to 0. After that it increases index value by one and loops through database searching for current day’s pizza/fries/pancakes order. Later, it adds orders to corresponding amount values. At the end it sets ‘amount label’ and ‘worth label’ to appropriate values.

Weekly statistics

Description of this algorithm is based on fries since it is the same for pizza and pancakes.

The Fries algorithm iterates through orders database ‘number of rows’ times. It starts with setting index value and amount value to 0. After that it increases index value by one and loops through database searching for ‘essa’ day’s fries order. ‘Essa’ is given when function is called in ‘fries7’ function. After that corresponding label is set to amount value and ‘essa’ is decreased by 1. The whole process repeats until ‘essa’ is equal to 1.

Adding and subtracting order size

This function was designed in a very specific way. Developers wanted to update the amount of each meal with every click so whenever + or – button is clicked, the meal amount changes by 1 and sets corresponding labels to current order size. Furthermore, the app automatically counts the total sum of each meal by increasing/decreasing it by its price on every click.

Savings

An algorithm simply multiplies each meal’s size by its original price and sums up all of them. Then savings are calculated in the following way: price_without_discount – discounted_price.

Order number generator

An algorithm simply takes ‘OrderNumber’ from the last row in ‘order’ database, adds one and displays a result.

Detailed description of how you have tested the application, the problems you have encountered during testing and the applied solutions

The app has run through several tests. It was used by developers in the first place and a few problems were encountered.

First problem

The most difficult one was to find out the way to display the weekly statistics. Very first idea was to simply subtract the device's current day by 7,6,5,4,3,2,1 respectively but it works only if the device's current day is between 8th and 30th day of a month. Brainstorming has been done and developers came up with the solution. Every day has its own 'DayID' and it allows an algorithm to easily subtract days since the lowest 'DayID' is 87.

Second problem

Another enormously big problem was loading the image between Screen 7 and Screen 8. Developers tried their best to implement it following different tutorials, but it came up as it is impossible. The solution has been discussed and it ended up on a small notice at the bottom of a page that loading to the next screen might take a while.

Third problem

The last problem was with a screen three. It came out that after going back from Screen 9 to Screen 2 and then Screen 3 displayed values were not set up to 0 again. The developers decided to set all variables to 0 whenever Screen 3 opens.

Conclusion

The core problem that has been identified is poor meal planning. For the past few weeks several reports have been made and submitted. Throughout these projects a solution of the core problem has been identified. Three potential solutions were made up to solve the problem of poor meal planning. After careful consideration the team decided that giving the potential solutions weights is the best way to see which of the three is the best. Creating an app that tackles the problem of poor meal planning came out on top. The app will help to reduce several problems including the core problem.

In the app users can order meals that they want to eat the next time they visit the theme park Picco Melodioso. Customers have until a maximum of 24 hours in advance to order the food of their choice. This way, the queueing times in the restaurant are not long at all because people are only queueing to pick up their food that already has been made by the kitchen staff. The staff knows what to make because of the orders people have been placing through the app. Because most people that are going to eat at the restaurant will order through the app, the restaurant knows what most of their customers want to eat already. They do not know the orders of people who did not order on the app. Because the cooks mostly know what meals they have to prepare for the day, the right amount of ingredients can be ordered. This will have the result that the restaurant in the theme park Picco Melodioso reduce their food waste. This is because the amount of meals leftover at the end of the day will decrease. This also lowers the overall costs because the restaurant also paid for the thrown away meals. The theme park encourages customers to order through the app with a discount given if place your order in the app.

The team is confident the app will have a great effect on the theme park Picco Melodioso. This is because the app has great details and it is properly tested and debugged multiple times with the help of multiple people. Lastly, this will cover the low revenue problems of the park.

Appendix A: User Manual

The database already has some orders in it due to testing purposes, the new orders will add up.

It is not recommended using the app on a mobile device as every resolution is different and it will not look as intended.

Login details

Public link to the working app:
https://x.thunkable.com/copy/8a8f29d41e64d370aa2a2e764ee257fe

For logging in, there are options for both customers and staff.

Accounts and activation codes stored in the database for logging in as customers:

customer1@testing.com//123456

customer2@testing.com//111111

customer3@testing.com//123123

Employee codes stored in the database for logging in as an employee:

