BİLKENT UNIVERSITY

Ticket Event Seller - Project

CS353 / Group 20

Yusuf Toraman - 22002885

Mehmet Burak Demirel - 22003396

Dilara Mandıracı - 22101643

Eren Hayrettin Arım - 22002306

Sümeyye Acar - 22103640

Spring 2024

Description

*Pick**a**Ticket* is an event ticket seller application that provides an efficient and robust interface to both ticket buyers and event organizers. There are three types of users in the system: event organizers, ticket buyers and administrators. Using the application, event organizers can announce events, sell tickets of those events, monitor the events regarding sales and participation. Event organizers can also make modifications and cancel the event before its due date. If an event is canceled then, money is refunded to the ticket buyers. Ticket buyers can discover events, filter and search among various ranges, buy tickets of events, and check their previous transactions. On the management side of the application, admin users can access the admin panel via their exclusive account in order to monitor the events, ticket buyers, organizers and to create reports based on those entities.

Contributions

All group members contributed to all three reports (Proposal, Design Report, and Final Report) and worked as full stack developers. We chose not to divide roles into backend and frontend because our intention was for all members to learn about database queries in the database course project and gain experience with frontend design processes.

Burak Demirel: Contributed to both the frontend and backend of the venue creation and seating plan arrangements. A seating matrix system is developed in order to pick each seat in the venue to both create the existing seats of a venue. Same system is used for selecting seats, reserve them and add them to carts. Also, I have designed a page where user can reserve a ticket while there is no seating arrangement provided.
Dilara Mandıracı: Contributed to both the frontend and backend of the event details and event insights pages. Used various queries to generate statistics, and to create

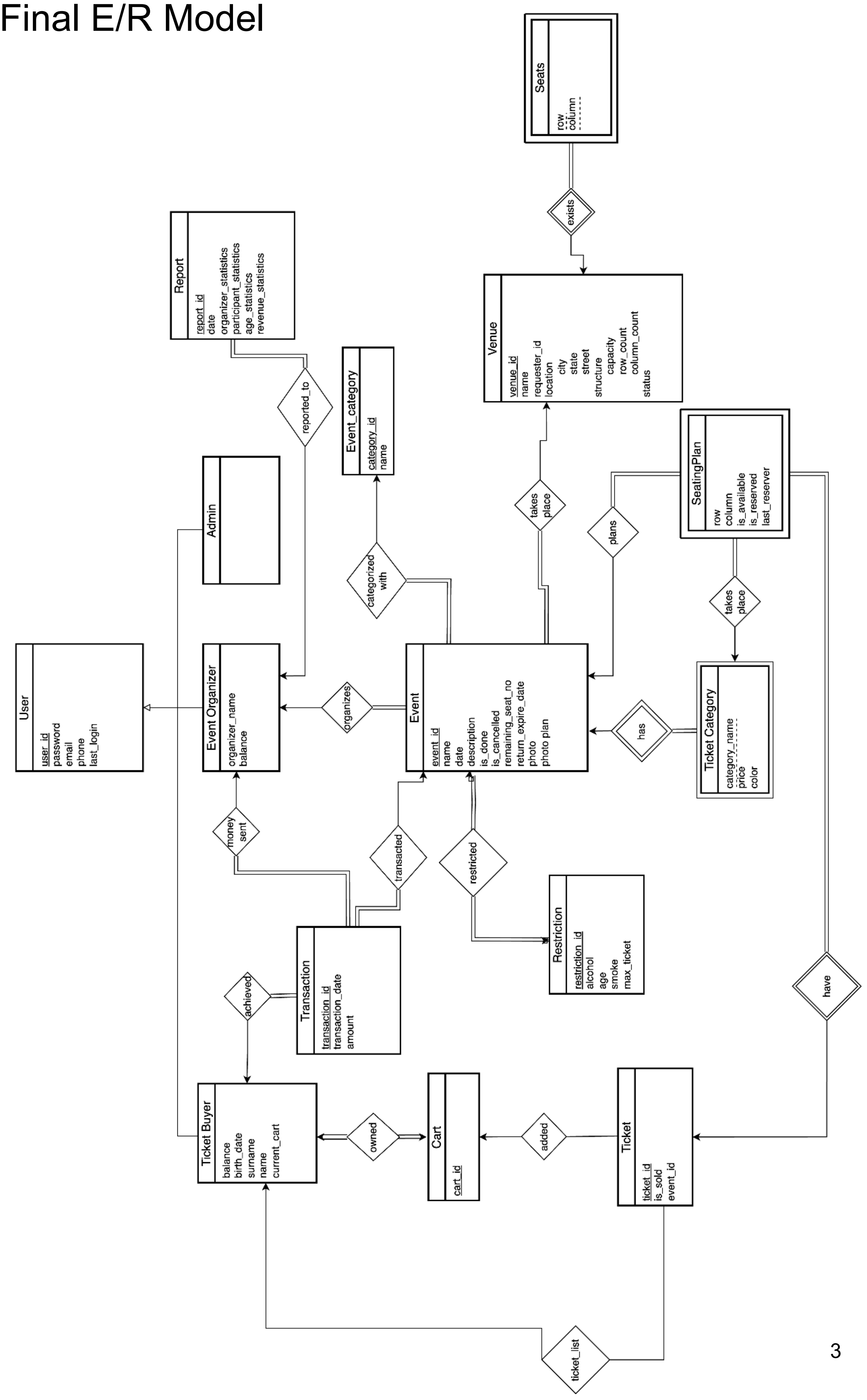
and retrieve all necessary details, tickets, seats, revenues, and participant information for the events. To display these statistics, I utilized different types of charts, tables, etc. On the insights pages, I also managed the cancel event feature by refunding money to the ticket buyers from the organizer's revenue. Contributed to the backend part of the admin panel in terms of listing users, locations, and verifying location requests, etc. Lastly, I contributed to some parts of the report feature by creating statistics (e.g., about revenue, sold/unsold tickets) with backend and database queries.

Eren Hayrettin Arım: Contributed to the creation of frontend designs of all pages and the construction of the general backend structure such as database connections and API structure at the beginning of the project. In the following stages, pages such as the main screen, login, register, admin login and admin panel, where events are listed, were made functional for both backend and frontend. Contributed to the functionalization of creating different types of events and creating seating plans by using seating matrix or photos. Implemented endpoints in the backend and implemented graphics and descriptions in the frontend for the report entity.

Yusuf Toraman: Contributed to both the frontend and backend of the shopping cart page. Used queries and functions to implement triggers and views. Search and filter features on the Main Page were implemented, and events were listed on the frontend accordingly. The frontend and backend of the add balance functionality in the buyer profile were implemented. Finally, I contributed to the writing of certain endpoints that could be beneficial to the project.

Sümeyye Acar: Contributed to both backend and frontend implementation of profile pages of the users and also event organizers. Profile pages are where contact information, events, reports, tickets, venue requests are listed and contain features like returning a ticket, adding to one’s balance, filtering events/tickets regarding the status of the event such as Upcoming, Passed, and Canceled, categorizing venue requests, and directing to reports of the event organizer. Used many different queries to get all these information from the database and implemented the endpoints accordingly. So, with the gathered data I structured a suitable frontend.

Final E/R Model



Final List of Tables

1. **Cart**
    - **Primary Key:** `cart_id`
    - **Columns:** `cart_id`, `is_gift`
2. **Users**
    - **Primary Key:** `user_id`
    - **Columns:** `user_id`, `password`, `email`, `phone`, `last_login`
3. **Ticket_Buyer**
    - **Primary Key:** `user_id`
    - **Foreign Keys:**
        - `user_id` **references** `Users(user_id)`
        - `current_cart` **references** `Cart(cart_id)`
    - **Columns:** `user_id`, `name`, `surname`, `balance`, `birth_date`, `current_cart`
4. **Admin**
    - **Primary Key:** `user_id`
    - **Foreign Key:** `user_id` **references** `Users(user_id)`
    - **Columns:** `user_id`
5. **Event_Category**
    - **Primary Key:** `category_id`
    - **Columns:** `category_id`, `name`
6. **Venue**
    - **Primary Key:** `venue_id`

- **Columns:** `venue_id`, `requester_id`, `name`, `city`, `state`, `street`, `status`, `capacity`, `row_count`, `column_count`

7. **Event_Organizer**
    - **Primary Key:** `user_id`
    - **Foreign Key:** `user_id` **references** `Users(user_id)`
    - **Columns:** `user_id`, `organizer_name`, `balance`
8. **Event**
    - **Primary Key:** `event_id`
    - **Foreign Keys:**
        - `category_id` **references** `Event_Category(category_id)`
        - `venue_id` **references** `Venue(venue_id)`
        - `organizer_id` **references** `Event_Organizer(user_id)`
    - **Columns:** `event_id`, `name`, `date`, `description`, `is_done`, `is_cancelled`, `remaining_seat_no`, `return_expire_date`, `organizer_id`, `venue_id`, `category_id`, `photo`, `photo_plan`
9. **Ticket**
    - **Primary Key:** `ticket_id`
    - **Foreign Key:** `event_id` **references** `Event(event_id)`
    - **Columns:** `ticket_id`, `is_sold`, `event_id`
10. **Added**
    - **Primary Key:** `cart_id`, `ticket_id`
    - **Foreign Keys:**
        - `cart_id` **references** `Cart(cart_id)`
        - `ticket_id` **references** `Ticket(ticket_id)`

- **Columns:** cart_id, ticket_id

11. Transaction

- **Primary Key:** transaction_id
- **Foreign Keys:**
    - organizer_id **references** Event_Organizer(user_id)
    - buyer_id **references** Ticket_Buyer(user_id)
    - event_id **references** Event(event_id)
- **Columns:** transaction_id, organizer_id, buyer_id, transaction_date, amount, event_id

12. Ticket_Category

- **Primary Key:** event_id, category_name
- **Foreign Key:** event_id **references** Event(event_id)
- **Columns:** event_id, category_name, price, color

13. Restriction

- **Primary Key:** restriction_id
- **Columns:** restriction_id, alcohol, smoke, age, max_ticket

14. Report

- **Primary Key:** report_id
- **Foreign Key:** organizer_id **references** Event_organizer(user_id)
- **Columns:** report_id, date, organizer_id, organizer_statistics, participant_statistics, age_statistics, revenue_statistics

15. Seating_Plan

- **Primary Key:** ticket_id

- **Foreign Keys:**
    - `event_id` **references** `Event(event_id)`
    - `event_id`, `category_name` **references** `Ticket_Category(event_id, category_name)`
    - `ticket_id` **references** `Ticket(ticket_id)`
- **Columns:** `event_id`, `ticket_id`, `category_name`, `row_number`, `column_number`, `is_available`, `is_reserved`, `last_reserver`

16. Seats

- **Primary Key:** `venue_id`, `row_number`, `column_number`
- **Foreign Key:** `venue_id` **references** `Venue(venue_id)`
- **Columns:** `row_number`, `column_number`, `venue_id`

17. Owned

- **Primary Key:** `user_id`, `cart_id`
- **Foreign Keys:**
    - `user_id` **references** `Ticket_Buyer(user_id)`
    - `cart_id` **references** `Cart(cart_id)`
- **Columns:** `user_id`, `cart_id`

18. Restricted

- **Primary Key:** `restriction_id`, `event_id`
- **Foreign Keys:**
    - `restriction_id` **references** `Restriction(restriction_id)`
    - `event_id` **references** `Event(event_id)`
- **Columns:** `restriction_id`, `event_id`

19. Ticket_List

- **Primary Key:** `user_id`, `ticket_id`
- **Foreign Keys:**
    - `user_id` **references** `Ticket_Buyer(user_id)`
    - `ticket_id` **references** `Ticket(ticket_id)`
- **Columns:** `user_id`, `ticket_id`

Implementation Details

For the implementation part, we have used the FAST API on the backend side with Python and React with JavaScript and CSS. For the database we used PostgreSQL. FAST API's asynchronous handles facilitated our usage of SQL queries. Both backend and database operations were fast and efficient.

In the Backend and Database part, for ease of use, we created an init.sql and connected it with our terminal, so that once we run our backend it creates tables if they do not exist. In addition to not creating database connection instances on each file, we created a database session.py to access our database. In that session, we used to **psycopg2** library to connect the database and get **connection, cursor** variables. When we need to execute a query on our database it is enough to use our cursor with the SQL statement, then committing the changes (if necessary) or rolling back them with the connection variable. Here you can see an example usage of database queries:

```
cursor.execute(
        "INSERT INTO item (name, description, price, tax) VALUES (%s, %s,
%s, %s) RETURNING id",
        (item.name, item.description, item.price, item.tax)
```

```
    )
    item_id = cursor.fetchone()[0]
    conn.commit()
```

We designed our Graphical User Interface with ReactJS and Ant Design UI Library. The use of the mentioned tools enabled us to come up with this sophisticated, user-friendly interface that not only meets modern usability standards but also eases user engagement through its aesthetic outlook. It was created by combining tables, forms, cards and many others components of antd library with react functions and states. In addition, in many error and success cases, information messages were added as toast messages to ensure that the user was informed.

The constraints discussed in the Advanced Database Components section are managed through role verification during session authentication and are also enforced at the API level to make sure both backend and frontend are working in a harmony. Timestamp comparisons were conducted before processing any request in some constraints. About email and phone validations, we used regular expressions in both the frontend and backend to ensure that the input data met the expected format standards. All together, constraints helped us with robustness of the project, and maintain and enhance security.

