Projektteilnehmer:

Julia Teichrib, Demian Lang, Maher Eata,

Abdollah Salehi, Milena Greulich, Steve Evans

KLARAKTIV Sportapp

Entwicklung einer mobilen Applikation zum Erfassen und Auswerten von Teilnahmedaten des Sporttags der Klara-Oppenheimer-Schule

Projektumfeld:

Klara-Oppenheimer-Schule

Stettiner Str. 1

97072 Würzburg

Inhalt

[1. Einleitung 1](#_Toc32926116)

[1.1. Projektanforderung 1](#_Toc32926117)

[1.2. Projektziel 1](#_Toc32926118)

[2. Projektplanung 1](#_Toc32926119)

[2.1. Projektphasen 1](#_Toc32926120)

[2.1.1. Anforderungsanalyse 1](#_Toc32926121)

[2.1.2. Design/Planung 2](#_Toc32926122)

[2.1.3. Implementation 2](#_Toc32926123)

[2.1.4. Test 2](#_Toc32926124)

[2.2. Projektvorgehen 2](#_Toc32926125)

[2.2.1. Ressourcenplanung 2](#_Toc32926126)

[2.2.2. Agiles Projektmanagement-Scrum 2](#_Toc32926127)

[3. Datenbanken 3](#_Toc32926128)

[4. HTML-Templates 3](#_Toc32926129)

[4.1. Login Lehrer 4](#_Toc32926130)

[4.2. Login Admin 4](#_Toc32926131)

[4.3. Login Schüler 4](#_Toc32926132)

[5. Anhang I](#_Toc32926133)

[5.1. Relationales Datenbankmodell I](#_Toc32926134)

[5.2. Sprintdokumentation I](#_Toc32926135)

[5.2.1. Sprint 1 I](#_Toc32926136)

[5.2.2. Sprint 2 II](#_Toc32926137)

[5.2.3. Sprint 3 II](#_Toc32926138)

[5.2.4. Sprint 4 III](#_Toc32926139)

[5.2.5. Sprint 5 III](#_Toc32926140)

[5.3. Screenshots Frontend IV](#_Toc32926141)

[5.3.1. Schüler IV](#_Toc32926142)

[5.3.2. Station V](#_Toc32926143)

[5.3.3. Admin V](#_Toc32926144)

# Einleitung

## Projektanforderung

Jährlich wird in der Klara-Oppenheimer-Schule ein offizieller Sporttag veranstaltet. Ziel ist es den Schülern möglichst verschiedene Aktivitäten nahe zu bringen. Deswegen gibt es zahlreiche Stationen, die ausprobiert werden können, wie Fußball, trommeln, nähen oder Salat machen. Durch den Besuch dieser Stationen sammeln die Schüler Punkte. Am Ende des Tages werden die Punkte der Klassen zusammengerechnet und die drei aktivsten Klassen erhalten einen Preis.

Um die Punkteverteilung einfach und übersichtlich zu gestalten, gab es bereits im vergangenen Jahr eine Online-Applikation. Jedem Schüler wurde eine Nummer zugeordnet und auf einem Zettel zusammen mit einem Stationen Plan ausgeteilt. Hat er eine Station abgeschlossen, konnte er der dortigen Aufsichtsperson seine Nummer sagen. Diese wurde dann am Computer oder Smartphone eingegeben und dem Schüler wurde automatisch die passende Punktezahl zugeordnet.

Da jedoch auch diverse Fehler aufkamen, sollten die Erfahrungswerte aus dem vergangenem Jahr für eine neue Version genutzt werden. Als zusätzliches Feature sollte ein QR-Code Scanner hinzugefügt werden, sodass man die Nummer nicht mehr per Hand eingeben, sondern die Schüler einfach scannen kann.

## Projektziel

Das Projektziel allgemein ist eine Applikation mit Anmeldemöglichkeiten für Schüler, Lehrer und Admin. Der Schüler soll seine Nummer angeben können und dann den aktuellen Punktestand seiner Klasse und die Rangliste sehen.