123456

222222

333333

Application screens

1. Customer log in page

2. Employee log in page

3. Daily statistics of purchases

4. Weekly statistics of purchases

5. Creation of new order screen

6. Product selection screen

7. Payment proceeding screen

8. Order confirmation

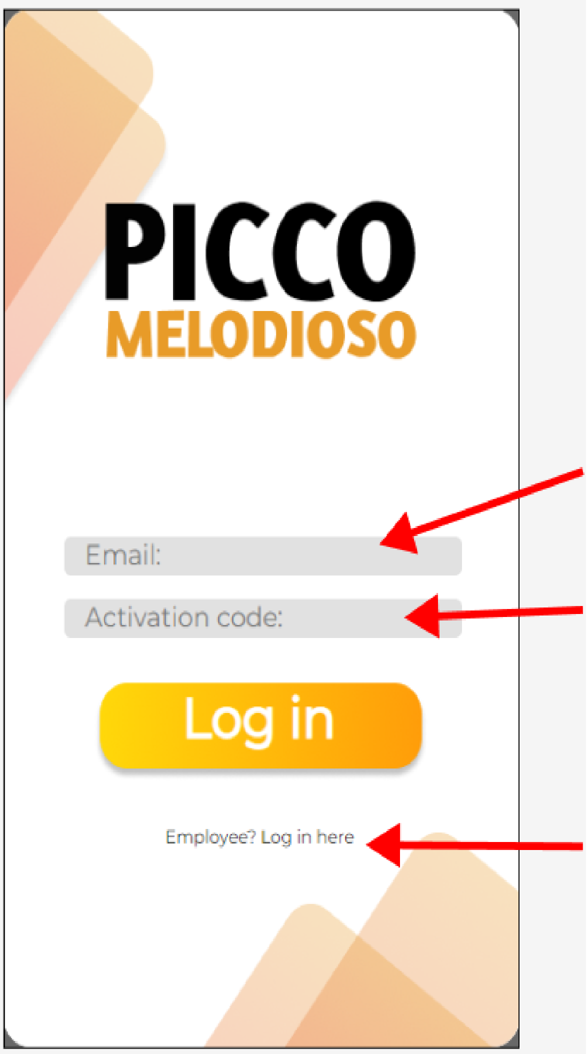
9. Purchase details

Features explanation

1. Customer log in page

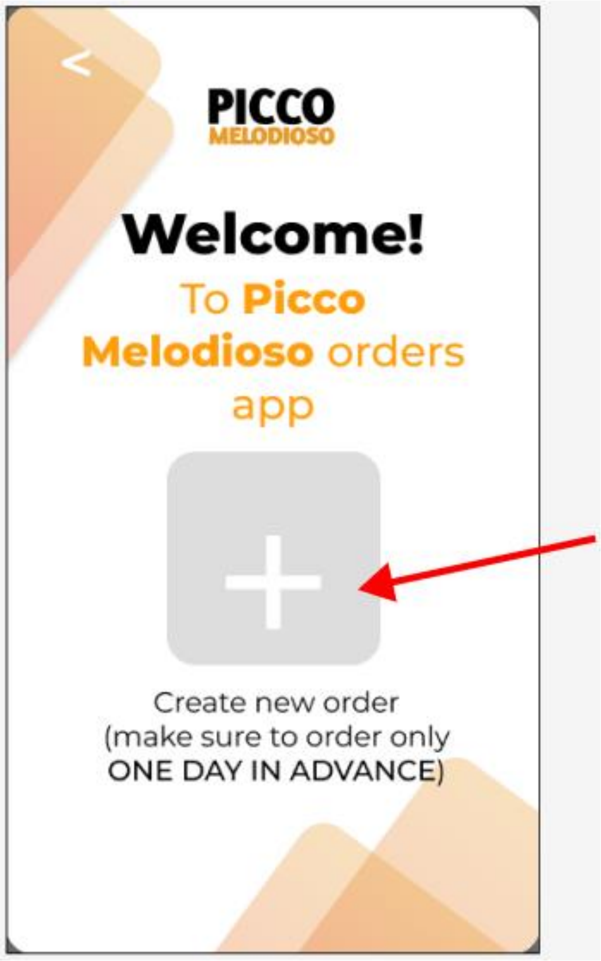
Because of the first time opening the app, the user is directed to the log in screen. Only if the user is not an employee, will it be asked to insert the email and the activation code. If the credentials match, the user will be redirected to the creation of a new order screen.

If the app is used by an employee, there is the option to connect as an employee, pressing the button that redirects the user to the respective page.



2. Creating a new order screen

When entering the welcome screen as a customer, a large “+” button is displayed for creating a new order, which would redirect the user to the product selection screen. Also, a button for going back to the login page is displayed.



Welcome!