Advanced Database Components

Trigger: Update Organizer Balance After is_sold = True

In the application, whenever the is_sold field turns true, the ticket's price should be added to the Organizer's balance. This is because, regardless of whether the ticket is gifted or not, the money from the sold ticket must go to the Organizer. Therefore, a trigger was written to handle this situation. This trigger will execute when is_sold changes to true, find the ticket's price, and add it to the Organizer's balance. Function and the trigger is provided below.

```
CREATE OR REPLACE FUNCTION update_organizer_balance() RETURNS TRIGGER AS
$$
DECLARE
    ticket_price DECIMAL;
    organizer_id UUID;
BEGIN
    SELECT tc.price INTO ticket_price
    FROM Seating_Plan sp
    JOIN Ticket_Category tc ON sp.event_id = tc.event_id AND
sp.category_name = tc.category_name
    WHERE sp.ticket_id = NEW.ticket_id;

    SELECT e.organizer_id INTO organizer_id
    FROM Event e
    WHERE e.event_id = NEW.event_id;

    UPDATE Event_Organizer eo
    SET balance = balance + ticket_price
    WHERE eo.user_id = organizer_id;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

The trigger is created with the following lines, and it runs for each row whenever is_sold = TRUE.

```
CREATE TRIGGER update_organizer_balance_trigger
AFTER UPDATE OF is_sold
ON Ticket
```

```
FOR EACH ROW
WHEN (NEW.is_sold = TRUE)
EXECUTE FUNCTION update_organizer_balance();
```

Insert to Ticket List Trigger for Max Ticket Condition

In this trigger, the number of tickets the user has added to their cart for the events they are trying to purchase is calculated. Then, using the necessary queries, the restriction table is checked to determine the maximum number of tickets that can be offered to one person for that event. The function is triggered before the user presses the buy button and the tickets are inserted into the ticket_list.

```
CREATE OR REPLACE FUNCTION check_max_ticket_limit() RETURNS TRIGGER AS $$
DECLARE
    user_ticket_count INT;
    max_tickets_allowed INT;
    event_id_from_ticket UUID;
BEGIN
    SELECT t.event_id INTO event_id_from_ticket
    FROM ticket t
    WHERE t.ticket_id = NEW.ticket_id;

    SELECT COUNT(*) INTO user_ticket_count
    FROM ticket_list tl
    JOIN ticket t ON tl.ticket_id = t.ticket_id
    WHERE tl.user_id = NEW.user_id AND t.event_id = event_id_from_ticket;

    SELECT r.max_ticket INTO max_tickets_allowed
    FROM restriction r
    JOIN restricted rd ON r.restriction_id = rd.restriction_id
    WHERE rd.event_id = event_id_from_ticket;

    IF user_ticket_count >= max_tickets_allowed THEN
        RAISE EXCEPTION 'Purchase limit exceeded for this event';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

The trigger is created with the following lines, and it runs for each row before insert on ticket_list

```
CREATE TRIGGER check_max_ticket_limit_trigger
BEFORE INSERT ON ticket_list
FOR EACH ROW
```

```
EXECUTE FUNCTION check_max_ticket_limit();
```

Top Revenue View

This view calculates the total revenue for each event and lists the highest-grossing events. The purpose of this view is to monitor and analyze the financial performance of events. By calculating the total revenue obtained from the tickets sold for each event, it allows organizers to see which events are more successful and generate more income. This helps in making more informed decisions in future event planning and marketing strategies. Additionally, by identifying the most successful events in terms of revenue, it aids organizers in using their resources more efficiently.

```
CREATE VIEW top_revenue_events AS
SELECT e.event_id, e.name AS event_name, SUM(tc.price) AS total_revenue
FROM event e
JOIN ticket t ON e.event_id = t.event_id
JOIN ticket_category tc ON t.event_id = tc.event_id
WHERE t.is_sold = TRUE
GROUP BY e.event_id, e.name
ORDER BY total_revenue DESC;
```

Constraints

The constraints within our ticket-selling platform were embedded with different rules to ensure role-based abilities and data integrity. One of the constraints is preventing users from inputting a future birth date in the registration. Similarly, events cannot have a past date as the event dates.

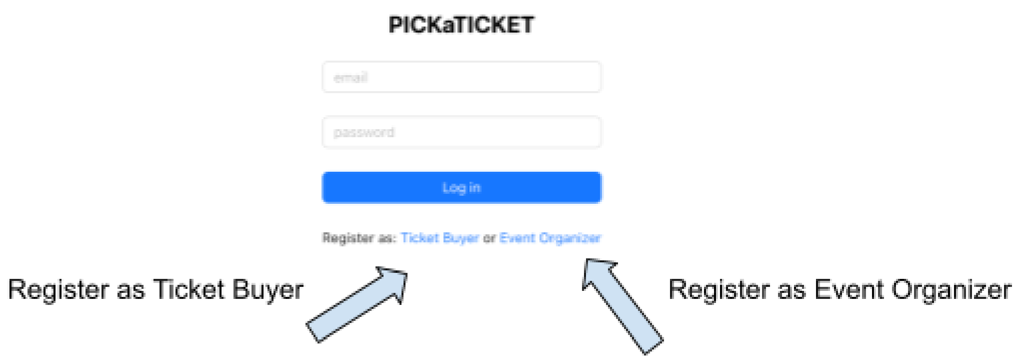
To enforce role-based (event organizer, ticket_buyer, and admin) actions, we performed checks to make sure a ticket buyer cannot create events and an event organizer cannot purchase tickets. These constraints are managed through role verification during session authentication and are also enforced at the API level. Moreover, we ensured that reserved tickets were not sold. This was controlled by a ticket status in the database that was checked before transactions, protecting those tickets from being sold.

Tickets also had an expiration date, which enabled us to prevent refunds once the expiration date had passed. This involved a timestamp comparison before processing any refund request. About email and phone validations, we used regular expressions in both the frontend and backend to ensure that the input data met the expected format standards. These validations helped us maintain the robustness of the project and enhanced security.

User Manual

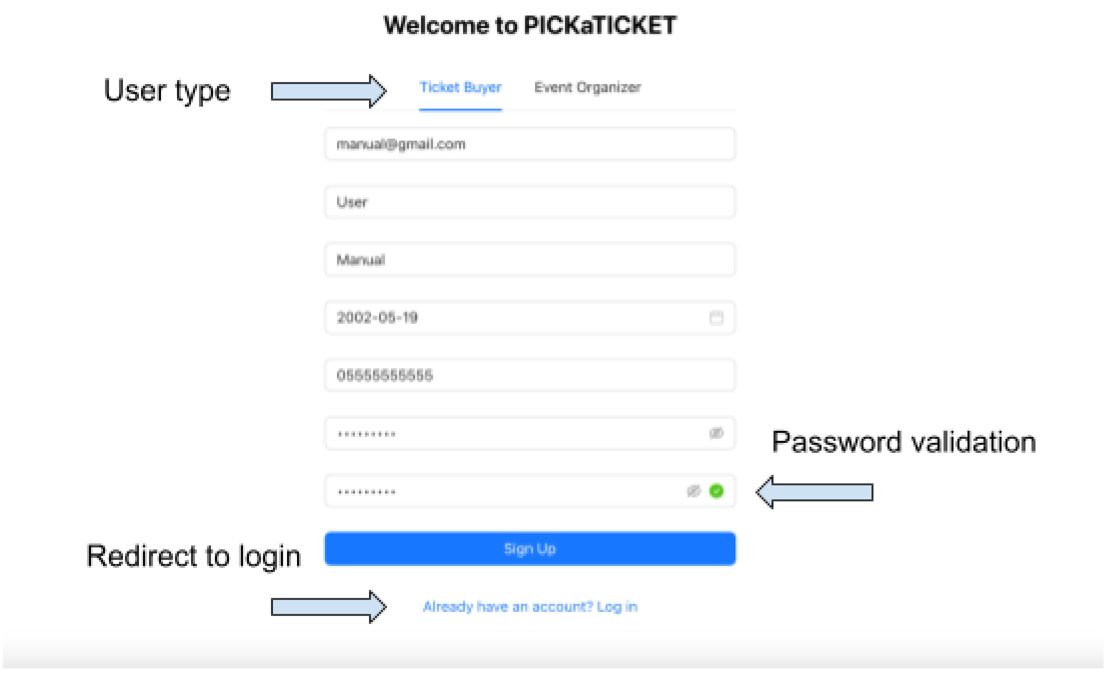
Login

When the user first opens the application, they are greeted by a login screen. If the user already has an account, they can easily log in from this screen. If the user does not have an account, they can create one using the sections indicated by the arrows below. Since there are two different types of users in the application, two different registration areas are visible on the login screen.


PICKaTICKET
email
password
Log in
Register as: Ticket Buyer or Event Organizer
Register as Ticket Buyer
Register as Event Organizer

Register as Ticket Buyer

On the registration page, the user is first presented with a choice of whether they want to register as a Ticket Buyer or an Event Organizer. In the figure shown below, the Ticket Buyer option is selected. Therefore, the form that appears in front of the user contains the fields that a Ticket Buyer needs to fill out when registering. After filling in the required fields, the user can sign up by clicking the Sign Up button. At the very bottom of the page, there is a link to the login page for users who already have an account, allowing them to log in directly.

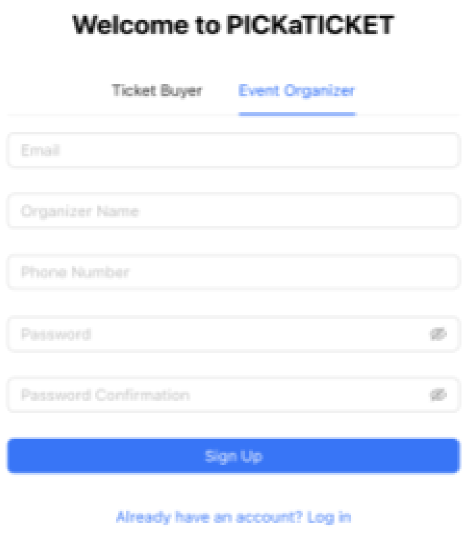


Welcome to PICKaTICKET

manual@gmail.com
User
Manual
2002-05-19
05555555555
•••••••••
•••••••••

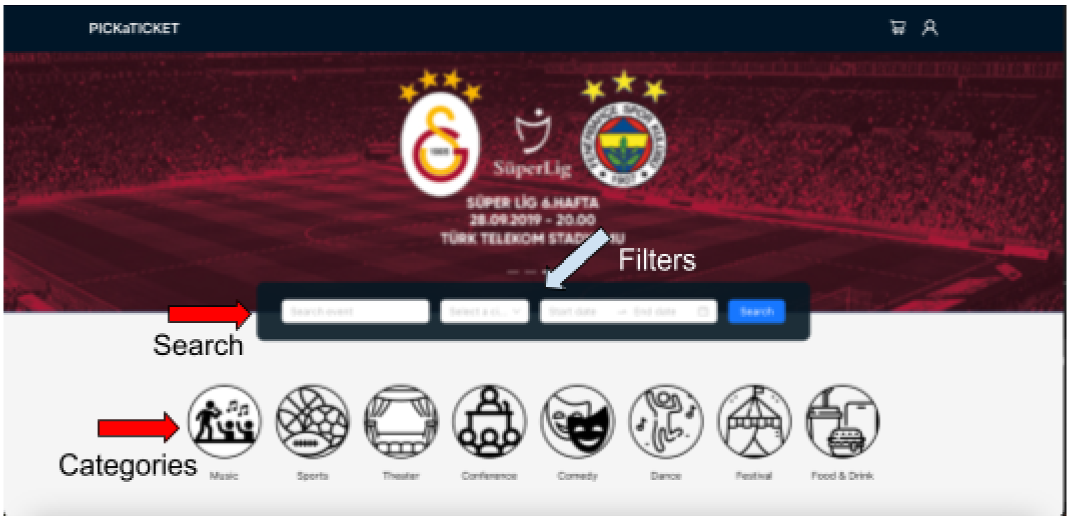
Register as Event Organizer

The Register as Event Organizer page is very similar in structure to the Register as Ticket Buyer page. The only difference between these pages is the input fields that the user needs to enter. For example, the event organizer's date of birth is not relevant to the application, so this field is not included. The fields shown in the image are sufficient.