Der Lehrer soll sich bei einer Station anmelden können und dann dort Eingabefelder für die Nummern und den Scanner sehen.

Für alle administrativen Funktionen soll es einen extra Admin User geben. Er kann Klassen anlegen und Löschen, Stationen anlegen und bearbeiten, die Schüler-Nummern generieren, Schüler krankmelden und die genaue Auswertung am Ende anschauen.

# Projektplanung

## Projektphasen

Das Projekt wurde zwischen September 2019 und März 2020 im während der Berufsschulzeit bearbeitet. Das Vorgehen dabei war agil und es wurde sich an Scrum orientiert. Aufgrund dieser Vorgehensweise wurden die einzelnen Berufsschulblöcke als Sprints behandelt, in denen jeweils die Aufgaben neu besprochen und neu verteilt wurde.

### Anforderungsanalyse

Zu Beginn des Projekts hat unser Team das Coding aus dem letzten Jahr erhalten, um sich einen Überblick über die Anforderungen zu verschaffen. Es erwies sich außerdem als vorteilhaft, dass wir selber Teilnehmer des Sporttags waren und mögliche Flaschenhals-Situationen, wie das eingeben der fünfstelligen Nummern, miterlebt haben. Weiterhin gab es eine Lehrerkonferenz mit Feedback zum Sporttag 2019, an dem zwei Teammitglieder teilnehmen durften und dadurch über Verbesserungswünsche informiert wurden.

Diese Phase fand über die ersten Wochen des Schuljahrs statt, als das Projekt noch nicht begonnen wurde, sondern der gesamten Klasse erstmal Theorie unterrichtet wurde.

### Design/Planung

Bevor eine Technik festgelegt wurde, musste man sich über das unterschiedliche Können und verschiedenen Stärken des Teams Gedanken machen. Da alle aus anderen Firmen kamen und mit anderen Programmiersprachen arbeiten, entschieden wir uns schließlich für HTML/CSS im Frontend mit Node.js, PHP und MySQL als Backend. Mithilfe von PowerPoint wurden zuerst Mock-ups für das Frontend angelegt, die dann auch den Sporttag Verantwortlichen gezeigt werden konnten. Anhand dieser Mock-Ups wurden Funktionen, Benennung, Routing und Datenverarbeitung dokumentiert.

### Implementation

Für die Implementierung gab es kleinere Gruppen im Team, die für die einzelnen Themen verantwortlich waren. Zeitlich mischten sich die Implementationsphase mit der Planung, da man bereits erste PHP Strukturen anlegen und Datenbankmodelle entwerfen konnte, als noch die Feinheiten der Mock-ups ausgearbeitet wurden.

### Test

Im Verlauf der Implementierung wurde immer wieder der Datenbankzugriff und die Kommunikation zwischen PHP und Node.js getestet. Weiterhin die automatische Generierung der Laufzettel für die mehr als tausend Schüler. Das Endprodukt ist zeitlich nicht fertig geworden und konnte dementsprechend nicht komplett getestet werden.

## Projektvorgehen

Wie bereits erwähnt, wurde im ersten Sprint das Projektteam in kleinere Gruppen unterteilt,

Als Tool für das agile Projektmanagement Scrum wurde die öffentliche Website Trello verwendet. Dort wurde ein Scrumboard angelegt und gepflegen, was einen dauerhaften Überblick über offene Aufgaben und TODOs ermöglichte.

### Ressourcenplanung

Um möglichst effizient arbeiten zu können, wurde die Aufgabenverteilung auf Einzelpersonen oder kleine Teams vorgenommen. Diese beschäftigten sich jeweils mit dem Frontend HTML und CSS, dem Backend in PHP, der Schnittstelle in JavaScript und der Datenbank in MySQL.

