



FLEXIGATHER

Event Management System

M4: Final Presentation and Results

Projeto em Informática 2023/24



universidade
de aveiro

Roberto Castro - 107133
Rafaela Abrunhosa - 107658
Marta Inácio - 107826
Tiago Gomes - 108307
Sara Almeida - 108796

Our team



Marta Inácio
Software Architect



Sara Almeida
Team Manager



Rafaela Abrunhosa
DevOps Master



Tiago Gomes
Product Owner



Roberto Castro
DBA / DevOps

Guiding Teacher



Joaquim Sousa Pinto

Table of contents

01 Context and State of The Art

02 Problem

03 Actors and Requirements

04 Data Model and Architecture

05 Implementations

06 Testing and Results

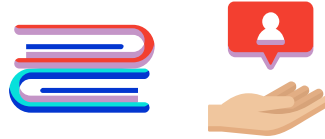
07 Deployment and Future Work

08 Demo

Context



The event management industry faces constant challenges.







Develop an integrated management system for the events:
Participants info, medical info, entry and exit system, etc ...



Able to simultaneously deal with large audiences in different activities of the same event.

State of The Art

				
Access Management	✓	✓	✓	✓
Balance Management	✓	✗	✗	✓
Payment Methods (QR Code)	✓	✗	✗	✓
Roles Management	✓	✓	✓	✓
Purchase Products via QR Codes	✓	✗	✓	✓
Account Loading for Cashless Transactions	✓	✗	✗	✓
Medical Attention Management	✗	✗	✗	✓

Our domain based system



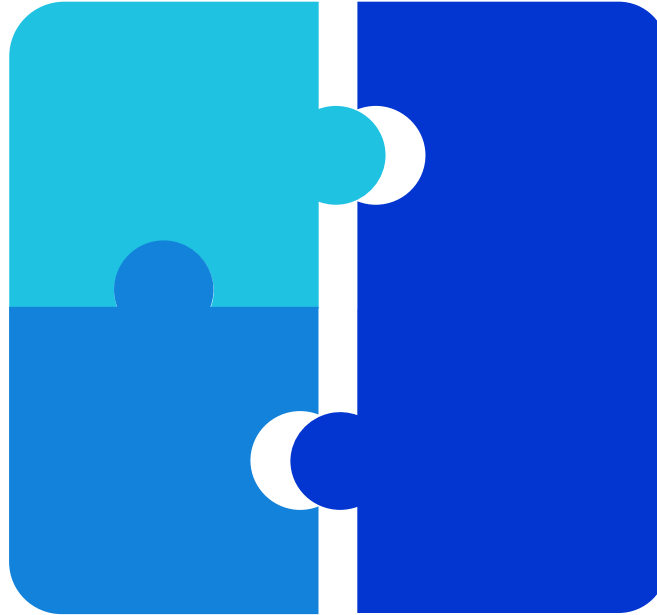
Medical Center

With pre-filled medical files to better attend **emergency** situations.



Events

Access and Kiosks logistics common to all this kind of events.



Theme

The theme of the event can be changed to **meet** the client **expectations** (i.e Scouts Event)

03

Actors and Requirements

Participants

- Access personal data
- Enter and leave the premises
- Add money to their accounts



Medical Staff

- Access medical information
- Create treatment files
- Access treatment files history



Administration

- Allow registration of participants
- Update product stock
- See the event participants flow



- Registration of sales
- Allow money to be load into participants accounts



Sales Staff

- Record entries of the participants
- Record exits of the participants



Entrance Staff

03

Non-functional Requirements

Functionality

The system should be able to read QRcodes/NFCcodes.

The system should allow the use of more than one browser.

Performance

The system should handle large volumes of audiences (data capacity).

The system should warn the responsible chiefs in less than 3 seconds whenever a child on their charge entries in the medical center (response time).

Security

Only medical and administration staff can have access to the participants health information.

Only sales staff can make a sale or exchange.

Availability

All the interfaces (except the kiosk) should be available 24/7, this including during the night.