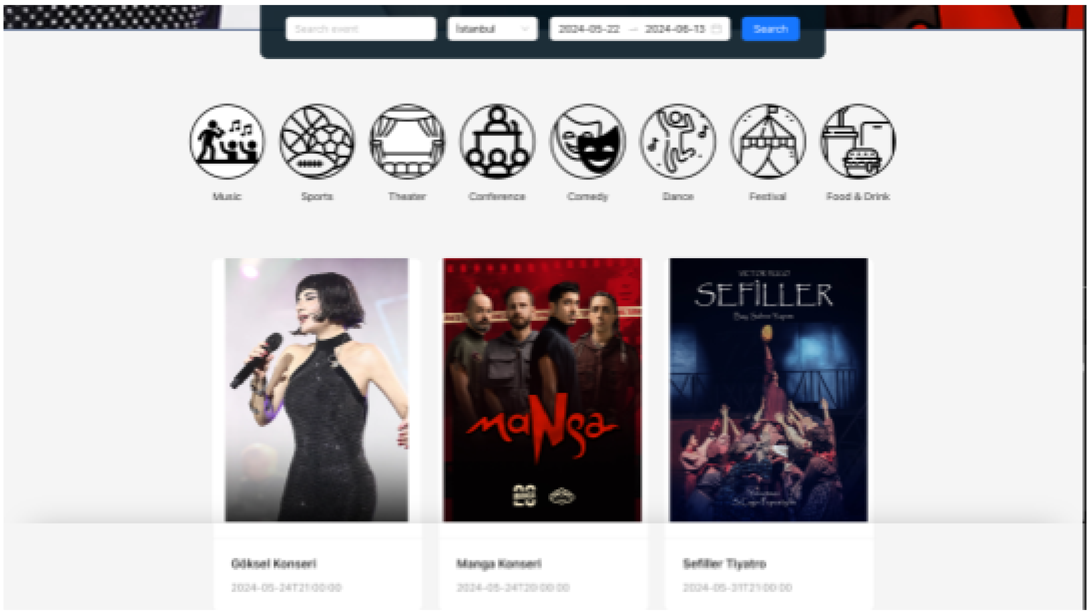


Main Page for Ticket Buyer

After logging in, the Ticket Buyer is greeted by a Main Page as shown in the image. This page lists upcoming events, and the user can click on any event to view its details and purchase tickets. Additionally, the Main Page features search and filter options. As seen in the image, there are input fields for search, city, and date range above the events. Based on the selections made here, pressing the search button filters the events and displays them to the user. Finally, as shown in the image, there is an area with eight categories. The user can select a category from this area, and as soon as they click on the icons, the events are instantly listed. The selected category is highlighted with the help of a UI indicator, and clicking on it again removes the category selection.



In the image below, you can see the filtered list of events after applying the search criteria.



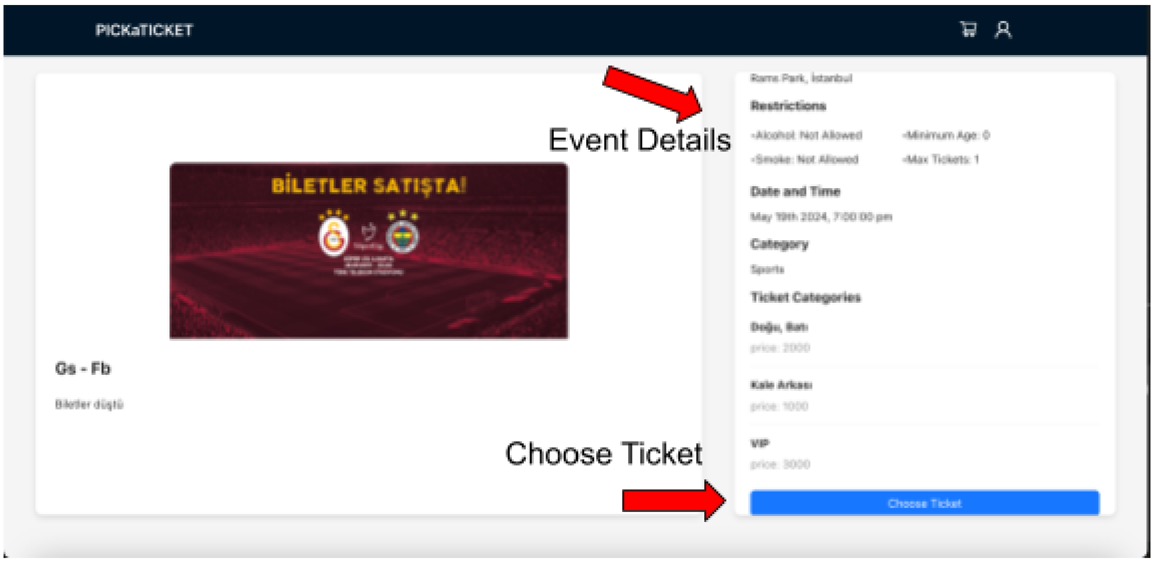
Göksel Konseri
2024-05-24T21:00:00

Manga Konseri
2024-05-24T20:00:00

Sefiller Tiyatro
2024-05-31T21:00:00

Event Detail Page

If the Ticket Buyer clicks on a specific event, they are taken to the event detail page. As shown in the image, all the details about the event are displayed on the right side of the page. At the very bottom, there is a "Choose Ticket" button that allows the user to purchase tickets for that event. When the Ticket Buyer clicks this button, a page opens where they can select their ticket.



Gs - Fb

Biletler düştü

Rams Park, İstanbul

Restrictions

-Alcohol: Not Allowed -Minimum Age: 0

-Smoke: Not Allowed -Max Tickets: 1

Date and Time

May 19th 2024, 7:00:00 pm

Category

Sports

Ticket Categories

Doğu, Batı

price: 2000

Kale Arkası

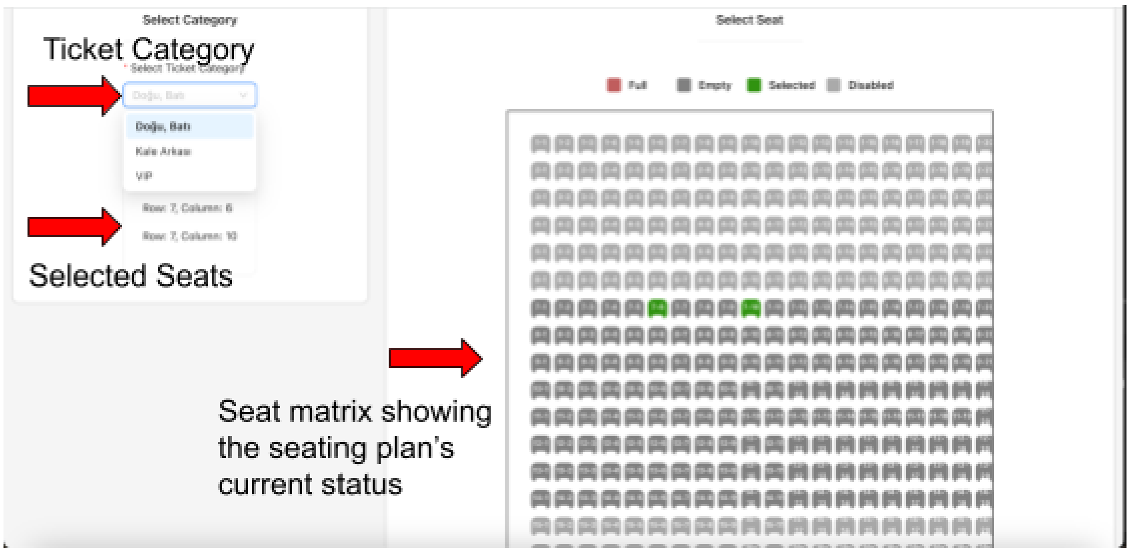
price: 1000

VIP

price: 3000

Choose Ticket Page

On the Choose Ticket page, the event's current seating plan is displayed on the right side, showing the statuses such as available, selected, and disabled. As seen on the left side of the image, the user selects the ticket category they want to purchase. The tickets within this category become available in the seating matrix, and the user can make their selection from there. Just below the category selection, the selected seats are displayed. The number of selected seats cannot exceed the max_tickets defined by the organizer when creating the event. At the very bottom of the page, there is an Add to Cart button, which, when clicked, adds the selected tickets to the user's cart.

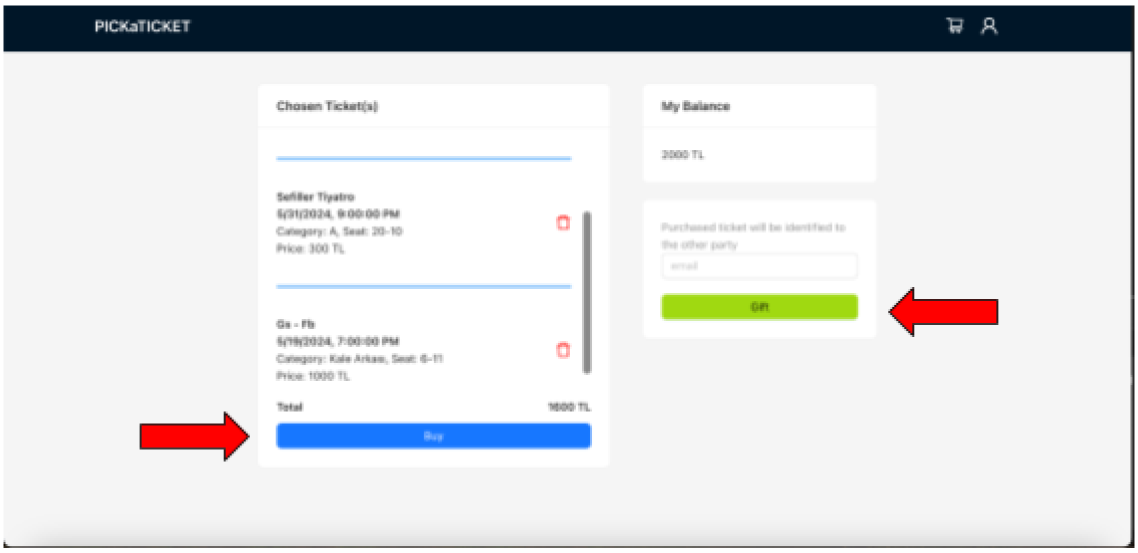


Row: 7, Column: 6

Row: 7, Column: 10

Shopping Cart

When the Ticket Buyer clicks the shopping cart icon in the navbar, they are redirected to the Shopping Cart Page. On this page, the tickets the user has added to their cart and their details are displayed in the chosen ticket(s) section. The Ticket Buyer can remove a ticket from their cart by clicking the trash icon next to each ticket. After adjusting the desired tickets, the user can click the Buy button to complete the purchase. The total amount is deducted from the Ticket Buyer's account and added to the Organizer's account. On the right side of the page, there is the account balance and a gift button. If the user clicks the gift button, they can send the tickets in their cart as a gift to another user by entering their email address.



Sefiller Tiyatro
5/31/2024, 9:00:00 PM
Category: A, Seat: 20-10
Price: 300 TL

Gs - Fb
5/19/2024, 7:00:00 PM
Category: Kale Arkası, Seat: 6-11
Price: 1000 TL

Total 1600 TL

3000 TL

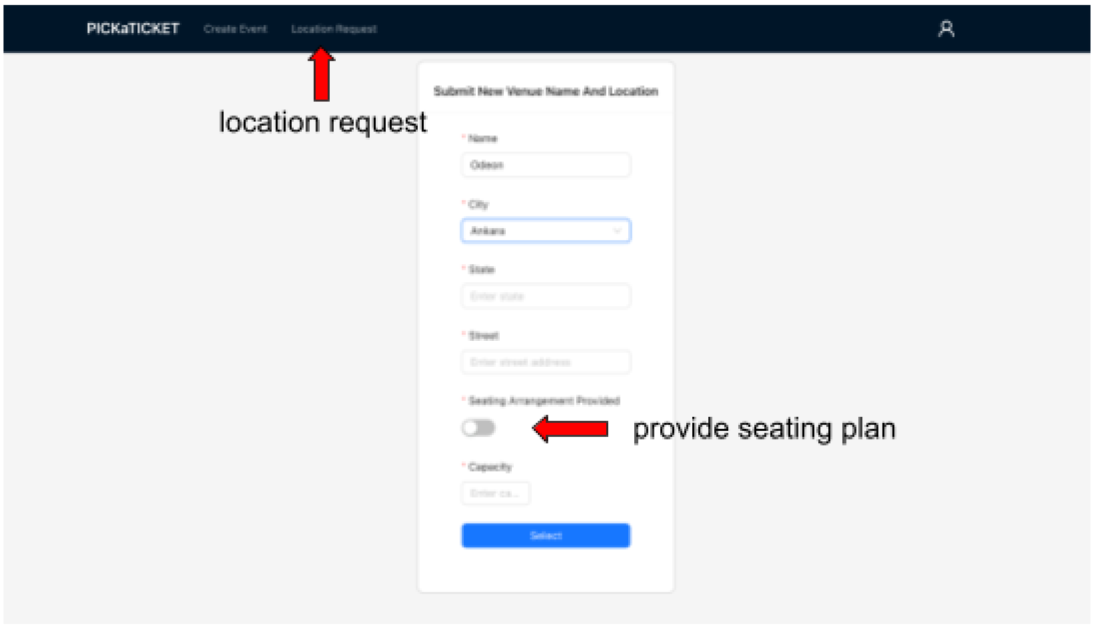
Purchased ticket will be identified to the other party

Publishing an Event

Event Organizers are able to create new event entries to our application. To create a new event a user must be registered as an "event organizer" type user. They can click on a "Create Event" on a navigation bar. To create an event, organizers can choose a location from previously submitted locations. If they want to add a new location to the system, they must first request a new location by providing the necessary details before creating an event.

1. Request Location

To add a new location to the system users must fill the location request form by providing necessary details such as the address, capacity and the seating plan of the location (if provided).