Nach abgeschlossener Aufgabe, wurden die Teams erweitert oder neuen Aufgaben zugewiesen. Anhand des Trello Boards und der Webanwendung zur Versionsverwaltung Github, konnte das ganze Team am Quellcode arbeiten.

### Agiles Projektmanagement-Scrum

#### Definition

Als Vorgehensweise für das Projekt wurde sich an Scrum orientiert. Dies ist ein Framework, welches das typische Wasserfall-Prinzip in der Software-Entwicklung ersetzt. Statt sequenziertem Ablauf des gesamten Projekts, plant man nur in kurzen Sprints, die maximal zwei Wochen lang sind. Am Anfang eines Sprints wird gemeinsam die Arbeitsmenge für diesen festgelegt, abhängig von den Fähigkeiten und Kapazitäten der einzelnen Entwickler. Sollen trotzdem nicht alle Aufgaben fertiggestellt werden könne, so werden sie mit in den nächsten Sprint übernommen. Am Ende der zwei Wochen folgt ein Feedback-Gespräch des gesamten Teams, um Probleme zu erkennen und Verbesserungen zu finden. Scrum lebt allgemein von einer sehr starken Feedback-Kultur und täglichen kurzen Absprachen, sogenannten Daily Scrums.

#### Anwendung im Projekt

Für die Sport Applikation wurde sich zwar an Scrum orientiert, allerding nur einzelne Aspekte übernommen. Aufgrund des Blockunterrichts in der Berufsschule, ließ sich die Arbeitszeit gut in Sprints unterteilen. Offene, erledigte und ausstehende Aufgaben des Sprints, ließen sich mit dem öffentlichen Online Tool Trello dokumentieren. Dort kann man Scrumboards anlegen und Aufgaben unter den Teammitgliedern verteilen und im Überblick behalten.

Allerdings wurden andere Aspekte, wie Daily Scrums nicht regelmäßig durchgeführt, sondern eher sporadisch, wenn man sich als gesamtes Team einmal absprechen musste.

# Datenbanken

Im Rahmen des Projekts wurde eine Datenbank mit fünf Tabellen angelegt. Diese dienen der Verwaltung der Schüler, Stationen, Administrator und Klassendaten.

Für jeden Schüler existiert ein Eintrag in der Tabelle *schueler* mit der eigenen ID *Schueler\_ID* als Primärschlüssel, seiner Klasse *Klasse\_ID* als Fremdschlüsseln, den erspielten Punkten *Schueler\_Punkte* und dem Boolean Wert *Schueler\_Krank*. Hier wird festgehalten, ob er krank ist und dementsprechend gar nicht teilnehmen kann. Der Name des jeweiligen Schülers wird aus Datenschutzgründen gar nicht verwendet, sondern dieser ist nur an der ID identifizierbar.

*Klasse\_ID* dient als Fremdschlüssel zu *schueler* und Primärschlüssel zu ihrer eigenen Tabelle *klasse*, wo zusätzlich noch die Schüleranzahl *Klasse\_Anzahlschueler* pro Klasse erfasst wird. Klasse\_ID hat als Typ einen varchar, sodass die richtigen Klassennamen, die auch aus Buchstaben bestehen, verwendet werden können.

Die Zwischentabelle *schuelerauswertung* enthält *Station\_ID*, *Schueler\_ID* und den Timestamp als zusammengesetzten Primärschlüssel für den Fall, dass ein Schüler eine Station mehrfach besucht.

Die Tabelle für die Stationen *station* hat den Primärschlüssel Station\_ID und speichert alle zugehörigen Daten. Dazu zählen *Station\_Name*, *Station\_Punkte*, *Station\_Ort*, *Station\_Zeitpunkt\_von*, *Station\_Zeitpunkt\_bis* und *Station\_Passwort*. Mithilfe der ID und des Passworts kann sich ein beliebiger Lehrer später bei einer Station einloggen.

Zueinander stehen die Tabellen in einer 1:n Verbindung.

