

Straka Add-Ons



Infomatik



PROJEKTIDEE

Die bereits bestehende App "Straka" ist eine App, mit der Teams (zB. Fußballmannschaften) ihre internen Strafen (Multe) einfach in einer App verwalten können. Alle Strafen werden übersichtlich erfasst und sind für alle nachvollziehbar. Wir wollen nun diese App mit einem Kalendermodul und einer PayPal-Schnittstelle erweitern.

Kalendermodul

Wir wollen Teams eine zentrale und übersichtliche Möglichkeit zur Organisation gemeinsamer Termine direkt innerhalb der App anbieten. Anstatt externe Kalender oder Chatnachrichten zu nutzen, steht dem Team ein geteilter Kalender zur Verfügung, der speziell auf die Bedürfnisse von Mannschaften zugeschnitten ist.

PayPal-Schnittstelle

Um Mannschaften eine nahtlose, bargeldlose und transparente Möglichkeit zur Begleichung von teaminternen Schulden direkt aus der App heraus zu bieten, wollen wir eine PayPal-Schnittstelle integrieren.



FUNKTIONSWEISE

Kalendermodul:

- Der Teammanager wählt im Kalender einen Tag aus, gibt Datum, Uhrzeit, Eventart, Ort und Notizen ein und klickt auf „Erstellen“.
- Das Event wird im System gespeichert und dem gemeinsamen Teamkalender hinzugefügt.
- Tage mit Events werden für alle Spieler farblich hervorgehoben.
- Spieler können den Kalender ansehen, aber keine Änderungen vornehmen.
- Beim Klick auf einen markierten Tag erscheint ein Pop-up mit allen Events und Details des Tages.

PayPal-Schnittstelle:

- Der Spieler sieht im Finanzbereich seine offenen Strafen bzw. sein aktuelles Guthaben.
- Mit einem Klick auf „Mit PayPal bezahlen“ wird die Zahlung erstellt und der Spieler zu PayPal weitergeleitet.
- Das Geld geht direkt an das PayPal-Konto des zuständigen Teammanagers.
- Die App prüft währenddessen automatisch den Zahlungsstatus.
- Nach erfolgreicher Zahlung wird die Strafen als bezahlt markiert.
- Der Manager sieht sofort, dass keine offenen Beträge mehr bestehen.



DATEN UND FAKTEN

Architektur:

- Client-Server-Architektur
- Trennung von Frontend, Backend und Datenbank

Frontend:

- Entwickelt mit Flutter
- Kommunikation mit dem Backend über HTTP

Backend:

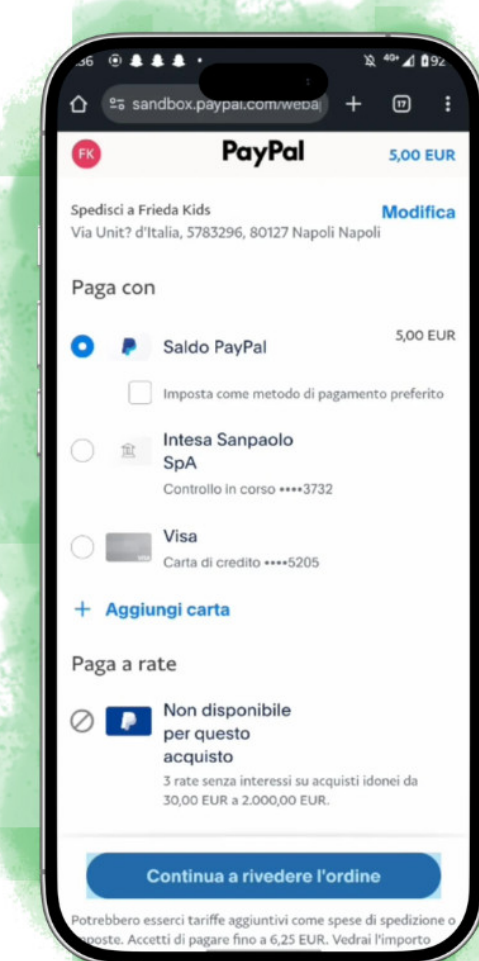
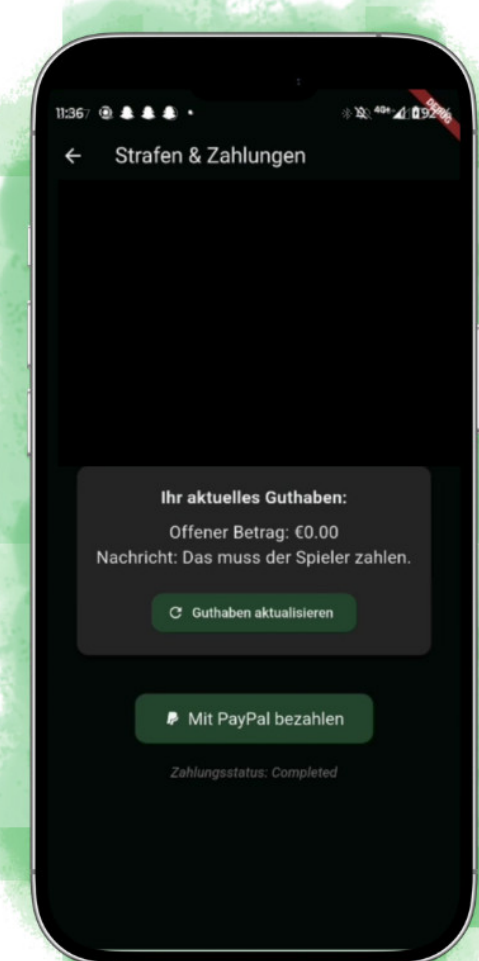
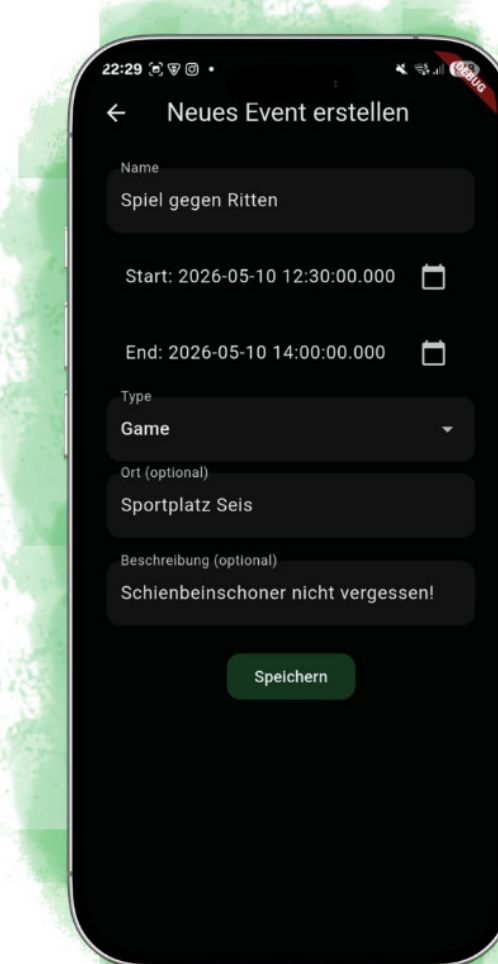
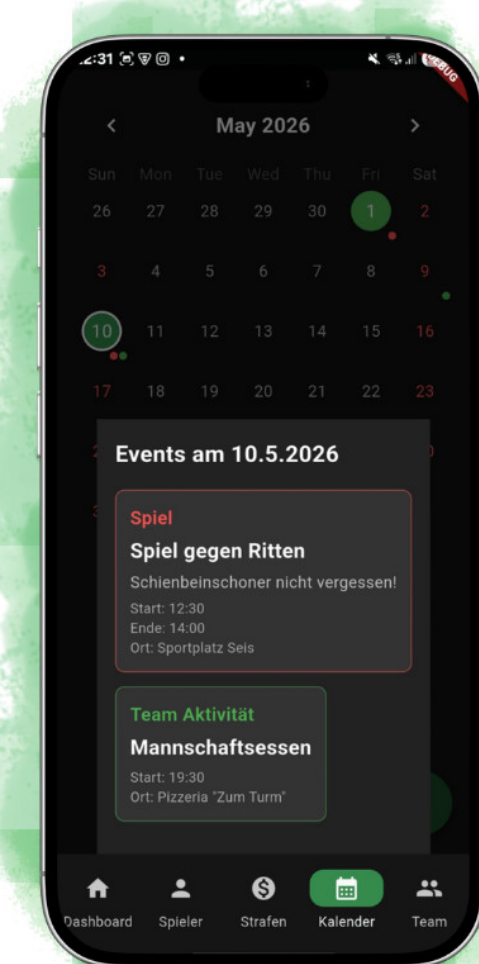
- C# (.NET)
- Bereitstellung einer API
- Rollenbasierte Zugriffskontrolle (Manager / Spieler)

Datenbank:

- PostgreSQL
- Speicherung von Event-Daten
- Strukturierte Tabellen für Teams, Benutzer etc.
- Lokale Offline-Datenbank: SQLite

Hosting & Deployment:

- Backend-Deployment über Railway
- Cloudbasiertes Hosting mit automatischem Build und Restart
- Zentrale Erreichbarkeit für alle Clients



FRONTEND
Flutter

BACKEND

DATENBANK

Projektteam:

Moritz Kob
Lukas Gasser



max valier
TFO BOZEN