Submit New Venue Name And Location

* Name
Odeon

* City
Ankara

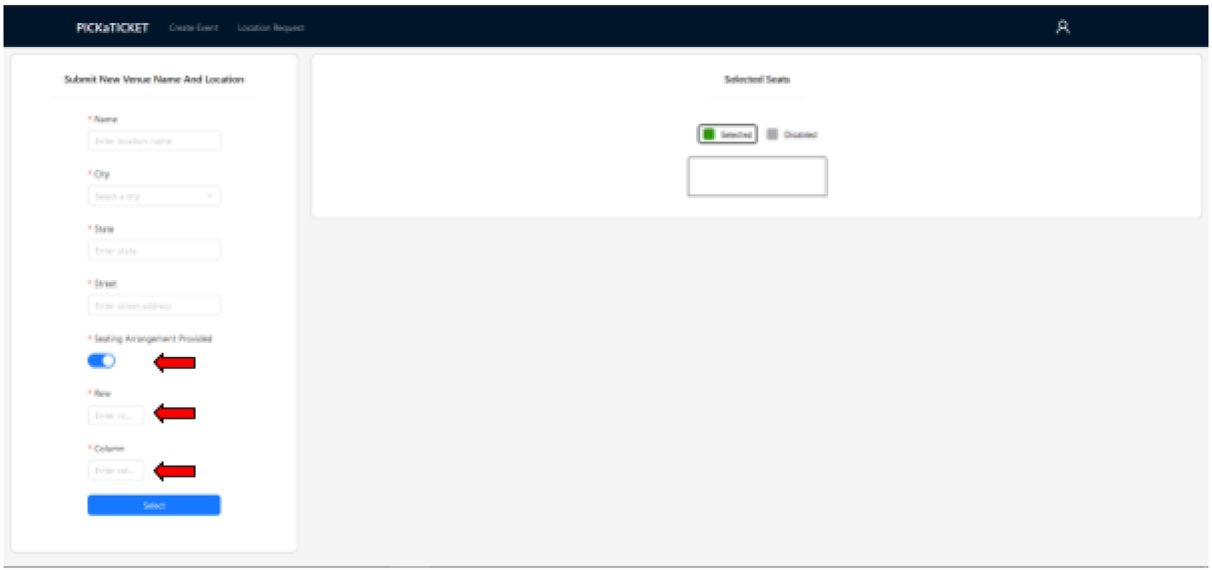
* State
Enter state

* Street
Enter street address

* Seating Arrangement Provided

* Capacity
Enter ca...

To specify a seating plan organizers should choose the option to provide the seating plan. If this option is selected, users should specify the



Submit New Venue Name And Location

* Name
Enter location name

* City
Select a city

* State
Enter state

* Street
Enter street address

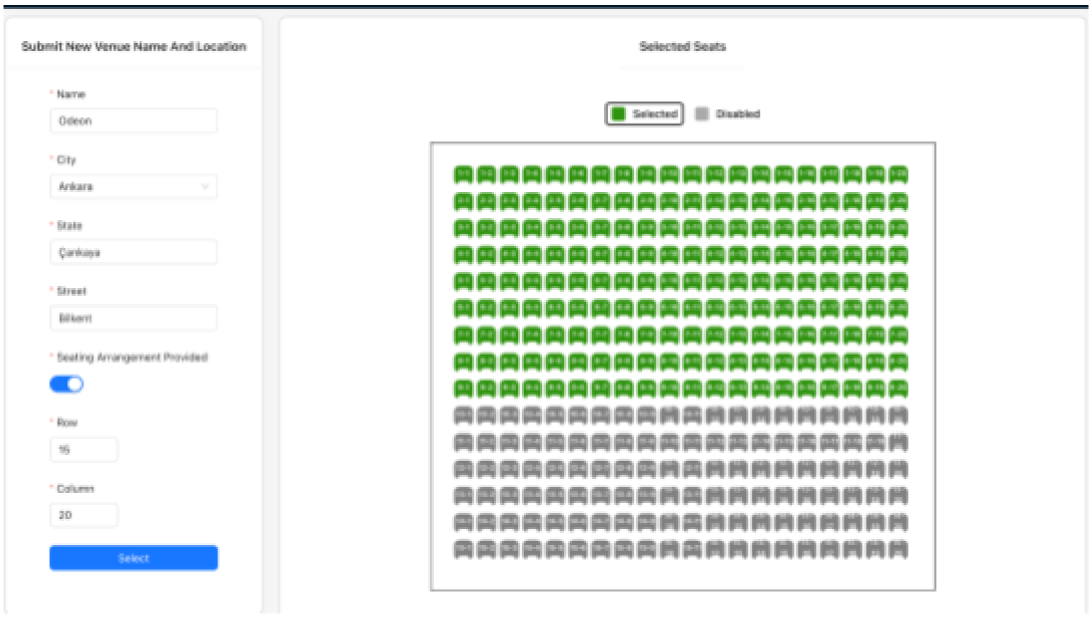
* Seating Arrangement Provided

* Row
Enter ro...

* Column
Enter col...

row and column numbers to construct the seating plan matrix.

However we are aware that not all locations have to have a rectangular shape or regular plan. So we have created a select and drag options to construct the actual shape of the location. You can create regular rectangular seating.



Submit New Venue Name And Location

* Name: Odeon

* City: Ankara

* State: Çankaya

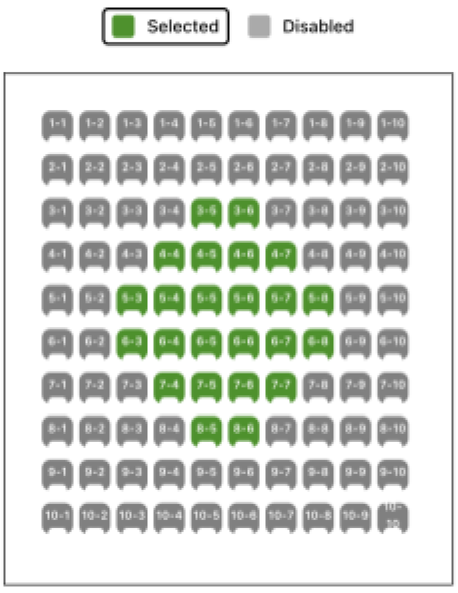
* Street: Bilkent

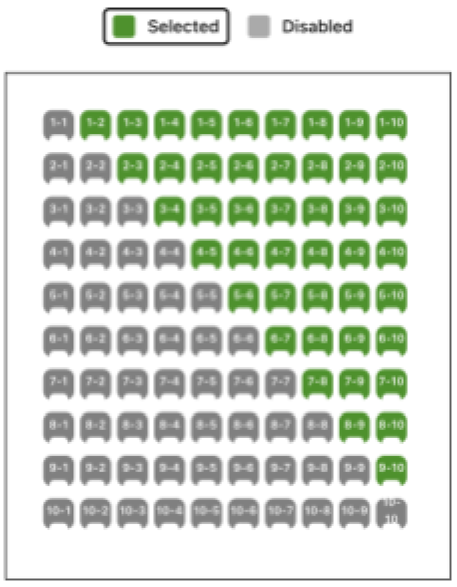
* Seating Arrangement Provided: on

* Row: 15

* Column: 20

Or you can create circular and triangular seating plans.

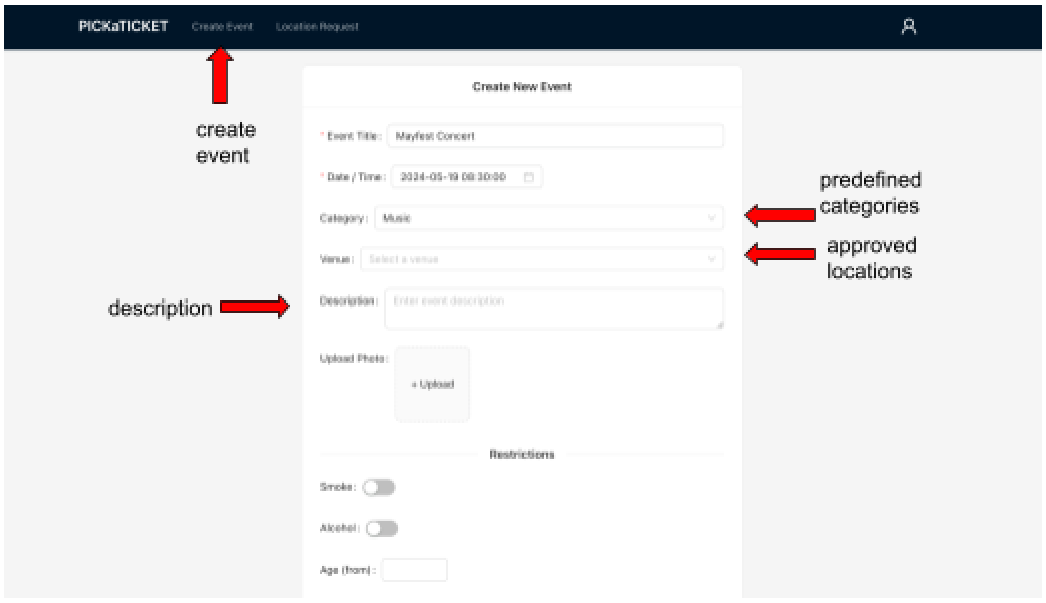




After providing your seating plan for a venue, your request will be sent to the admin to be approved or rejected (see Admin, Location Requests Section). You can see your requests' status from the organizer profile page (see Event Organizer Profile Section).

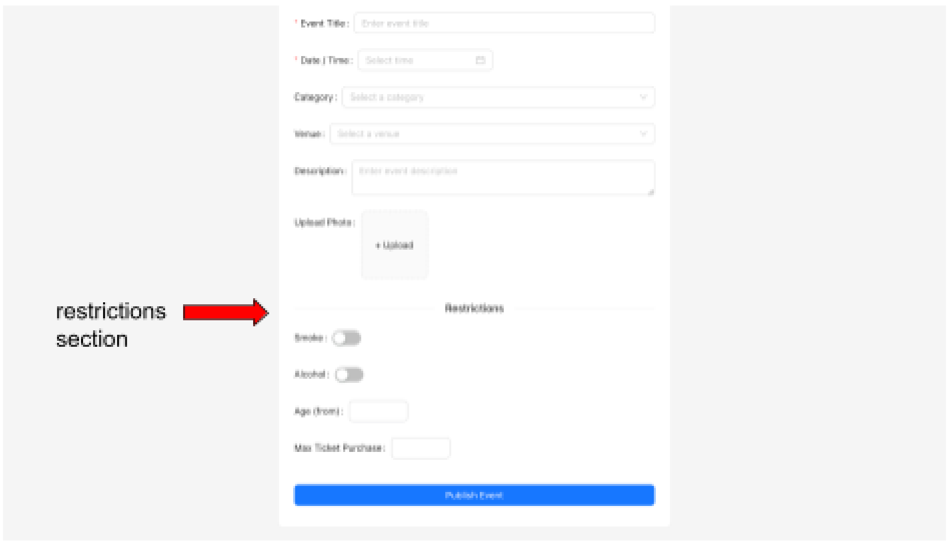
2. Create Event

After successfully getting approval for your new location you can easily create your event. Or you might use previous approved locations from the location list if you wish, without proceeding with step 1. You should specify details of the event such as its category, location, description, photo etc.



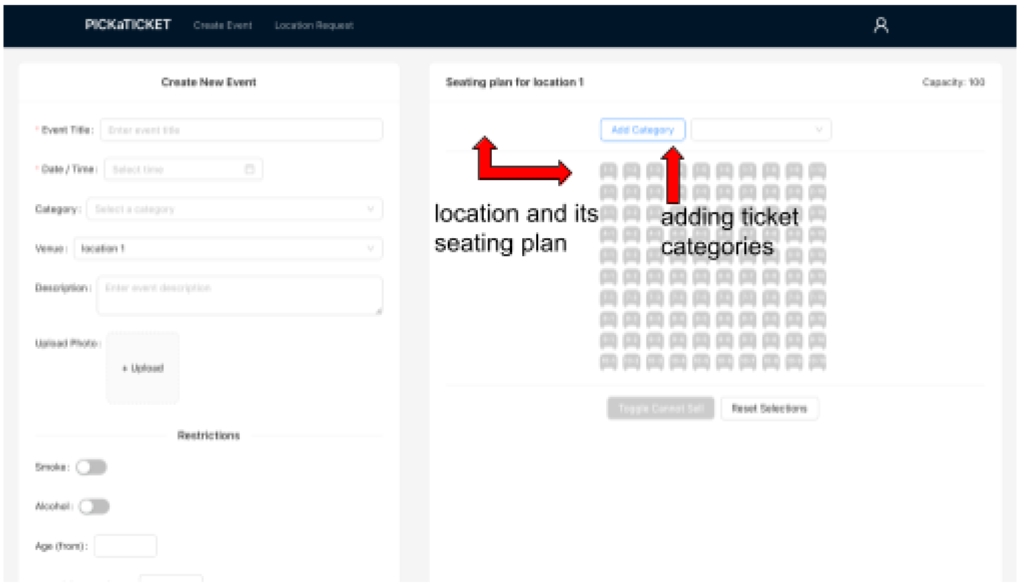
* Event Title: Mayfest Concert
* Date / Time: 2024-05-19 08:30:00
Category: Music
Venue: Select a venue
Description: Enter event description
Upload Photo:
Smoke:
Alcohol:
Age (from):

If there are some restrictions you should also specify them. We have 4 restriction fields which are: smoke, alcohol, age, and the max ticket purchase per transaction.