Unabhängig von den restlichen Tabellen, existiert noch *administrator*, welche *Admin\_ID* und *Admin\_Passwort* enthält. Hier ist nur ein Eintrag vorhanden, da es ein allgemeines Admin Passwort gibt, was auf dem Admin Screen verwendet werden kann.

# JavaScript Coding

Zu jeder PHP-Datei, die das Frontend Coding enthält, existiert eine JavaScript Datei, die die Funktionen und das Routing der PHP Seiten im Frontend definiert und per JSON mit dem Backend kommuniziert.

## loginAdmin.js

Auf der Login Seite hat der Lehrer die Möglichkeit sich als Admin oder als eine Station anzumelden. Die Stationen werden in einem Dropdown Menü angezeigt. Sie müssen mit Aufruf der Seite bereits geladen werden. Für die Verbindung wurde die HTTP Request Variable *xmlhttp* erstellt. Weiterhin wurden die folgenden Funktionen angelegt:

* open() und send(): Diese Funktionen werden bei jedem Request an das PHP Backend verwendet. Man legt fest, ob es sich um ein GET oder ein POST handelt und welche Daten genau zurückgegeben werden. Daraus baut sich die Request URL auf.
* onreadystatechange(): Auch diese Funktion wird mehrfach automatisch aufgerufen. Nach jedem Verwenden der Variable xmlhttp, wird hier erst der Status der Verbindung überprüft und dann das jeweilige Coding ausgelöst.
* loginLehrer(): Zunächst wird das eingegeben Passwort abgefragt. Wurde keines eingegeben, erscheint eine Fehlermeldung. Das Passwort wird zusammen mit der ausgewählten Station Teil der Request URL. Im Backend wird auf Richtigkeit überprüft und ein Boolean Wert zurückgegeben. Ist dieser true, wird zur nächsten Seite weitergeleitet, ansonsten gibt es eine Fehlermeldung
* loginAdmin(): loginAdmin() funktioniert genau, weil loginLehrer(). Allerdings ist es ein extra Feld mit einem eigenen Button, sodass nur das Passwort an PHP übergeben werden muss.

Mit Aufruf der Seite wird bereits die erste Request für alle Stationen an das Backend geschickt, damit der Lehrer auch alle Stationen in dem Dropdown Menü angezeigt kriegen kann. Das Füllen dieser Tabelle findet mithilfe einer FOR-Schleife im ersten Aufruf von onreadystatechange() statt. Die anderen Funktionen reagieren auf Knopfdruck.

Open(), send() und onreadystatechange() werden in jeder JavaScript Datei benutzt, die mit dem Backend kommuniziert und wird im Folgenden nicht jedes Mal erwähnt werden.

## adminMenu.js

Nach erfolgreichem Login des Admins, kommt er zu einem Menü, in dem er seine nächste Aktion wählen kann. Funktional handelt es sich hier hauptsächlich um Routing. Es wurden die folgenden Funktionen entsprechend des Routings angelegt:

* admStation(): Routing zur Stationenverwaltung (stationen.php)
* admKlassen(): Routing zur Klassenverwaltung (admin\_klasse\_verwalten.php)
* genNummer(): Generiert die Laufzettel für die Schüler
* krank(): Routing zur Krankheitsadministration (krankMeldung.php)
* auswertung(): Routing zur Auswertung des Endergebnisses (schuelerAuswertung.php)

## Stationen.js

Weitergeleitet von dem Admin Menü, werden hier die Stationen verwaltet. Man kann per Dropdown eine auswählen und bearbeiten oder eine neuanlegen. Dafür wurden folgende Funktionen benutzt.

* admNeueStation(): Routing zum Anlegen einer neuen Station ()
* admBearbeiten(): Routing zu selben Seite (edit\_station.php), nur werden diesmal die die Stationsdaten in die leeren Felder geladen, damit man sie bearbeiten kann

## edit\_station.js

Hier werden Stationen geladen oder neuangelegt. Dafür gibt es diese Funktionen:

* getUrlVars():
* verwerfen(): Routing zurück zur Stationenverwaltung (stationen.php)
* speichern(): Eingegebene oder veränderte Daten werden über document.getElementById() aus den Feldern ausgelesen, an das Backend geschickt und dadurch in der Datenbank angepasst.