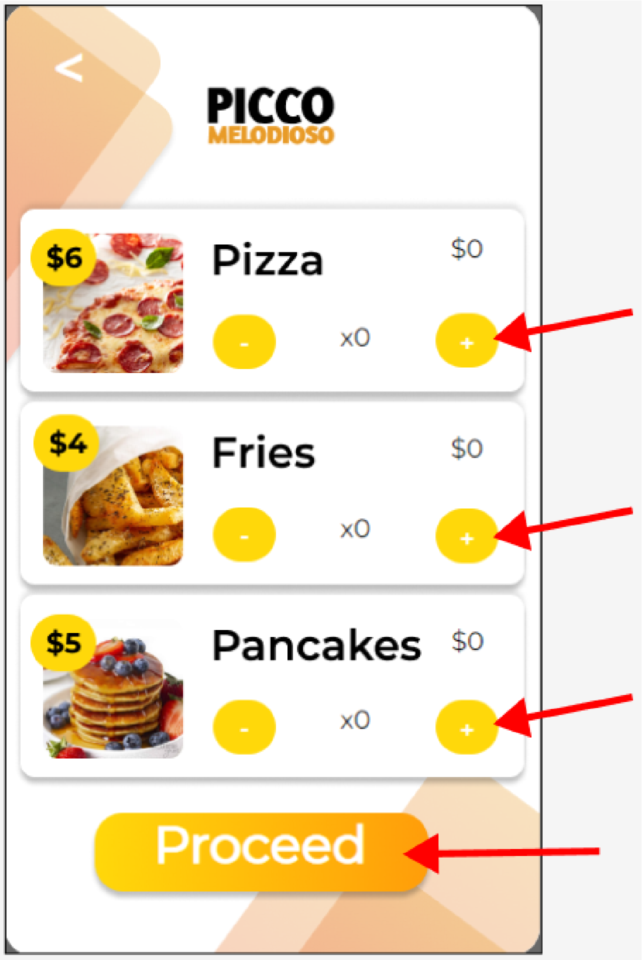
To Picco Melodioso orders app

Create new order (make sure to order only ONE DAY IN ADVANCE)

3. Meal selection screen

The menu options are displayed on this screen. Each product has a “-“ and “+” button, designed for setting the number of pieces of each product wanted. For each product the total cost is displayed, depending on the amount selected. When ready, the user will press the “Proceed” button, also showing the total sum of the order. When pressed, this button will redirect to the payment proceeding screen.

In the upper left corner of the screen there is shown the button for going back to the creation of the new order screen.

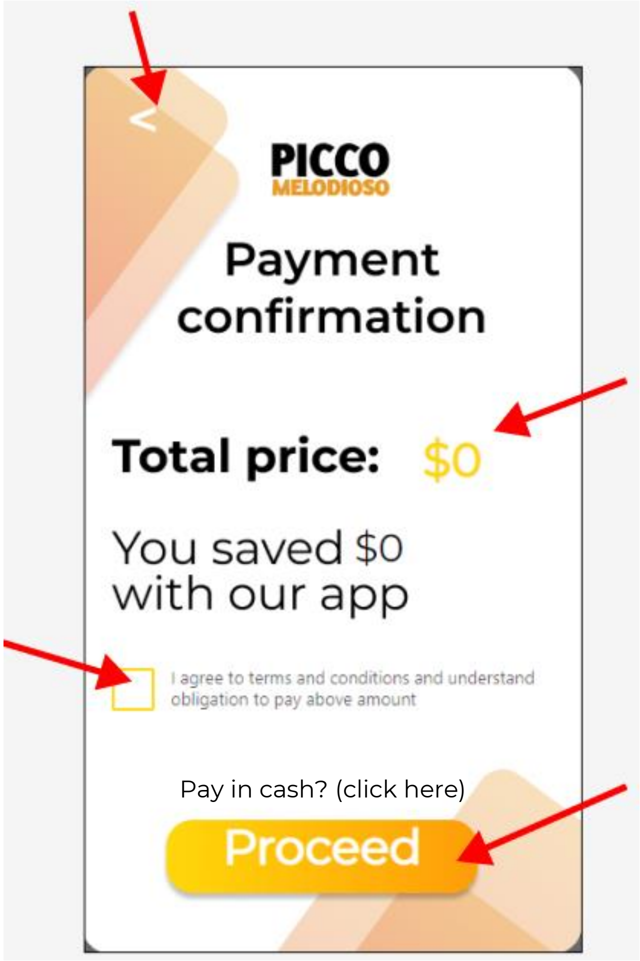


4. Payment proceeding screen

Proceeding the payment consists of user checking the box for agreement of terms and conditions, also for understanding obligation to pay displayed amount representing the total cost of the order. Once the box is checked, the user can choose if they want to pay in cash at the restaurant or pay online in the app. After choosing they will be redirected to the order confirmation page.

If the agreement box is not checked, a popup will be displayed reminding that the terms and conditions are not agreed by the user.

Once again, a button redirecting to the previous page is shown, in the upper left corner.