* Event Title: Enter event title
* Date / Time: Select time
Category: Select a category
Venue: Select a venue
Description: Enter event description
Upload Photo:
Smoke:
Alcohol:
Age (from):
Max Ticket Purchase:

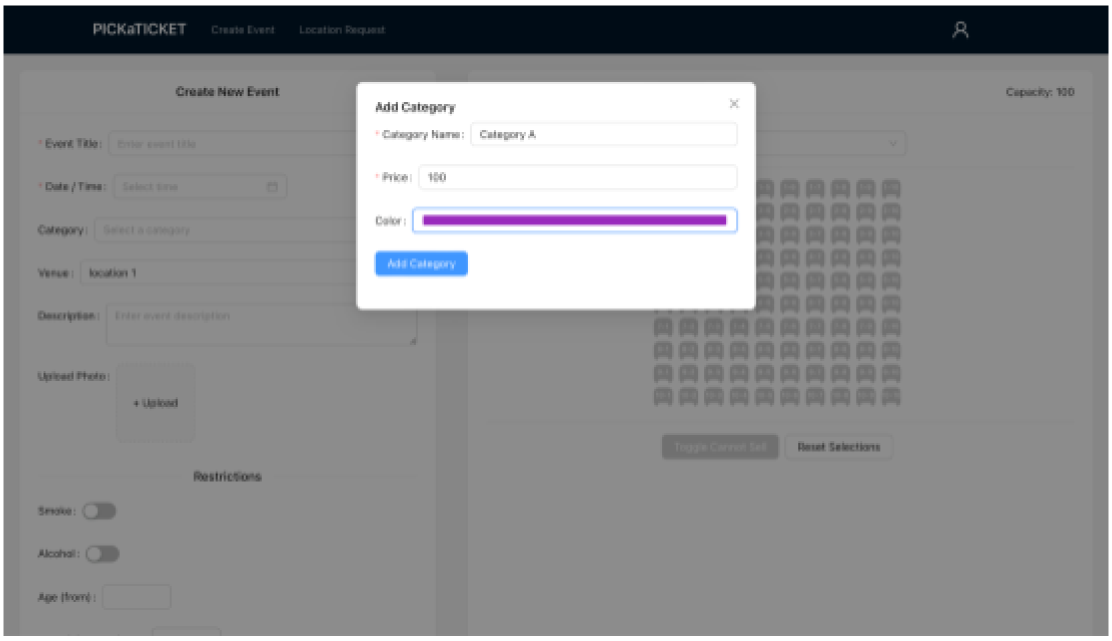
To handle the prices of tickets and seating plan of the event, organizers can set ticket categories with different prices and assign seats to the relevant categories. For simplicity drag and drop is available while choosing the seats.



Venue: location 1

Capacity: 100

Organizers can assign different fees and different colors for each category to represent its category.



Venue: location 1

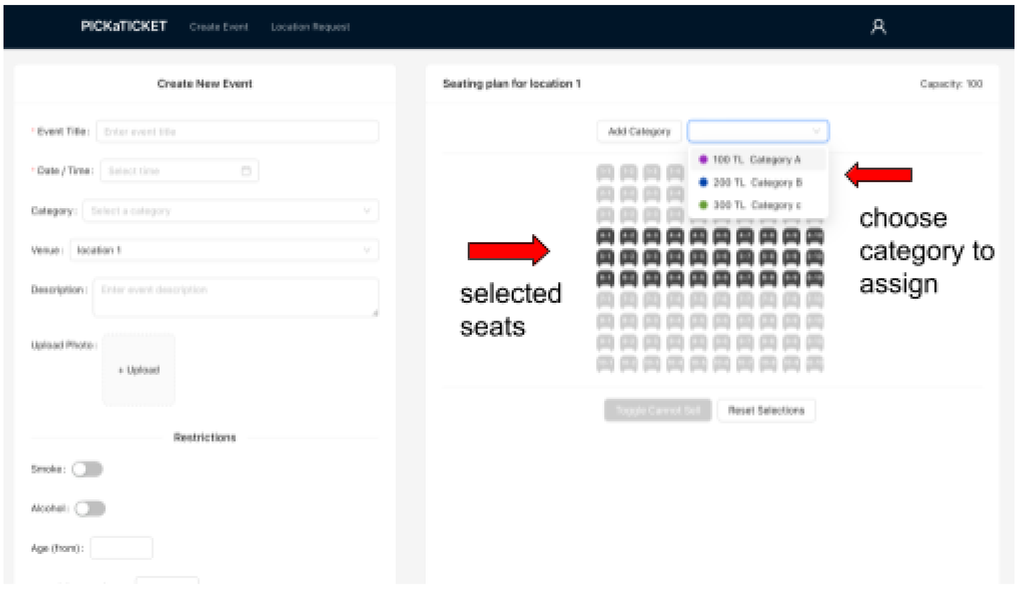
Capacity: 100

Category Name: Category A

Price: 100

Color:

Once adding categories, now they can assign seats for each relevant category. Drag and drop the seats and then choose the category from the category list.



Create New Event

* Event Title: Enter event title

* Date / Time: Select time

Category: Select a category

Venue: location 1

Description: Enter event description

Upload Photo: + Upload

Restrictions

Smoke:

Alcohol:

Age (from):

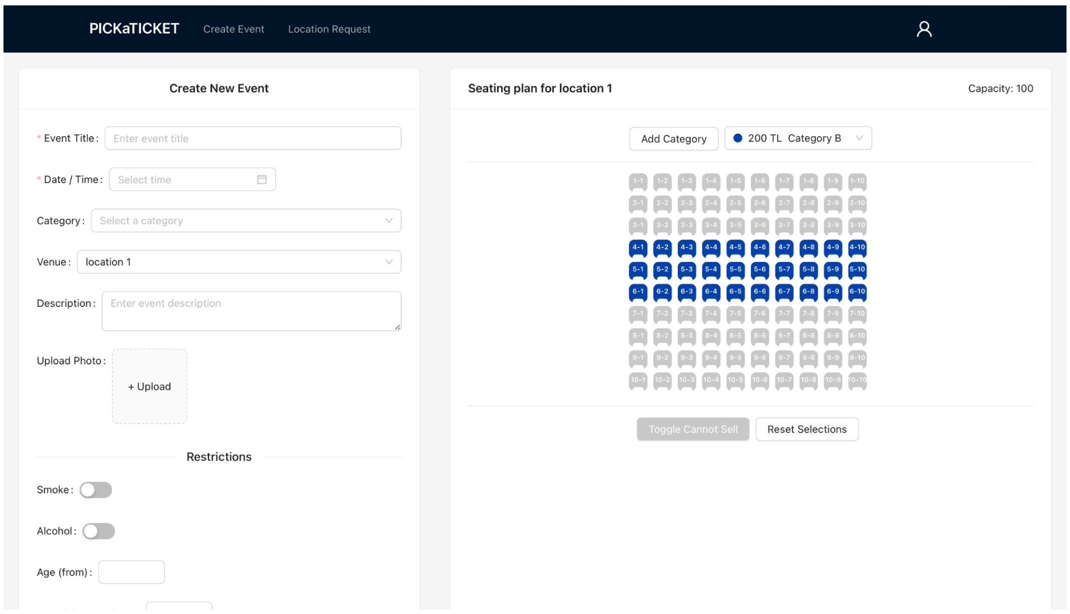
Seating plan for location 1 Capacity: 100

100 TL Category A

200 TL Category B

300 TL Category c

And Voila! Those seats are now assigned for Category B and their prices are set to 200TL.



Create New Event

* Event Title: Enter event title

* Date / Time: Select time

Category: Select a category

Venue: location 1

Description: Enter event description

Upload Photo: + Upload

Restrictions

Smoke:

Alcohol:

Age (from):

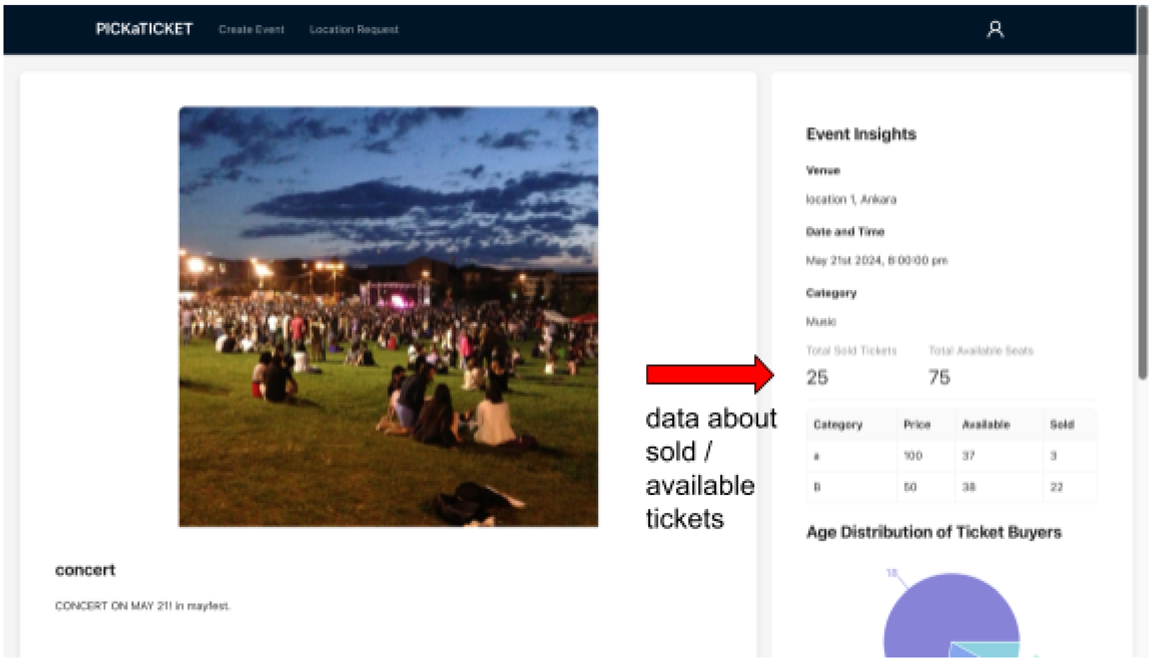
Seating plan for location 1 Capacity: 100

● 200 TL Category B

After providing all necessary details, you can publish your event to the main page of ticket buyers.

Event insights

On the event insights page, organizers are able to see collected and processed data about their events. They are able to see sold tickets for each category, and for total, age distribution of ticket buyers and total revenue for each category of tickets.



concert

CONCERT ON MAY 21! in mayfest.