## klassenMenu.js

Dies ist eine weitere Menü-Seite, die nur zur Bedienungsfreundlichkeit für den User existiert. Dementsprechend gibt es nur Routingfunktionen:

* newClass(): Routing zum Anlegen einer neuen Klasse (adminNeuKlasse.php)
* Bearbeiten(): Routing zum Bearbeiten einer Klasse (schuelerAnzahlBearbeiten.php)
* del(): Routing zum Löschen einer Klasse (klassenLoeschen.php)

# HTML-Templates

Die HTML-Templates wurden auf Grundlage der zuvor in Microsoft Visio erstellten Mockups mit Hilfe von Bootstrap Studio erstellt. Bootstrap Studio ist eine Desktop App, die viele Komponenten des Bootstrap Frameworks enthält. Per Drag and Drop können Webseiten individuell erstellt und der HTML-Code kann anschließend exportiert werden.

Bei der Erstellung der Templates wurde vor Allem darauf geachtet, dass die Seiten auf mobilen Geräten dargestellt werden können und der User dabei möglichst nicht scrollen muss.   
Es wurde sich überwiegend an den in Bootstrap Studio voreingestellten Auflösungen für das Galaxy S10 (360px × 760px) und das iPhone 8 (375px × 667px) orientiert, wobei aber darauf geachtet wurde, dass die Seiten auch auf dem Desktop ordentlich angezeigt werden.

Für einen einheitlichen Look über alle Seiten wurden eigene CSS-Klassen für Überschriften und verschiedene Buttons in der *styles.css* erstellt und vorhandene Klassen von Bootstrap oder Eigenschaften von Elementen überschrieben.

Für die Elemente *html* und *body* wurde eine Höhe von 100%, für die Elemente *nav* und *footer* jeweils eine Höhe von 10% und für das Element *main* eine Mindesthöhe von 80% festgelegt. Damit wurde sichergestellt, dass der Inhalt auf der gesamten Seite verteilt ist, sodass die navbar am oberen und der footer am unteren Rand der Seite platziert sind, unabhängig von der Displaygröße.

Es gibt rote, grüne und orange Buttons. Rote und grüne Buttons haben mit Aktionen, die sich auf die Datenbank beziehen zu tun, wie Speichern oder Löschen. Orange Buttons dienen lediglich der Navigierung zwischen den verschiedenen Seiten.

## Login Lehrer

Der Login für Lehrer ist aufgeteilt in Station und Admin. Lehrer, die für eine bestimmte Station verantwortlich sind, können sich mit der jeweiligen Station und dem dazugehörigen Passwort anmelden und werden anschließend auf die Seite *stationscannen.php* weitergeleitet. Dort sehen sie den Namen ihrer Station und die Anzahl der Punkte, die die Schüler für die Teilnahme erhalten.   
Sie können außerdem auf die Checkliste der Station zugreifen, die als PDF hinterlegt ist. Diese Checkliste wird von den Lehrern vorher bereitgestellt und enthält Informationen, die für die Vorbereitung der jeweiligen Station benötigt werden. Die Punktevergabe geschieht entweder durch Scannen des QR-Codes oder durch Eingabe der Schülernummer in eines der fünf vorhandenen Felder.

## Login Admin

Für den Admin-Login muss lediglich das Administrator-Passwort eingegeben werden. Der Admin kann anschließend von einem Menü aus Stationen und Klassen anlegen bzw. bearbeiten oder auch Klassen löschen, die Schülernummern und Laufzettel generieren, Schüler krankmelden und die Auswertung einsehen.  
In der Auswertung werden die drei besten Klassen und die Top 3 aller Schüler angezeigt.