Payment confirmation

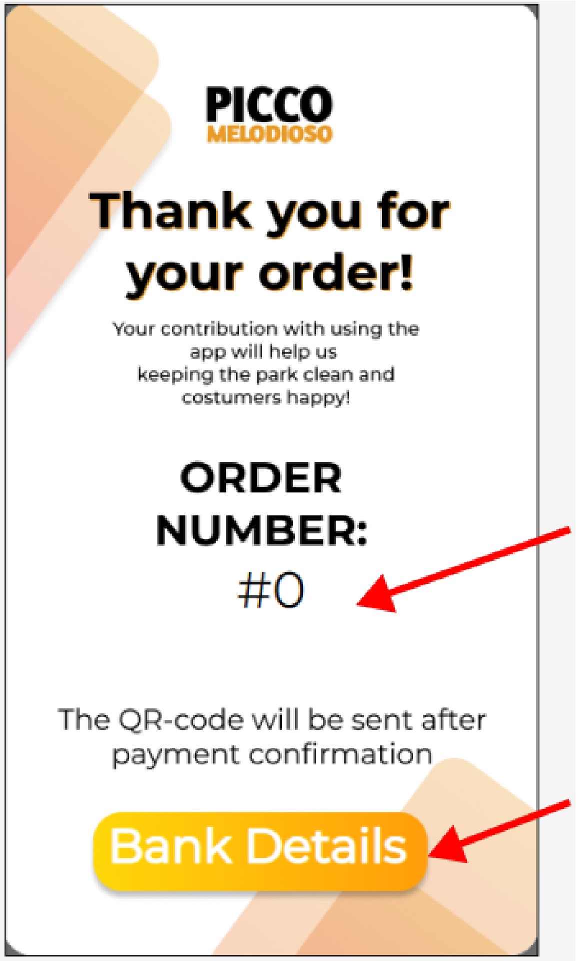
Total price: $0

You saved $0 with our app

□ I agree to terms and conditions and understand obligation to pay above amount

5. Order confirmation

After the order confirmation is entered, a message thanking the customer is displayed, along with a confirmation number, and a button redirecting to the purchase details screen. If the customer choose to pay in cash at the restaurant, the “Bank Details” button will not be visible and this is the final page of the app.



PICCO MELODIOSO

Thank you for your order!

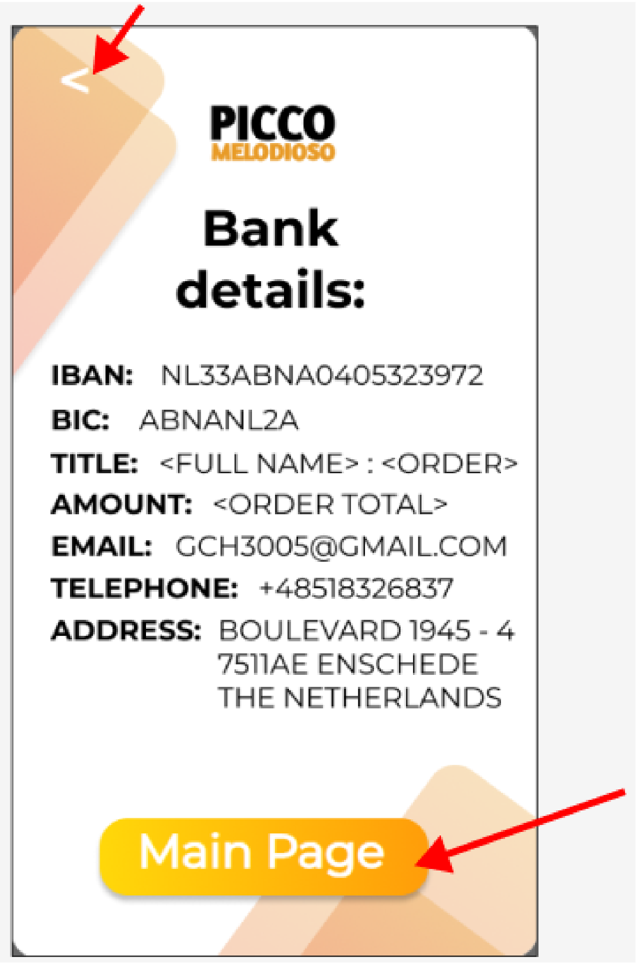
Your contribution with using the app will help us keeping the park clean and costumers happy!

ORDER NUMBER: #0

The QR-code will be sent after payment confirmation

6. Purchase details

Finally, the user can see the bank details of the bank account used for pressing the order. Here, the customer has the option to go back to the main page if considering placing another order. There is also the option to go back to the order confirmation screen, by clicking the button represented by the less-than sign.



Bank details:

IBAN: NL33ABNA0405323972

BIC: ABNANL2A

TITLE: <FULL NAME> : <ORDER>

AMOUNT: <ORDER TOTAL>

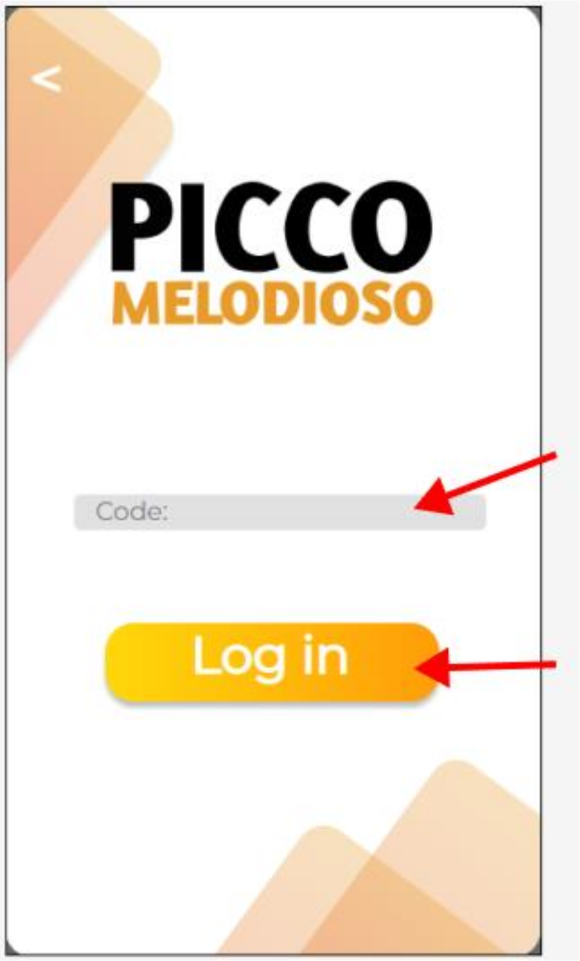
EMAIL: GCH3005@GMAIL.COM

TELEPHONE: +48518326837

ADDRESS: BOULEVARD 1945 - 4
7511AE ENSCHEDE
THE NETHERLANDS

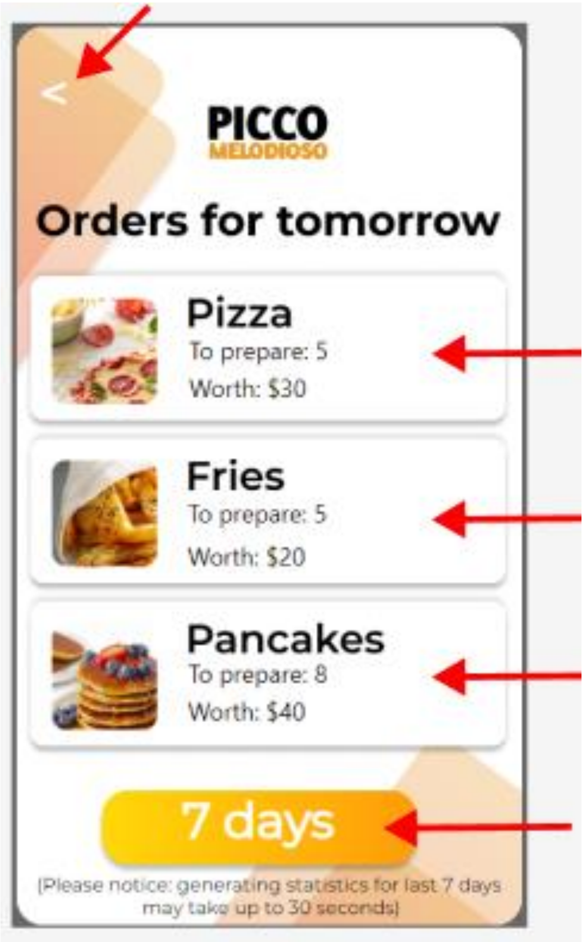
7. Employee login page

The application has a different use assigned to the workers of the theme park. For this reason, there is a separate login page assigned for the employees. Once a valid employee code is inserted, the redirection will go to the daily statistics of purchases screen.



8. Next day's orders

On this screen there the statistics are being displayed of the meals sold on the actual day. Both the quantity of a product sold and the amount in dollars of each meal on the menu is being showed. The user has the option to go back to the login page, or to see the statistics for the last 7 days, entering the weekly statistics of purchases screen.



Orders for tomorrow

Pizza
To prepare: 5
Worth: $30

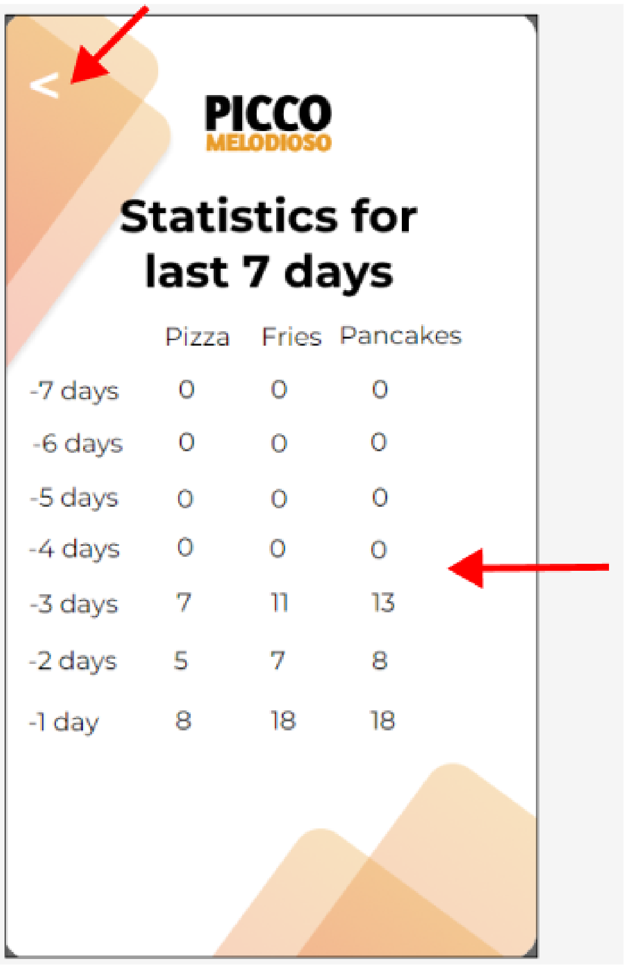
Fries
To prepare: 5
Worth: $20

Pancakes
To prepare: 8
Worth: $40

(Please notice: generating statistics for last 7 days may take up to 30 seconds)

9. Last 7 days statistics

On the weekly statistics of purchases screen, there are shown the number of each product on the menu sold in every of the last 7 days to the actual moment. The user also has a button to be redirected to the daily statistics of purchases screen.