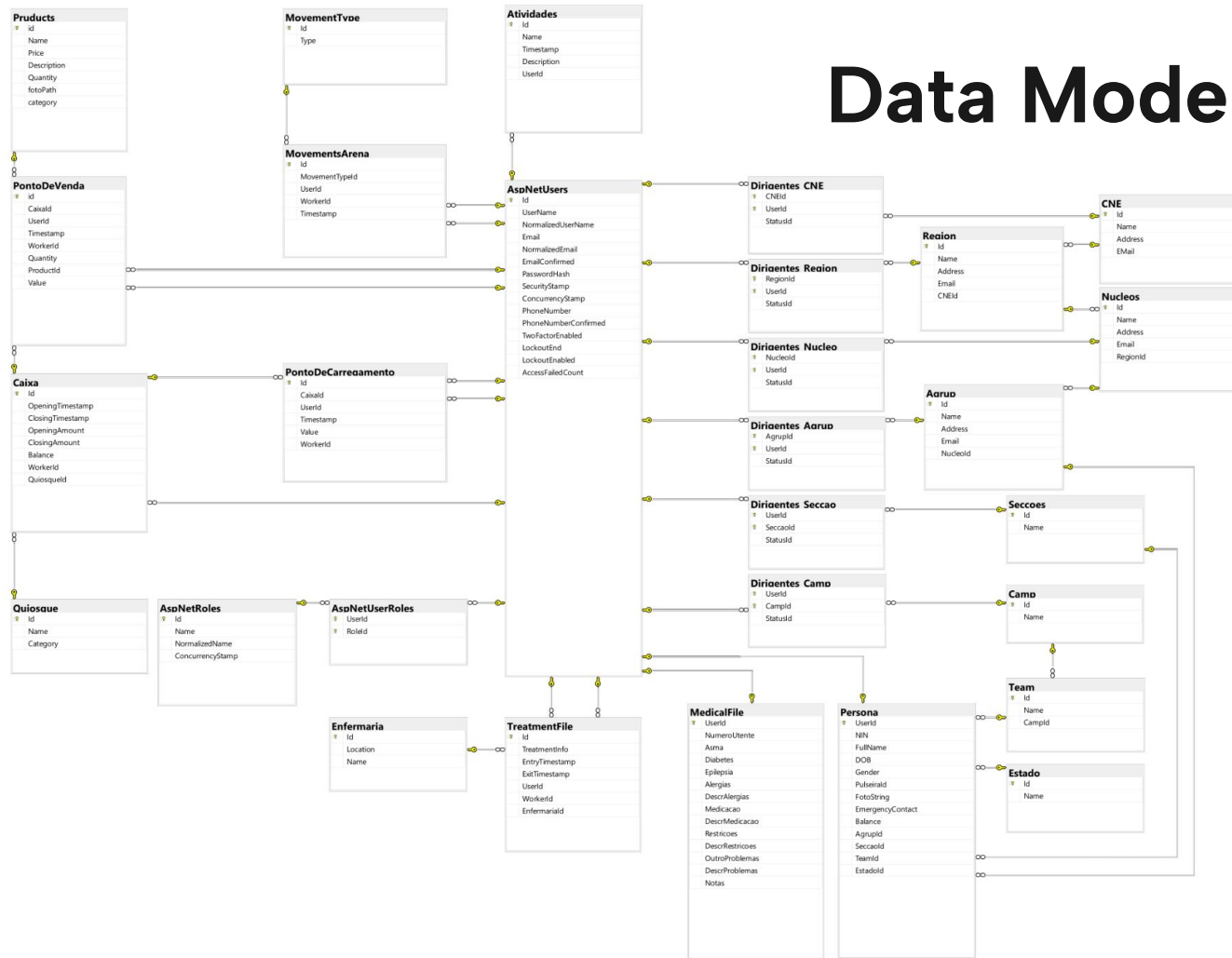
If the system crashes, the data should not be lost.