## Login Schüler

Schüler können sich auf einer anderen Seite mit ihrer Schülernummer einloggen und ihren Punktestand, sowie den aktuellen Rang ihrer Klasse, die Bestpunktzahl aller Schüler und die Top 3 aller Klassen einsehen. Auf der Login-Seite gibt es außerdem die Buttons „Klassenübersicht“ und „Stationen“.   
Hier können zum einen die Punkte der einzelnen Schüler zu den jeweiligen Klassen eingesehen werden und wie viele Schüler einer Klasse krank oder inaktiv sind. Diese Übersicht ist vor Allem für die Klassenlehrer gedacht, damit diese die Leistung ihrer Klasse beurteilen können. Weil keine persönlichen Daten enthalten sind, ist die Übersicht aber für alle zugänglich.   
Zum anderen liefert der Button „Stationen“ den Schülern eine Übersicht aller Stationen mit allen wichtigen Informationen, sodass diese jederzeit für die Schüler zugänglich sind.

# Fazit

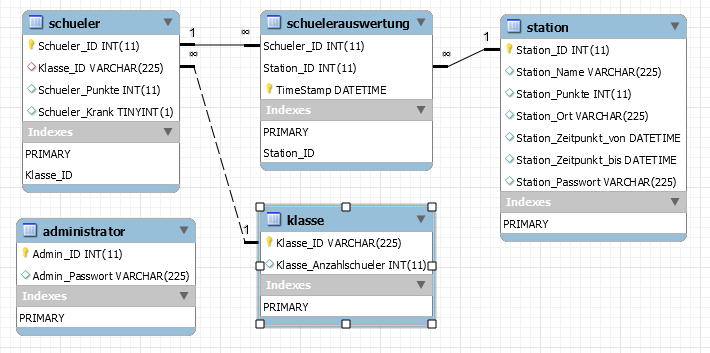
Die Sport Applikation entstand auf Initiative der Verantwortlichen für den Klaraktiv Tag, um den Schülern und den Lehrern eine einfache Organisation und reibungslosen Ablauf zu ermöglichen. Im Verlauf des Projekts sind wir jedoch auf einige Probleme gestoßen. Mangel an Absprachen führten zu Dokumenten in unterschiedlichen Versionen. Die Namensgebung wurde nicht eingehalten, sodass am Ende noch viel Zeit in das Sortieren der PHP und JavaScript Dateien floss. Auch wurden manche Prozesse nicht genau überdacht und die HTML Templates mussten hinterher noch einmal aufgenommen und angepasst werden. Komplikationen entstanden ebenfalls durch das anfängliche Unterteilen in Themenverantwortliche aufgrund der Kompetenzen. Dadurch hat jeder zuerst für sich gearbeitet ohne Input des restlichen Teams. Später wurde es dann schwer sich in die anderen Themengebiete einzuarbeiten und sich gegenseitig zu unterstützen. In Kombination mit vielen Krankheitsfällen führte das zu einer starken zeitlichen Verzögerung, die trotz Zusatzstunden nicht eingeholt werden konnte.

Positiv dagegen verlief die Definition der Datenbank und der Beziehungen der Tabellen. Auch die separate Generierung der Laufzettel war erfolgreich und kann später von anderen Klassen aufgegriffen werden. Die HTML Templates kann man ebenfalls weiterverwenden. Es müssen maximal noch Kleinigkeiten angepasst werden, bei denen die generierte Version von Bootstrap Studio möglicherweise nicht den genauen Anforderungen der Entwickler entspricht.

Zusammenfassend lässt sich sagen, dass die Fertigstellung an verschiedenen Stellen gescheitert ist, aber die Teammitglieder dennoch wichtige Rückschlüsse aus dem Projekt ziehen konnten. Kommunikation steht in einer Teamarbeit an oberster Stelle.