Statistics for last 7 days

	Pizza	Fries	Pancakes
-7 days	0	0	0
-6 days	0	0	0
-5 days	0	0	0
-4 days	0	0	0
-3 days	7	11	13
-2 days	5	7	8
-1 day	8	18	18

Appendix B: Group Reflection

From the start of working together as a group, everyone had their goal clear – to complete all the assignments near to perfect. At this time in the project a team contract was made. This covered the most important agreements that the team made up. These are well executed by the team throughout the projects. From time to time, it was not that clear in the beginning how to split tasks in order to save time and let everyone make their contribution to the report, so sometimes one person was writing down the whole group's brainstorming.

After some time of working as a team, every member got a specific role, to which we tried to stick until the end of the projects. Although each person had their own responsibility in every assignment, they could always ask the others for help, in case they were facing any difficulties.

Every week the time we dedicate for the project assignments differed, as of the amount of meetings. In case one of the group members missed the meeting they would be informed regarding all the important things that were discussed.

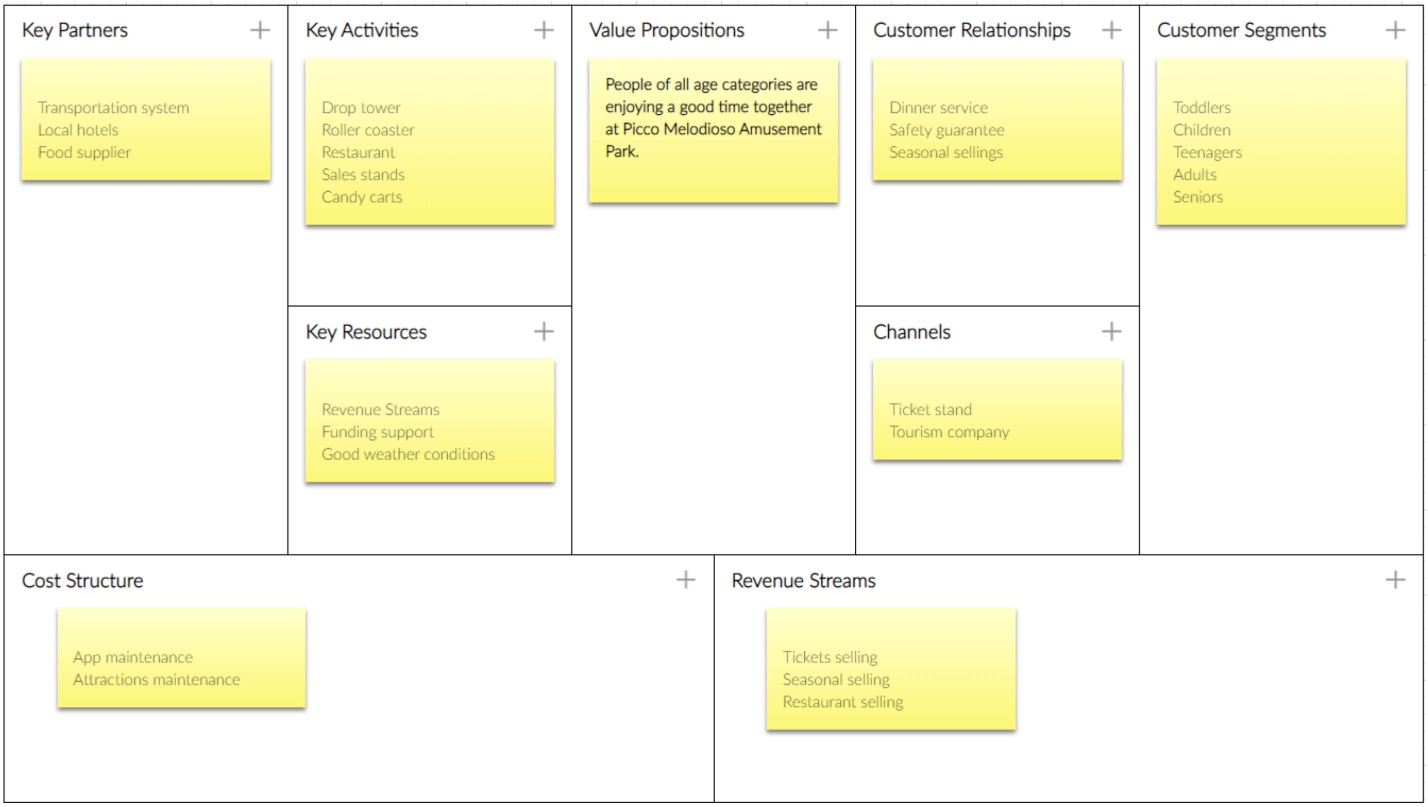
Even though there appeared some difficulties in communication and times when people were being less productive, the team was able to overcome those. This was done by discussing the problems with the team and the tutor, and staying consistent in working as a team.

Clearly, everyone got great experience working on such case, enlarging the knowledge in both the business and IT field. Working as a team gave us a lot more opportunities and with teamwork, we were able to achieve more than all the expectations.