Event Insights

Venue

location 1, Ankara

Date and Time

May 21st 2024, 8:00:00 pm

Category

Music

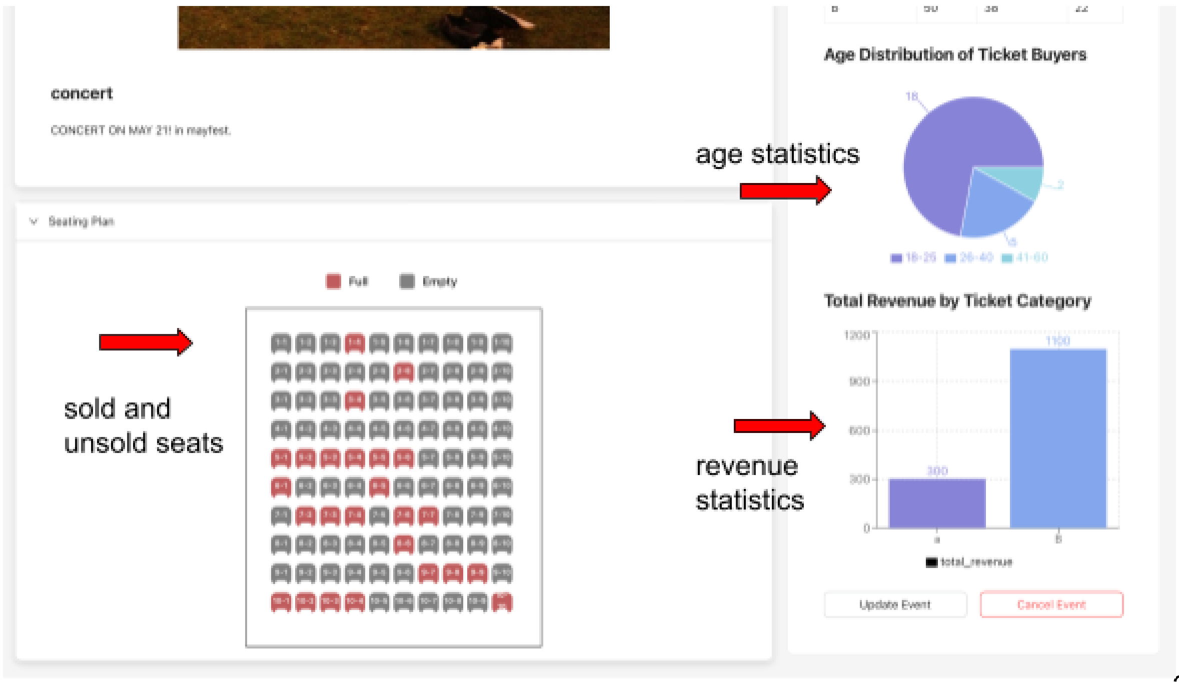
Total Sold Tickets: 25

Total Available Seats: 75

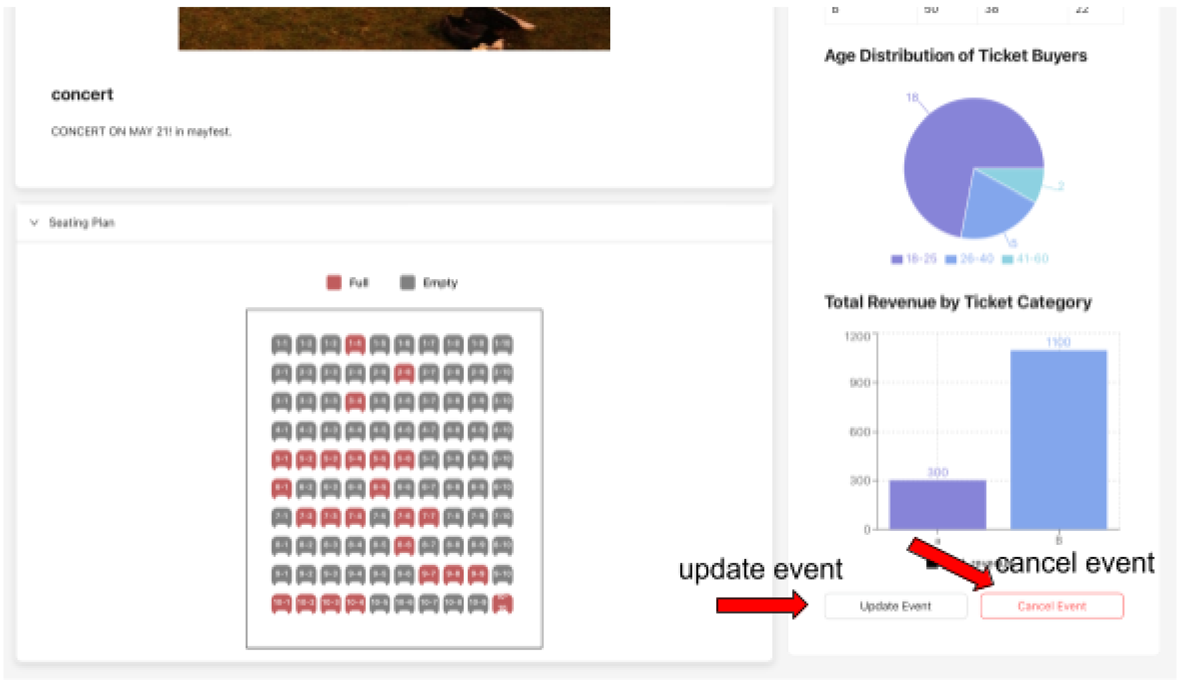
Category	Price	Available	Sold
a	100	37	3
B	50	38	22

Age Distribution of Ticket Buyers

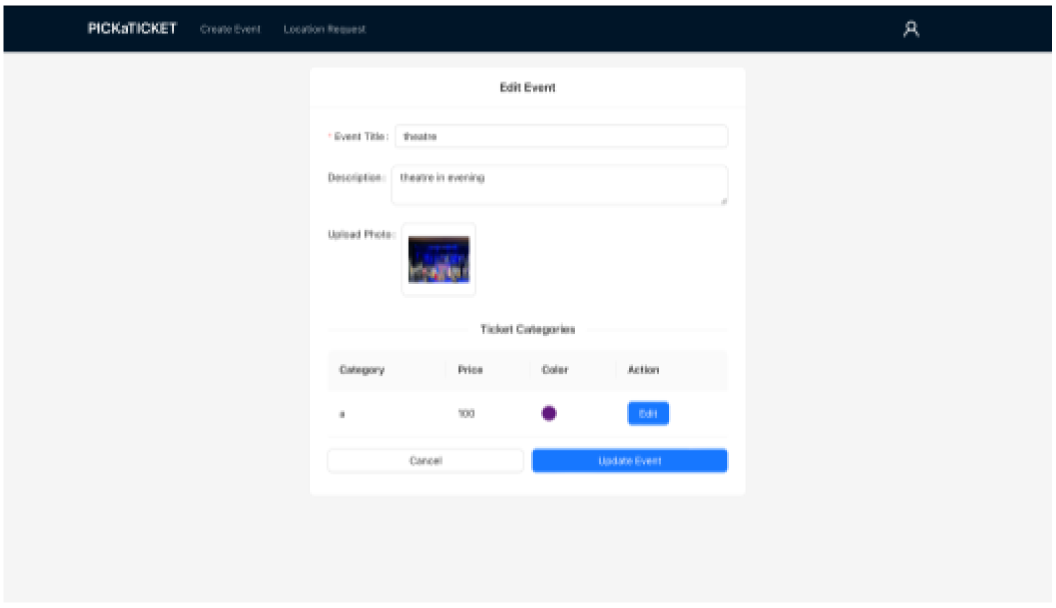
Also they are able to see the seating plan with sold and unsold places, so that they can have information about which seats are preferred so far.



Additionally, organizers can cancel and update their events from the insights page. To cancel an event, an organizer must have sufficient balance to refund all tickets to the buyers. If an event is canceled, money corresponding to the transaction will be refunded to the ticket buyer's balance, and they will be able to see the relevant tickets marked as canceled on their profile pages.



Organizers can also update certain details about the event; however, these changes are restricted to ensure they do not compromise the accuracy of the event's original properties. For example, they can modify ticket prices, such as offering discounts or price adjustments, but they cannot change restrictions (e.g., from 'all ages' to '18+') to maintain consistency for tickets that have already been purchased.



Edit Event

* Event Title: theatre

Description: theatre in evening

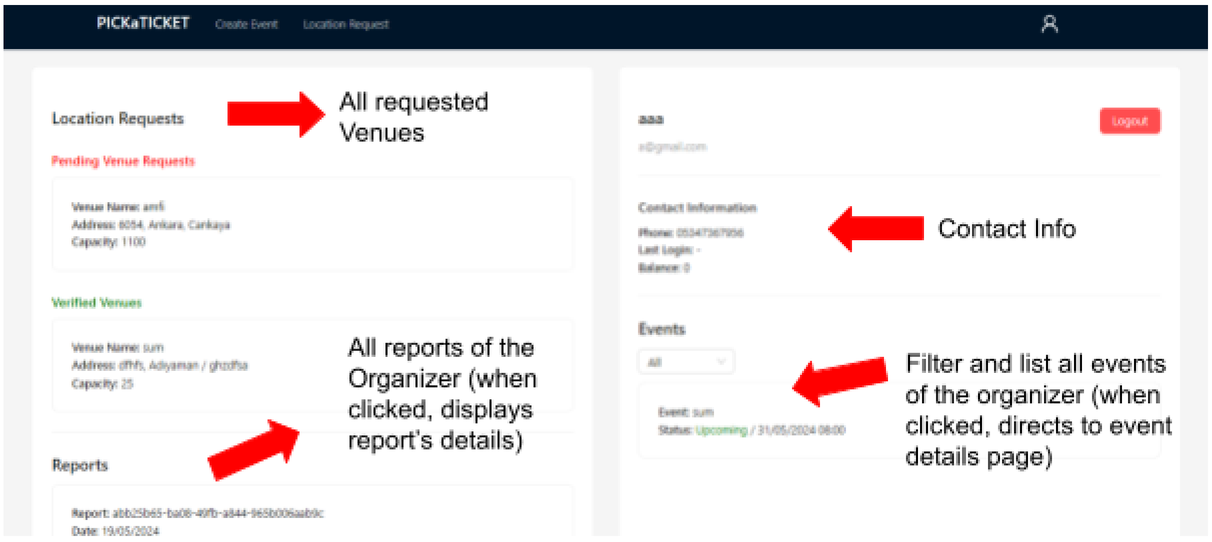
Upload Photo:

Ticket Categories

Category	Price	Color	Action
a	100	●	Edit

Event Organizer Profile

Event organizers are able to keep track of their location requests, their published events, their reports which are created by the admin and their personal information on the organizer profile page. Each report will be displayed in detail if they are clicked. Organizers also are able to filter their events by status such as “Upcoming”, “Passed”, “Canceled”. Clicking to an event will route the organizer to the event insights.


PICKaTICKET
Create Event
Location Request
Logout
All requested Venues
All reports of the Organizer (when clicked, displays report's details)
Contact Info
Filter and list all events of the organizer (when clicked, directs to event details page)

Location Requests

Pending Venue Requests

Venue Name: amfi
Address: 6054, Ankara, Cankaya
Capacity: 1100

Verified Venues

Venue Name: sum
Address: dfhfs, Adiyaman / ghzdfsa
Capacity: 25

Reports

Report: abb25b65-ba08-49fb-a844-965b006aab9c
Date: 19/05/2024

aaa
a@gmail.com

Contact Information
Phone: 05347367956
Last Login: -
Balance: 0

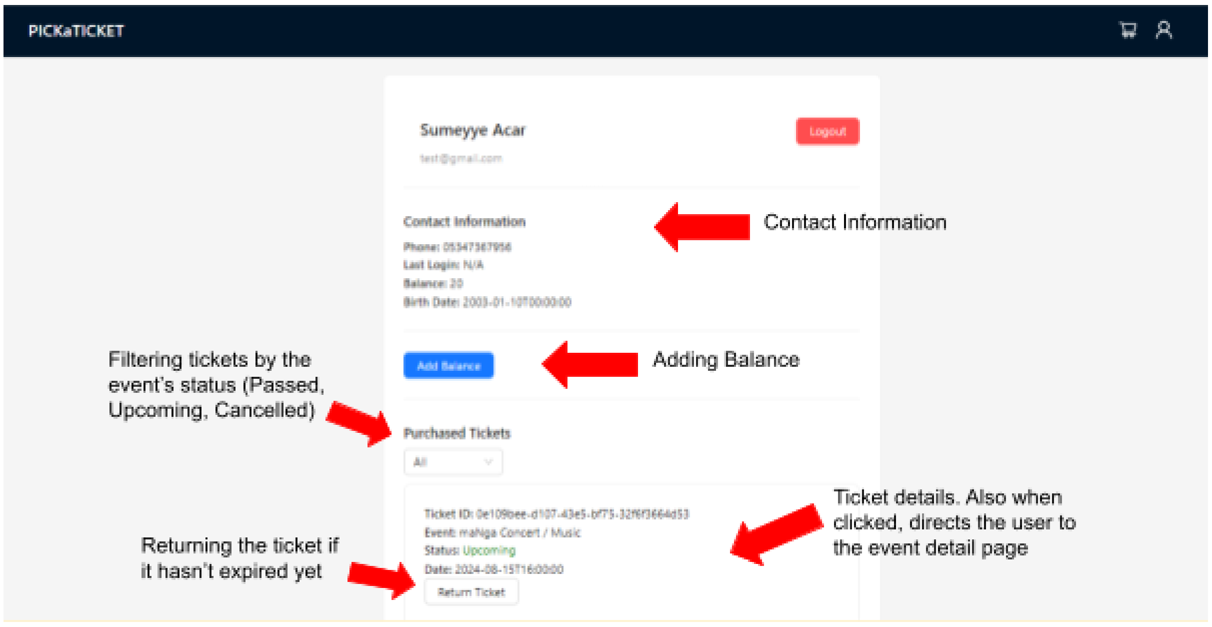
Events

All

Event: sum
Status: Upcoming / 31/05/2024 08:00

Ticket Buyer Profile

Ticket buyers can keep track of the tickets they bought and their personal information on the ticket buyer profile page. The buyers can update their balance form here using the “Add Balance” button after they provide the necessary information. They also are able to filter their tickets by the status of the event such as “Upcoming”, “Passed”, “Canceled”. If the user needs more information about the event for which he/she bought a ticket, all ticket cards are clickable and navigate the user to the events page. Tickets can be refunded from this page using the “Return” button which only appears for tickets that are not expired. After that, the ticket will be available to be sold again and the purchase amount will be refunded to the user.


PICKaTICKET
Sumeyye Acar
test@gmail.com
Logout
Contact Information
Add Balance
Purchased Tickets
All
Return Ticket
Contact Information
Adding Balance
Filtering tickets by the event’s status (Passed, Upcoming, Cancelled)
Ticket details. Also when clicked, directs the user to the event detail page
Returning the ticket if it hasn’t expired yet

Phone: 05347367956
Last Login: N/A
Balance: 20
Birth Date: 2003-01-10T00:00:00

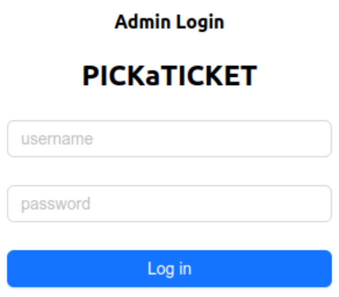
Ticket ID: 0e109bee-d107-43e5-bf75-32f6f3664d53
Event: maNga Concert / Music
Status: Upcoming
Date: 2024-08-15T16:00:00

Admin Panel

Our admin panel provides all necessary details about locations, events, users and their detailed statistics. Also admin can manage all events, locations and users from the panel.