# Anhang

## Relationales Datenbankmodell

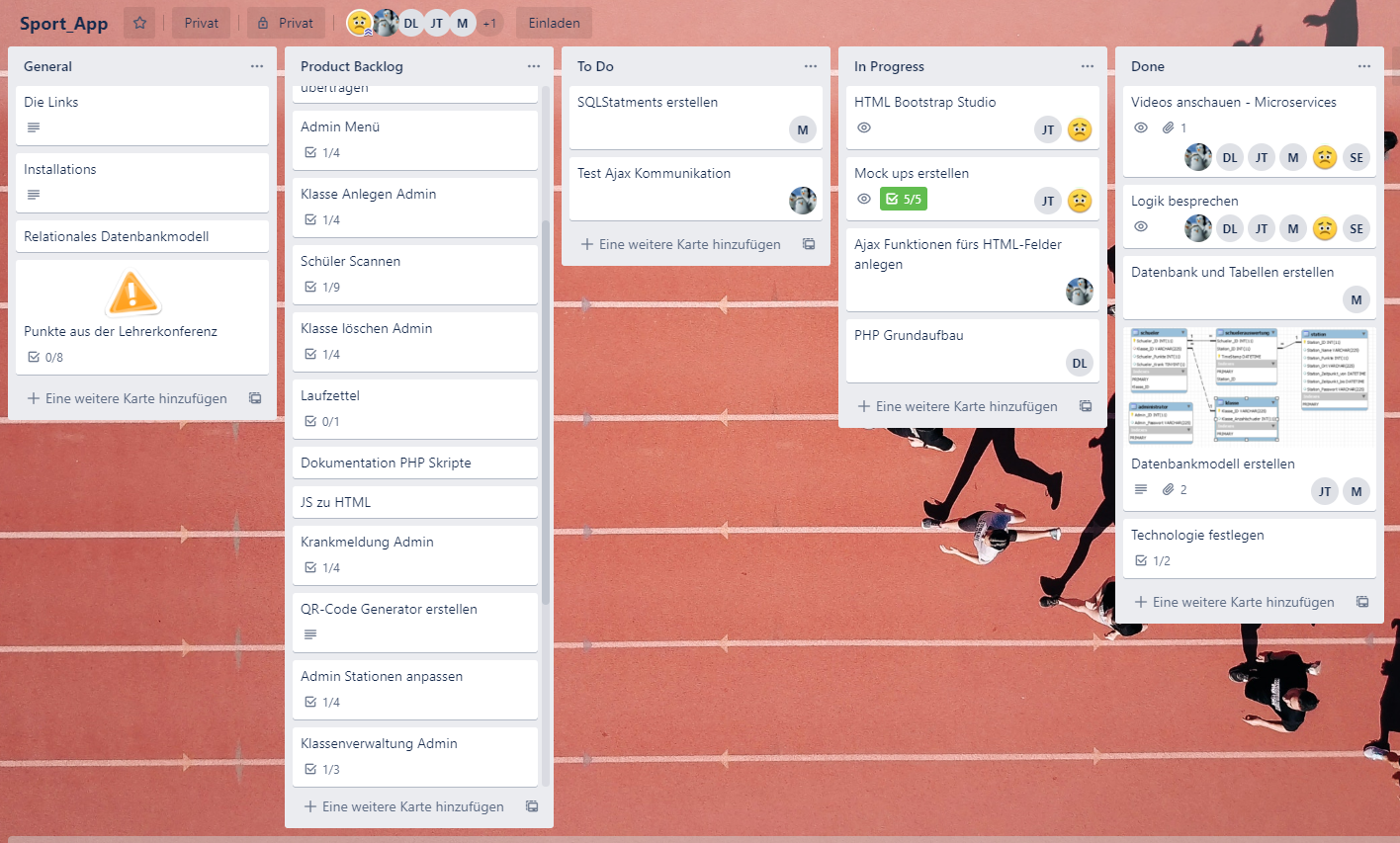