Looking back there are some fields on what we can improve on. Throughout the projects some group members did more than the other ones. We would change and improve the communication and the planning.

Appendix C

C1 Osterwalder Business Model



Key Partners +

Transportation system
Local hotels
Food supplier

Key Activities +

Drop tower
Roller coaster
Restaurant
Sales stands
Candy carts

Key Resources +

Revenue Streams
Funding support
Good weather conditions

Value Propositions +

People of all age categories are enjoying a good time together at Picco Melodioso Amusement Park.

Customer Relationships +

Dinner service
Safety guarantee
Seasonal sellings

Channels +

Ticket stand
Tourism company

Customer Segments +

Toddlers
Children
Teenagers
Adults
Seniors

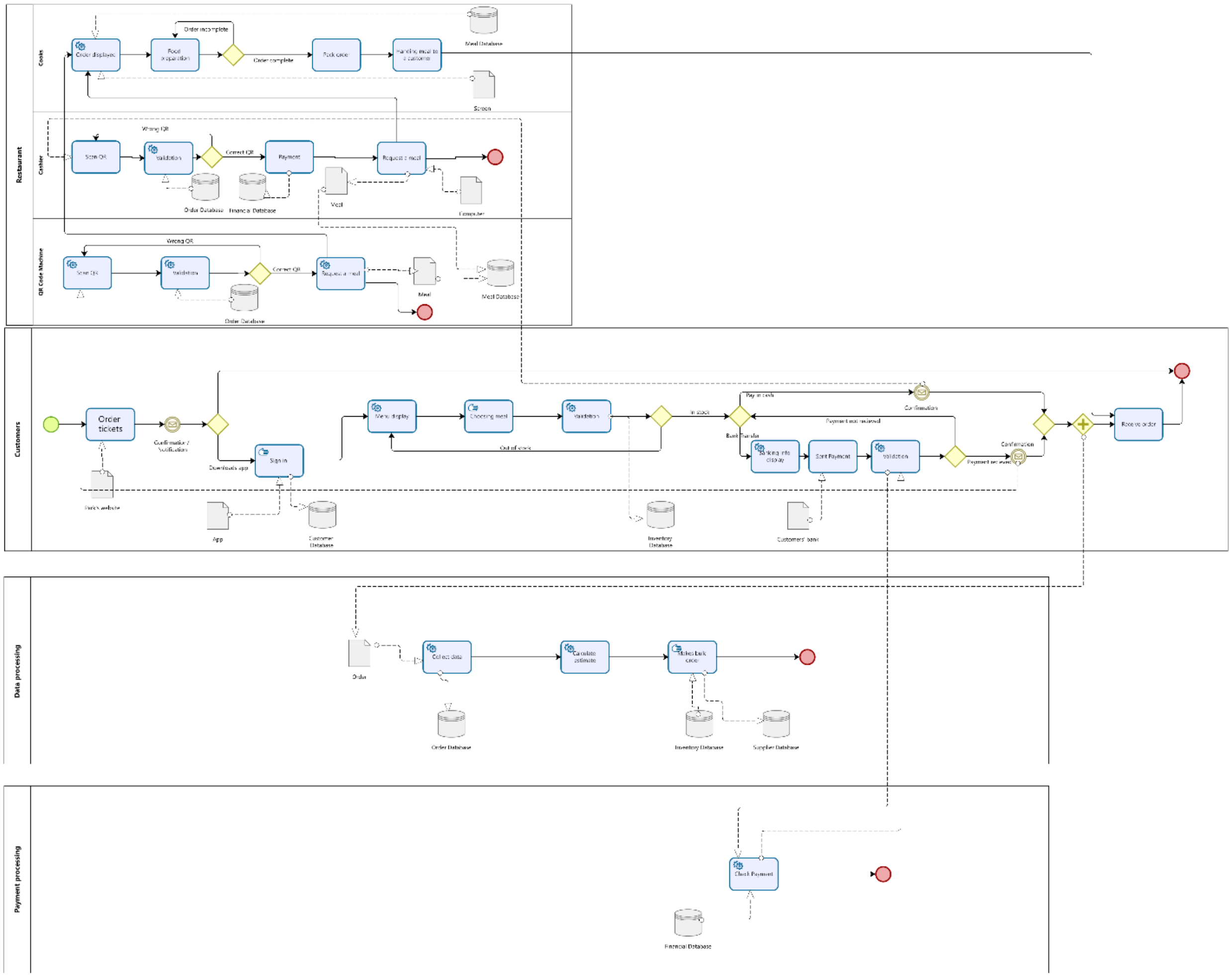
Cost Structure +

App maintenance
Attractions maintenance

Revenue Streams +

Tickets selling
Seasonal selling
Restaurant selling

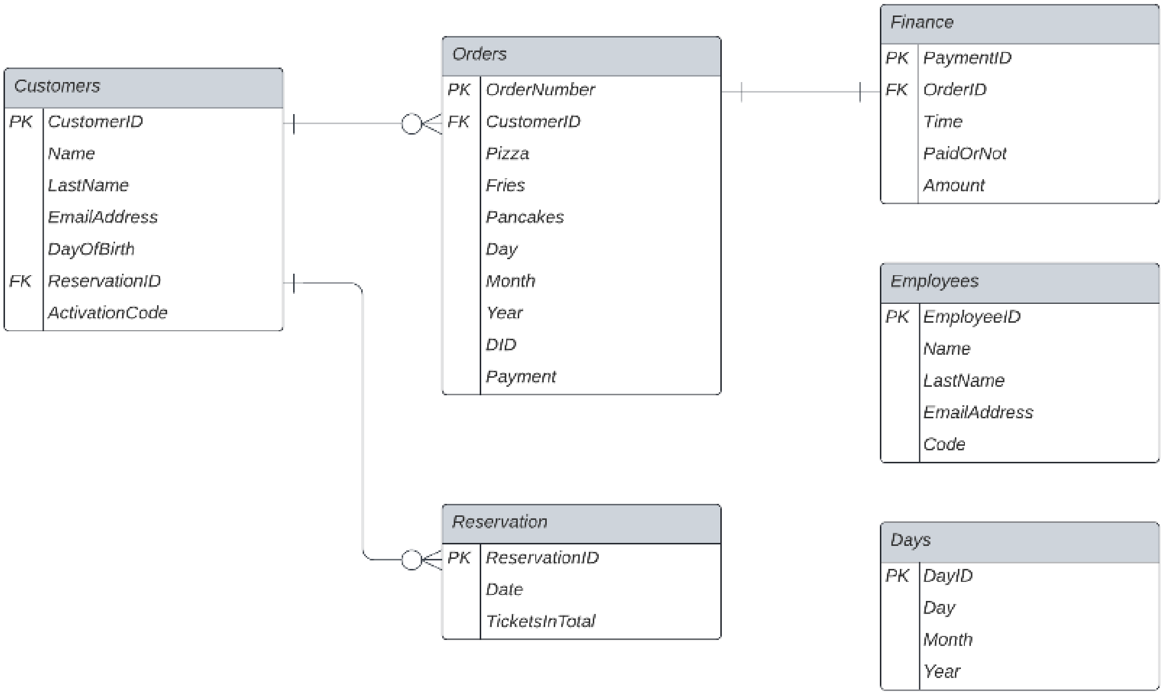
C2 BPMN


Restaurant
Cooks
Order displayed
Food preparation