03

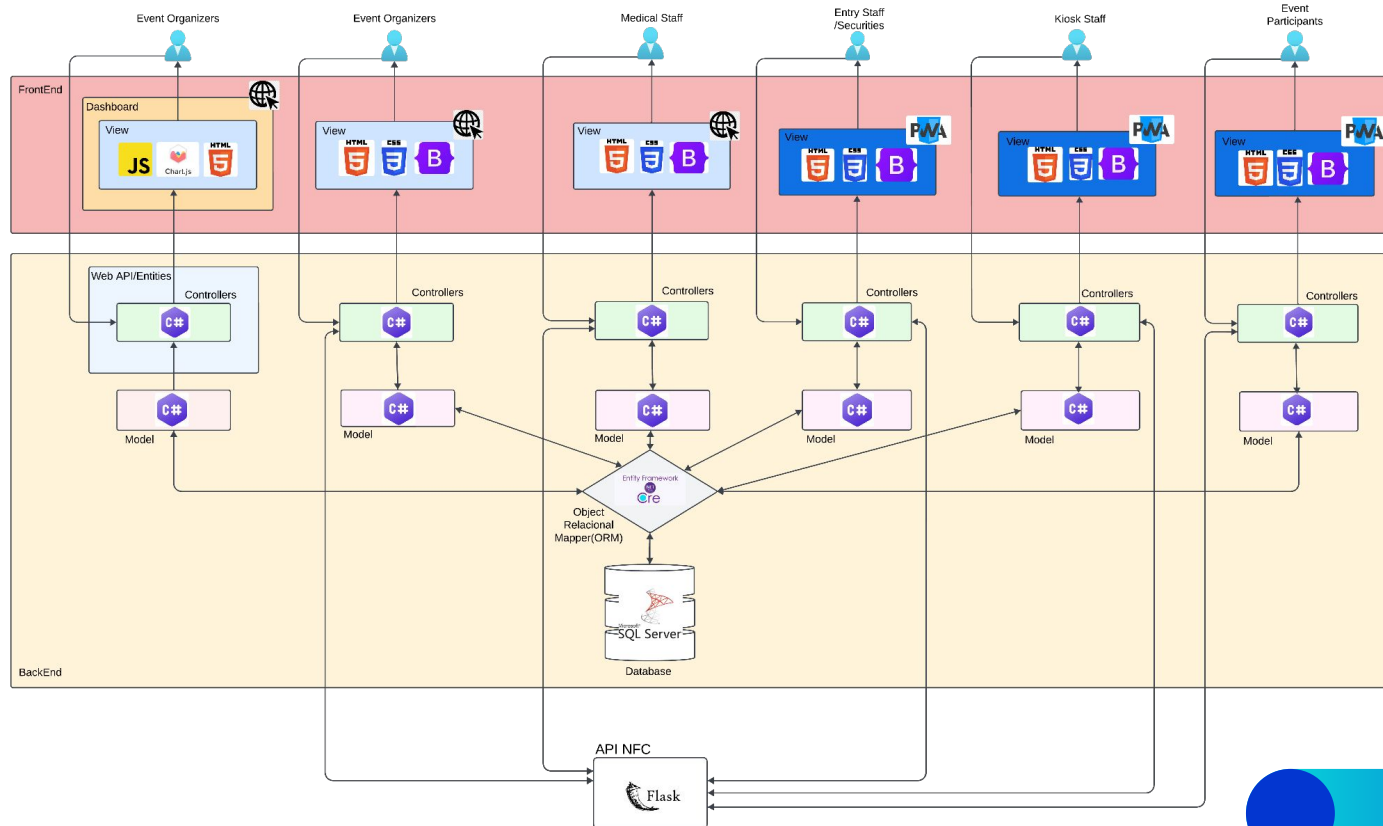
User's Identification

- NFC bracelets
- QR Codes





System Architecture



Implementations

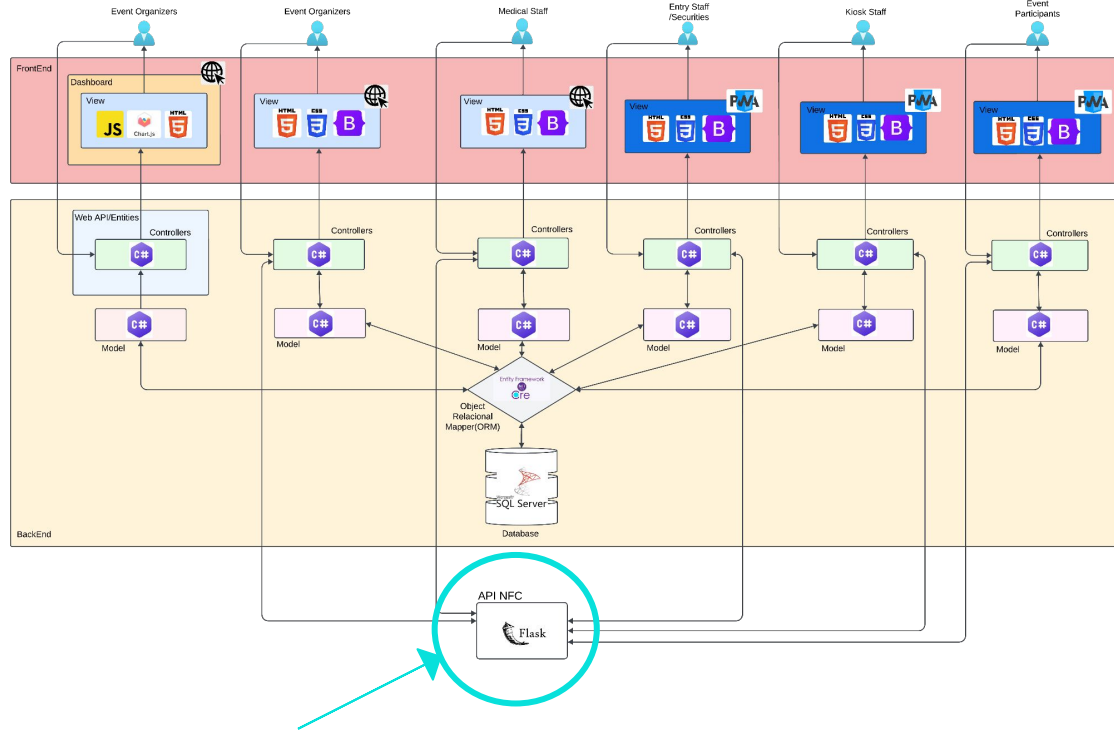
05



NFC Integration

API development

GET: /api/nfc/read



Implementations



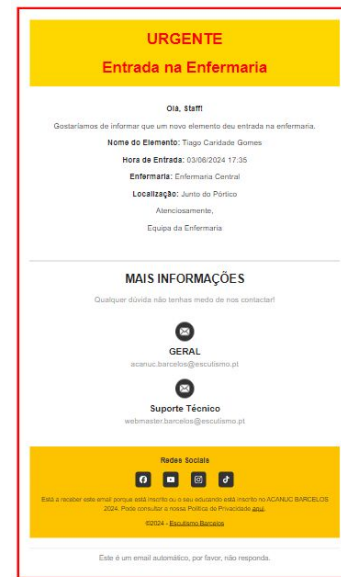
Participants photo identification

The images identifying the participants will be stored in the wwwroot folder.



Notifications

The notifications are meant to alert the **responsible** of the children for their entry and movement in the medical center.

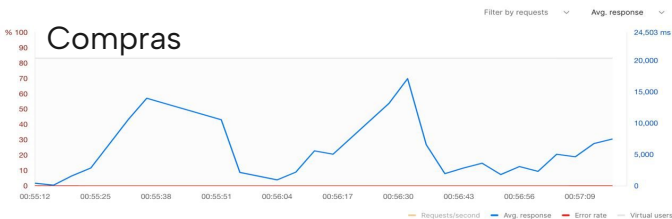


Testing and Results

Total requests sent ② 202 Requests/second ① 1.54 Avg. response time ① 3,202 ms Error rate ① 0.00 %



Total requests sent ② 224 Requests/second ① 1.75 Avg. response time ① 4,324 ms Error rate ① 0.00 %



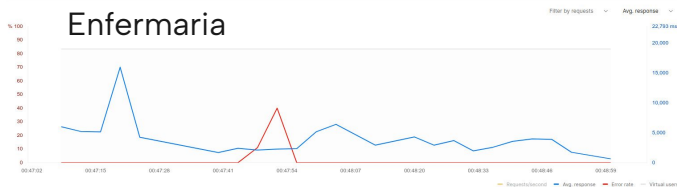
Total requests sent ② 185 Requests/second ① 1.41 Avg. response time ① 3,590 ms Error rate ① 5.41 %



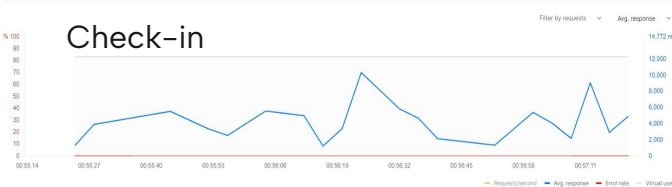
Total requests sent ② 180 Requests/second ① 1.38 Avg. response time ① 3,650 ms Error rate ① 5.56 %



Total requests sent ② 93 Requests/second ① 0.73 Avg. response time ① 3,306 ms Error rate ① 5.39 %



Total requests sent ② 190 Requests/second ① 1.45 Avg. response time ① 3,921 ms Error rate ① 0.00 %



Publicar

Onde a publicação será feita hoje?

Destino



Azure

Host your application to the Microsoft cloud



Registro de Contêiner do Docker

Publicar o aplicativo em qualquer Registro de Contêiner compatível que funcione com imagens do Docker



Pasta

Publicar o aplicativo em uma pasta local ou em um compartilhamento de arquivo



Servidor FTP/FTPS

Publicar o aplicativo em um servidor FTP/FTPS



Servidor Web (IIS)

Publicar o aplicativo no IIS usando Implantação da Web ou Pacote de Implantação da Web



Importar Perfil

Importar as configurações de publicação para implantar o aplicativo

Voltar

Próximo

Concluir

Cancelar

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=192.168.160.27;Database=FlexiGather;User Id=FlexiGather;Password=..."
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}

AppSettings Release
```

```
{
  "DetailedErrors": true,
  "ConnectionStrings": {
    "DefaultConnection": "Server=localhost;IntegratedSecurity=true;MultipleActiveResultSets=true;Trust..."
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  }
}

AppSettings Development
```


Future Work



Asynchronous messaging

Like Kafka to help with traffic and scalability.



Notifications

Implement proper system notification.



Payment methods

Add the possibility of charging bracelets without physical money.



08

Demo



Thanks!