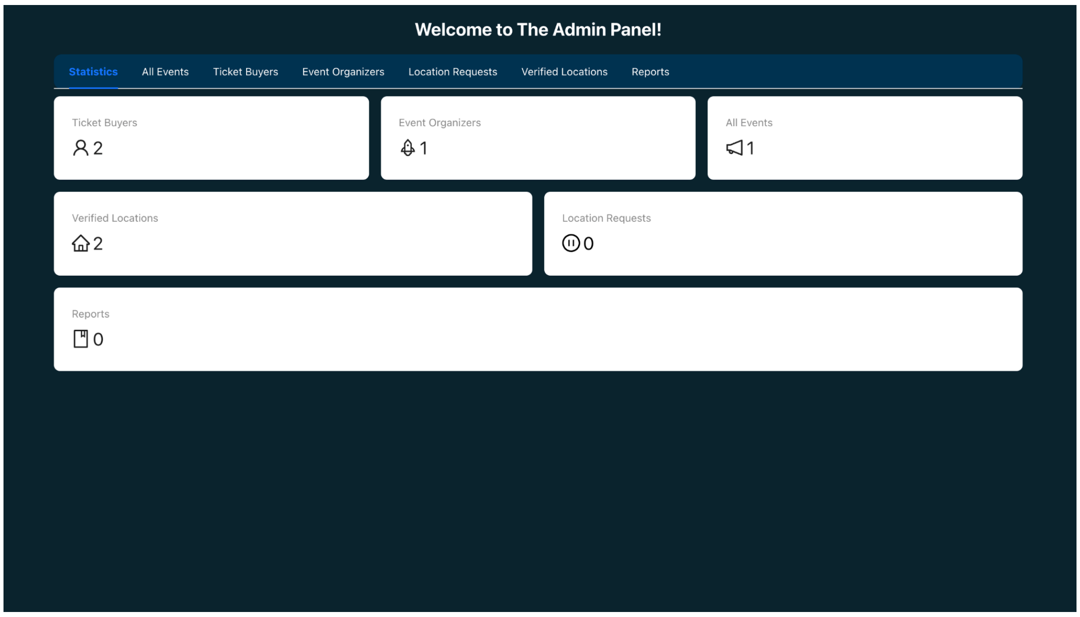
1. Admin Login

This page is the login page for admins. They can enter the system with their username and password.



2. Statistics

This page is the main page of the admin panel to have an overview about the quantities in the application. It can be considered as a brief summary. By clicking the navigation bar or the statistics itself, admin will be directed to the detailed page of the relevant object.

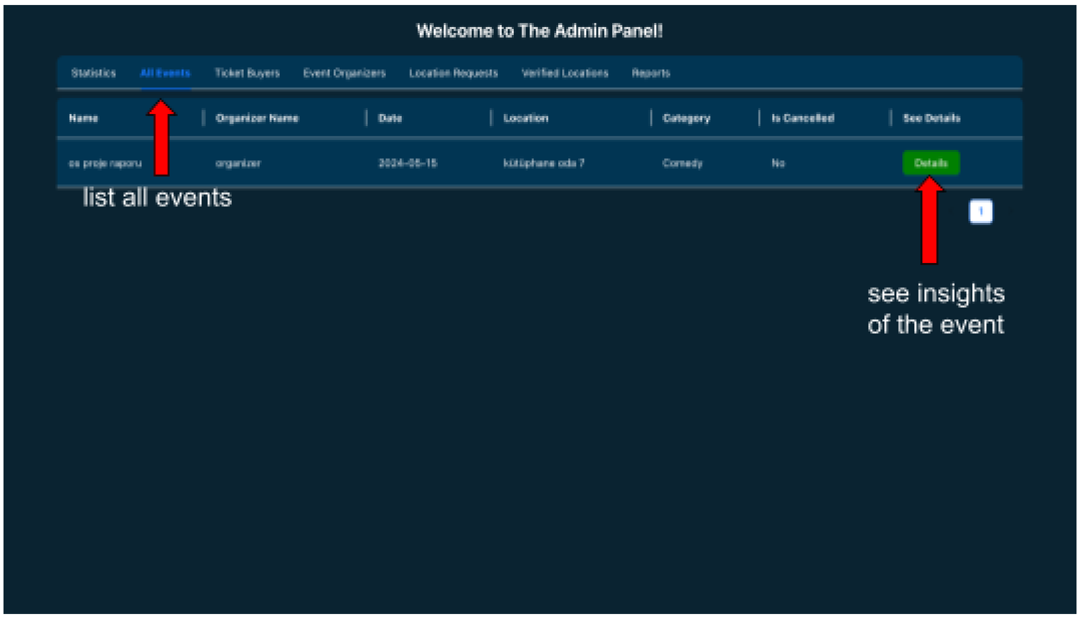


Welcome to The Admin Panel!

Ticket Buyers	2
Event Organizers	1
All Events	1
Verified Locations	2
Location Requests	0
Reports	0

3. All Events

From the all events page, the admin can list all the events created in the application. Also from the details page admin will be able to see the insights of the event, and will be able to both update and cancel the event.

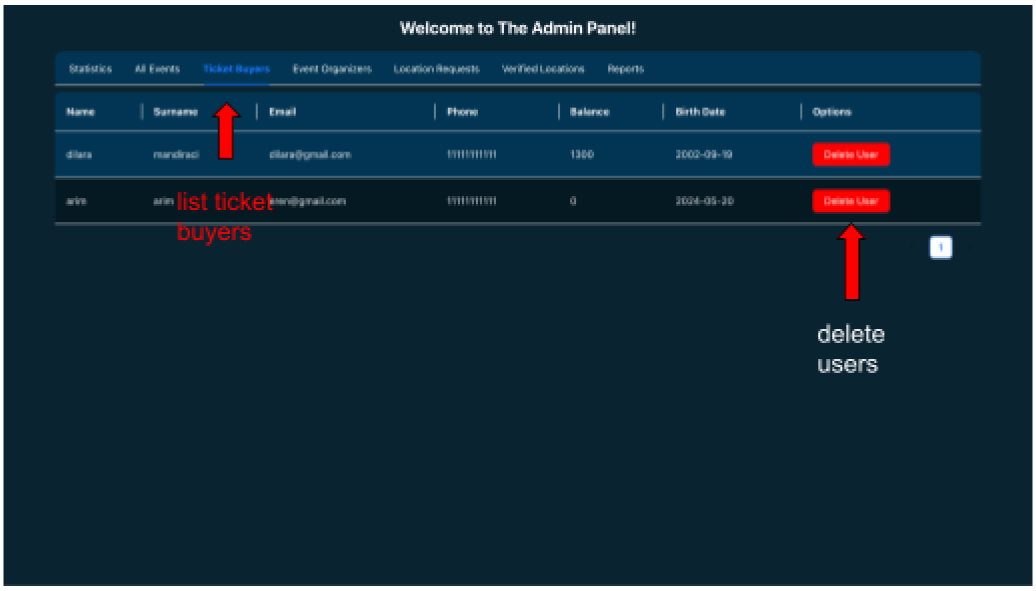


Welcome to The Admin Panel!

Name	Organizer Name	Date	Location	Category	Is Cancelled	See Details
os proje raporu	organizer	2024-05-15	kütüphane oda 7	Comedy	No	Details

4. Ticket Buyers

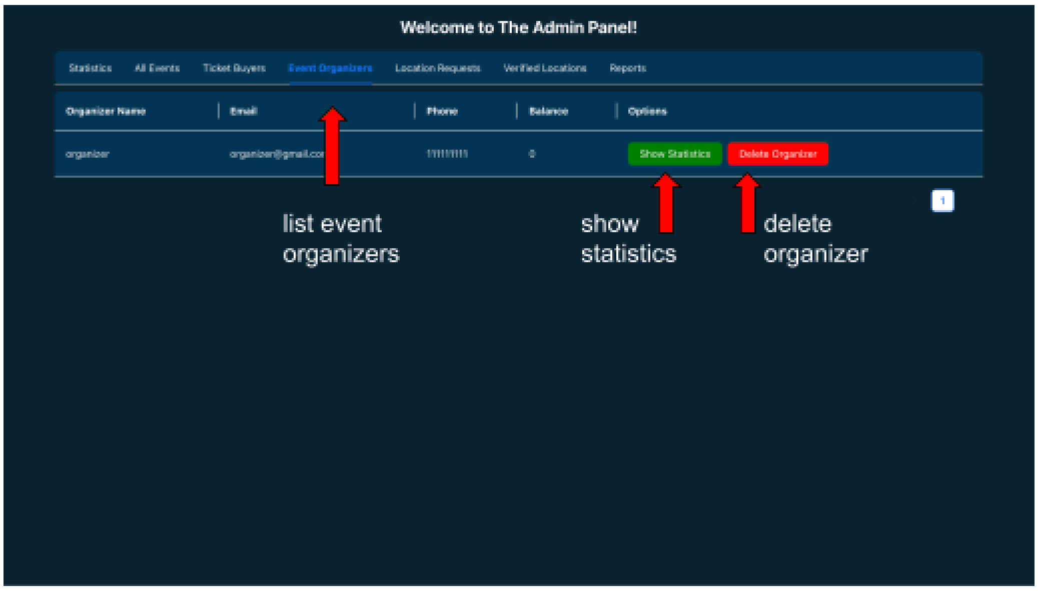
Admin will be able to list all ticket buyers in the system and delete them if necessary.


Welcome to The Admin Panel!

Name	Surname	Email	Phone	Balance	Birth Date	Options
dilara	mandiraci	dilara@gmail.com	11111111111	1300	2002-09-19	Delete User
arin	arin	eren@gmail.com	11111111111	0	2024-05-20	Delete User

5. Event Organizers

Also admin can view all the event organizers, and can delete the organizer user. Different from ticket buyers an admin can also view statistics of the event organizers.

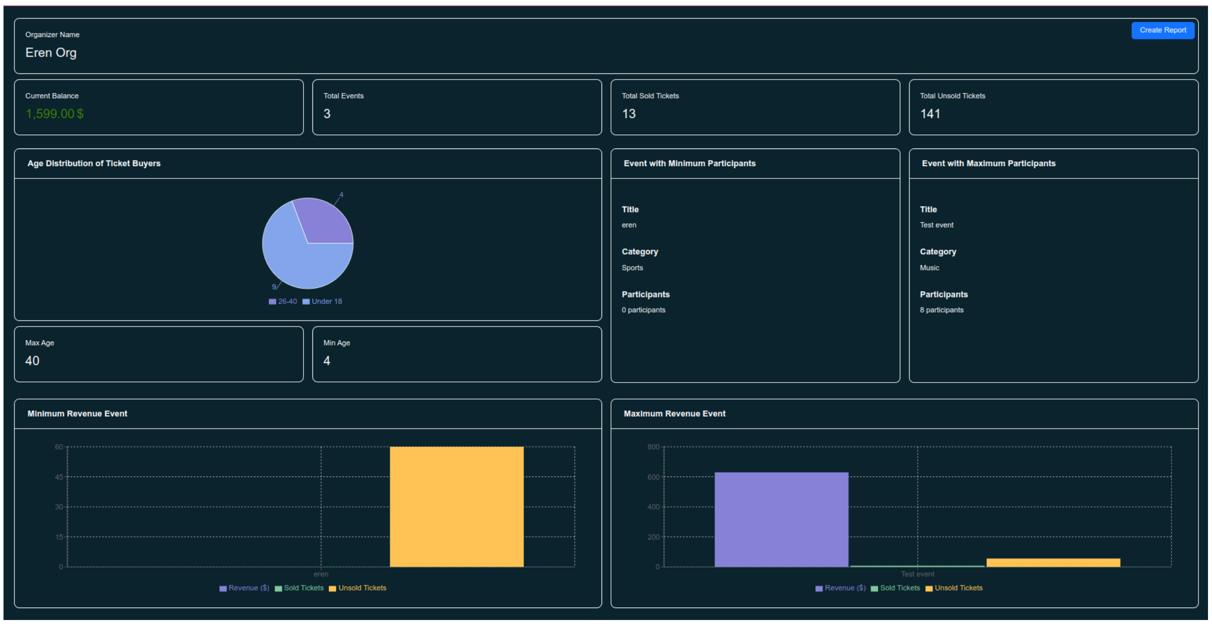

Welcome to The Admin Panel!

Organizer Name	Email	Phone	Balance	Options
organizer	organizer@gmail.co	111111111	0	Show Statistics Delete Organizer

From show statistics page (in section 5) they can see statistics about organizer.

6. Reports

Admin will be able to see the details of the **current** statistics. Also he can create reports about the statistics of an event organizer on that specific day and will send it to the relevant organizers' profile page (see Event Organizer Profile).