Order incomplete
Order complete
Pack order
Handing meal to a customer
Meal Database
Screen
Cashier
Wrong QR
Scan QR
Validation
Correct QR
Payment
Request a meal
Order Database
Financial Database
Meal
Computer
QR Code Machine
Wrong QR
Scan QR
Validation
Correct QR
Request a meal
Meal
Meal Database
Order Database
Customers
Order tickets
Confirmation / Notification
Park's website
Downloads app
Sign in
App
Customer Database
Menu display
Choosing meal
Validation
Out of stock
In stock
Inventory Database
Pay in cash
Confirmation
Payment not recieved
Bank Transfer
Banking info display
Sent Payment
Validation
Customers' bank
Confirmation
Payment recieved
Receive order
Data processing
Order
Collect data
Calculate estimate
Makes bulk order
Order Database
Inventory Database
Supplier Database
Payment processing
Check Payment
Financial Database

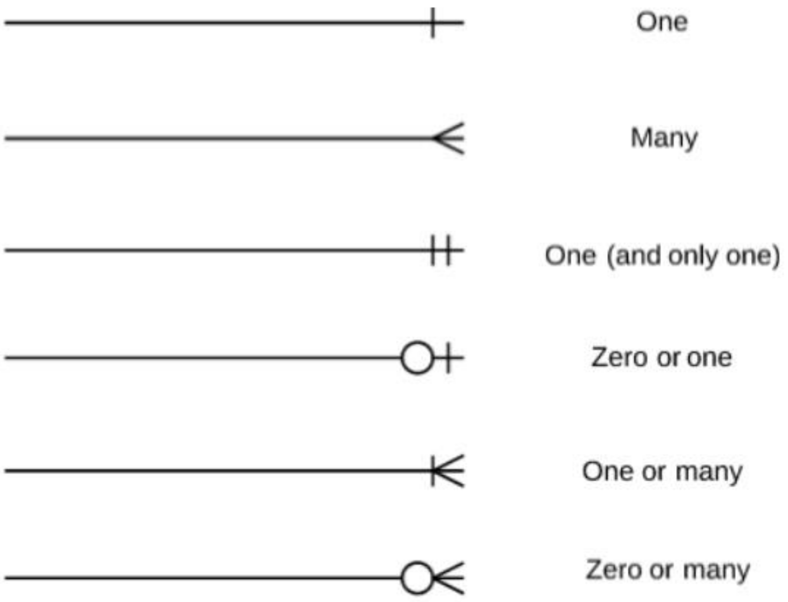
C3 ER Diagram

Database ER diagram (crow's foot)

G CH | November 2, 2022



C4 The crow's-foot notation


One
Many
One (and only one)
Zero or one
One or many
Zero or many