Organizer Name
Eren Org

Create Report

Current Balance
1,599.00 $

Total Events
3

Total Sold Tickets
13

Total Unsold Tickets
141

Age Distribution of Ticket Buyers

Max Age
40

Min Age
4

Event with Minimum Participants

Title
eren

Category
Sports

Participants
0 participants

Event with Maximum Participants

Title
Test event

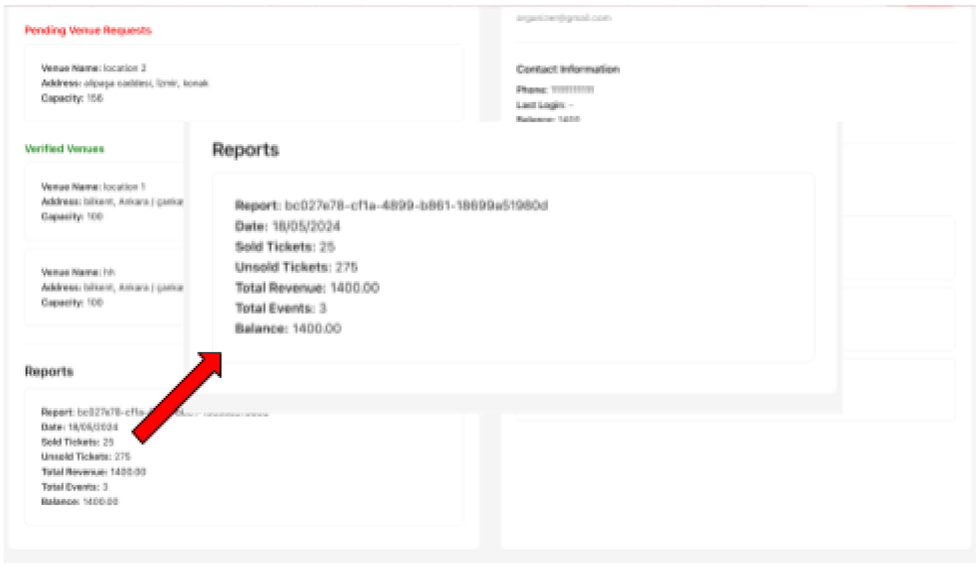
Category
Music

Participants
8 participants

Minimum Revenue Event

Maximum Revenue Event

in event organizer profile page:



Pending Venue Requests

Venue Name: location 2
Address: alpaga caddesi, İzmir, konak
Capacity: 156

Verified Venues

Venue Name: location 1
Address: bilkent, Ankara | çanka

Capacity: 100

Venue Name: hh
Address: bilkent, Ankara | çanka
Capacity: 100

Reports

Report: bc027e78-cf1a-4899-b861-18699a51980d
Date: 18/05/2024
Sold Tickets: 25
Unsold Tickets: 275
Total Revenue: 1400.00
Total Events: 3
Balance: 1400.00

organizer@gmail.com

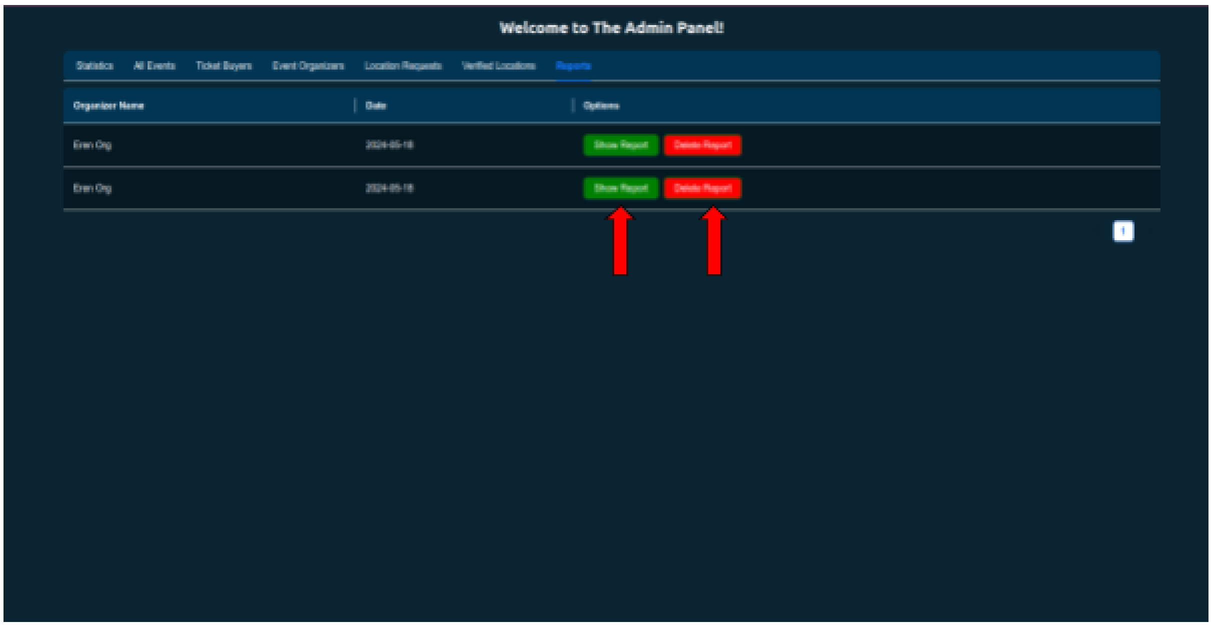
Contact Information

Phone: 1111111111
Last Login: -
Balance: 1400

Reports

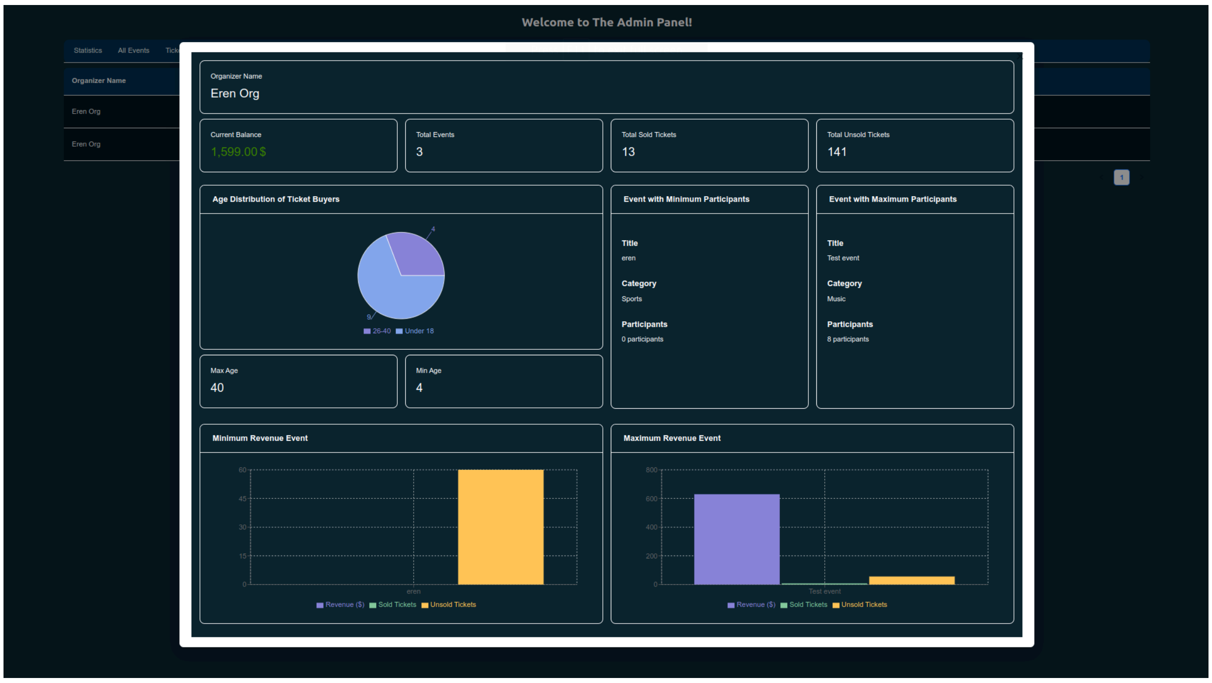
Report: bc027e78-cf1a-4899-b861-18699a51980d
Date: 18/05/2024
Sold Tickets: 25
Unsold Tickets: 275
Total Revenue: 1400.00
Total Events: 3
Balance: 1400.00

Admin can also list the all created reports with their dates and organizers. On this page, previously created reports can be viewed by pressing the show report button or deleted by pressing the delete button.



Organizer Name	Date	Options
Eren Org	2024-05-18	Show Report Delete Report
Eren Org	2024-05-18	Show Report Delete Report

The screen that opens when the show report button is pressed. This page shows statistics created on a specific date.



Organizer Name
Eren Org

Current Balance
1,599.00 $

Total Events
3

Total Sold Tickets
13

Total Unsold Tickets
141

Age Distribution of Ticket Buyers

Event with Minimum Participants
Title
eren
Category
Sports
Participants
0 participants

Event with Maximum Participants
Title
Test event
Category
Music
Participants
8 participants

Max Age
40

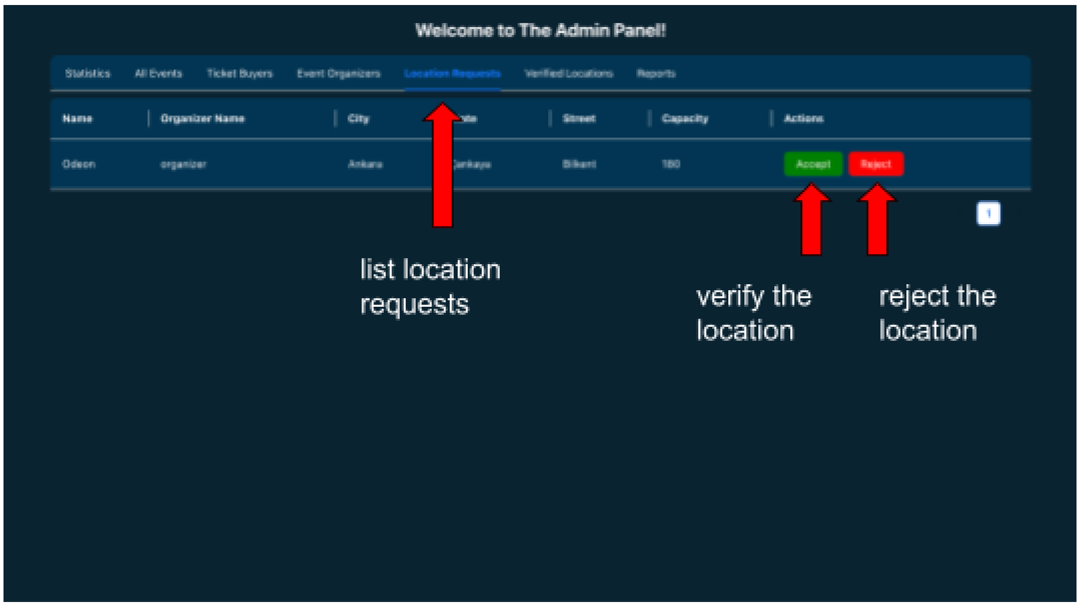
Min Age
4

Minimum Revenue Event

Maximum Revenue Event

7. Location Requests

An admin is responsible for accepting or rejecting location requests. It is important to verify a venue to make it available in our system because we only want approved locations in our system to prevent adding inappropriate locations (like someone's home) etc.

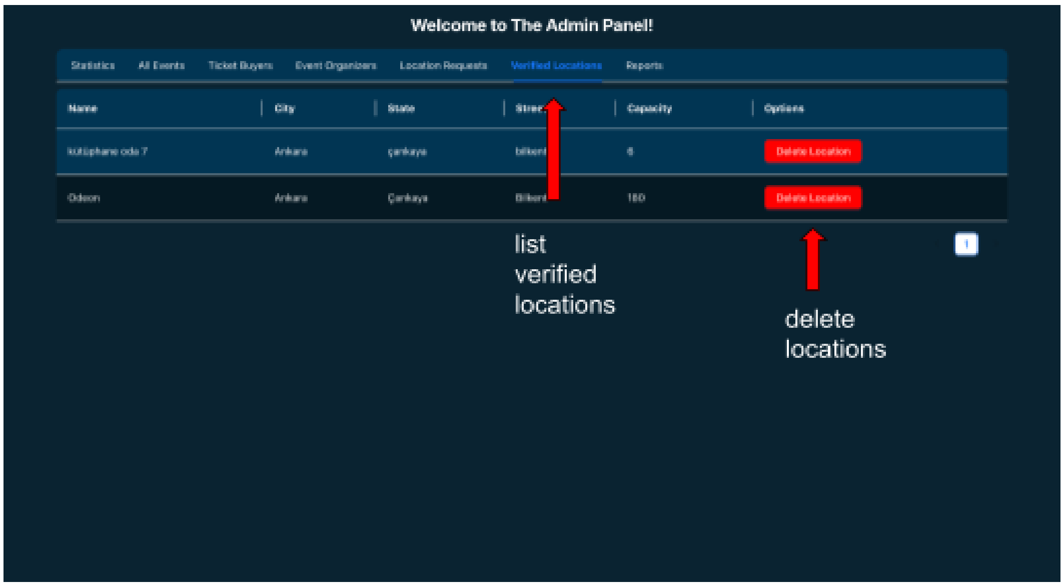


Name	Organizer Name	City	State	Street	Capacity	Actions
Odeon	organizer	Ankara	Çankaya	Bilkent	180	Accept Reject

Only the verified locations are visible to event organizers, and they are allowed to create events at those venues.

8. Verified Locations

Admin can list all the approved locations and is able to delete the approved locations from the system.



Name	City	State	Street	Capacity	Options
kütüphane oda 7	Ankara	çankaya	bilkent	6	Delete Location
Odeon	Ankara	Çankaya	Bilkent	180	Delete Location