## Sprintdokumentation

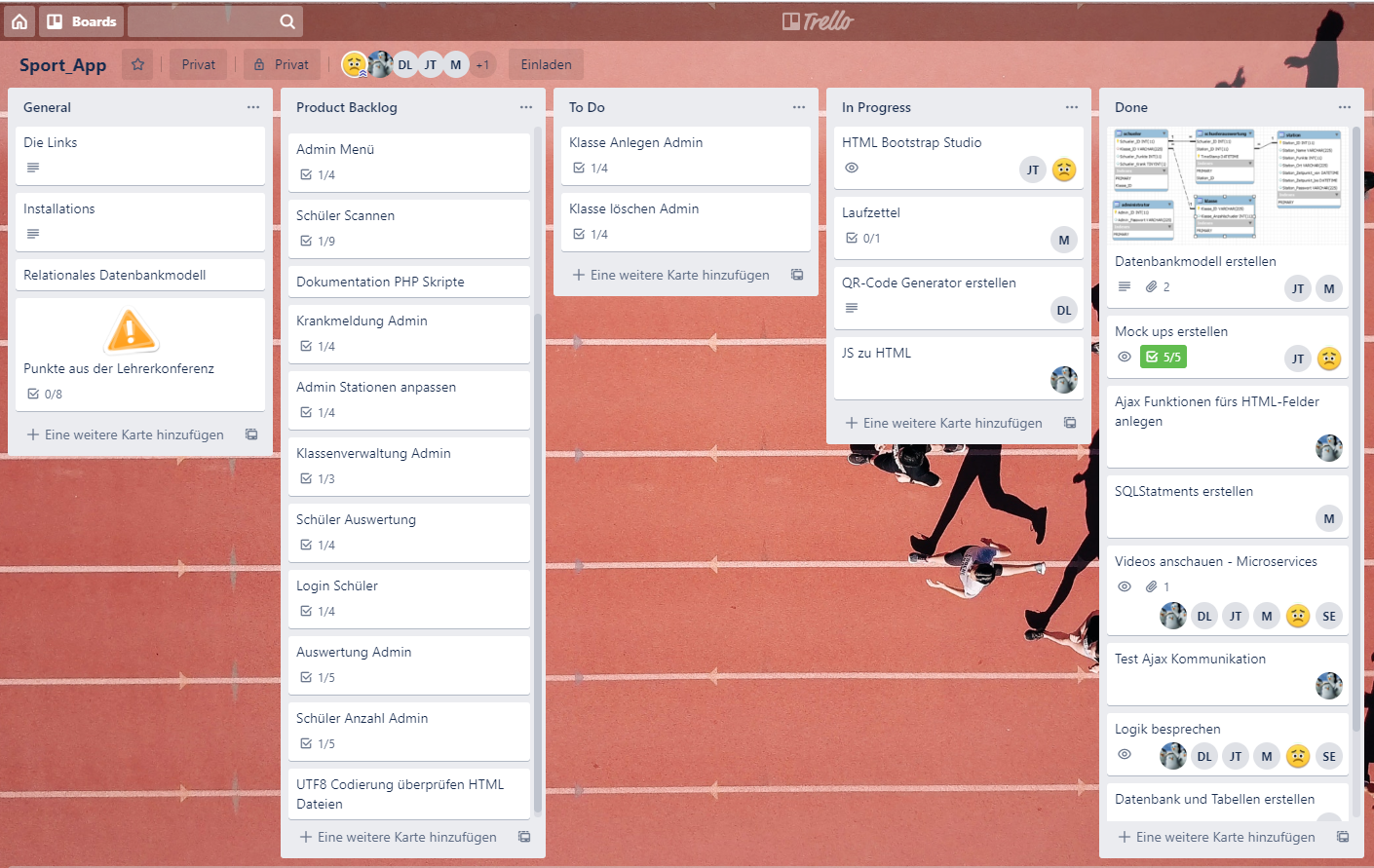
### Sprint 1



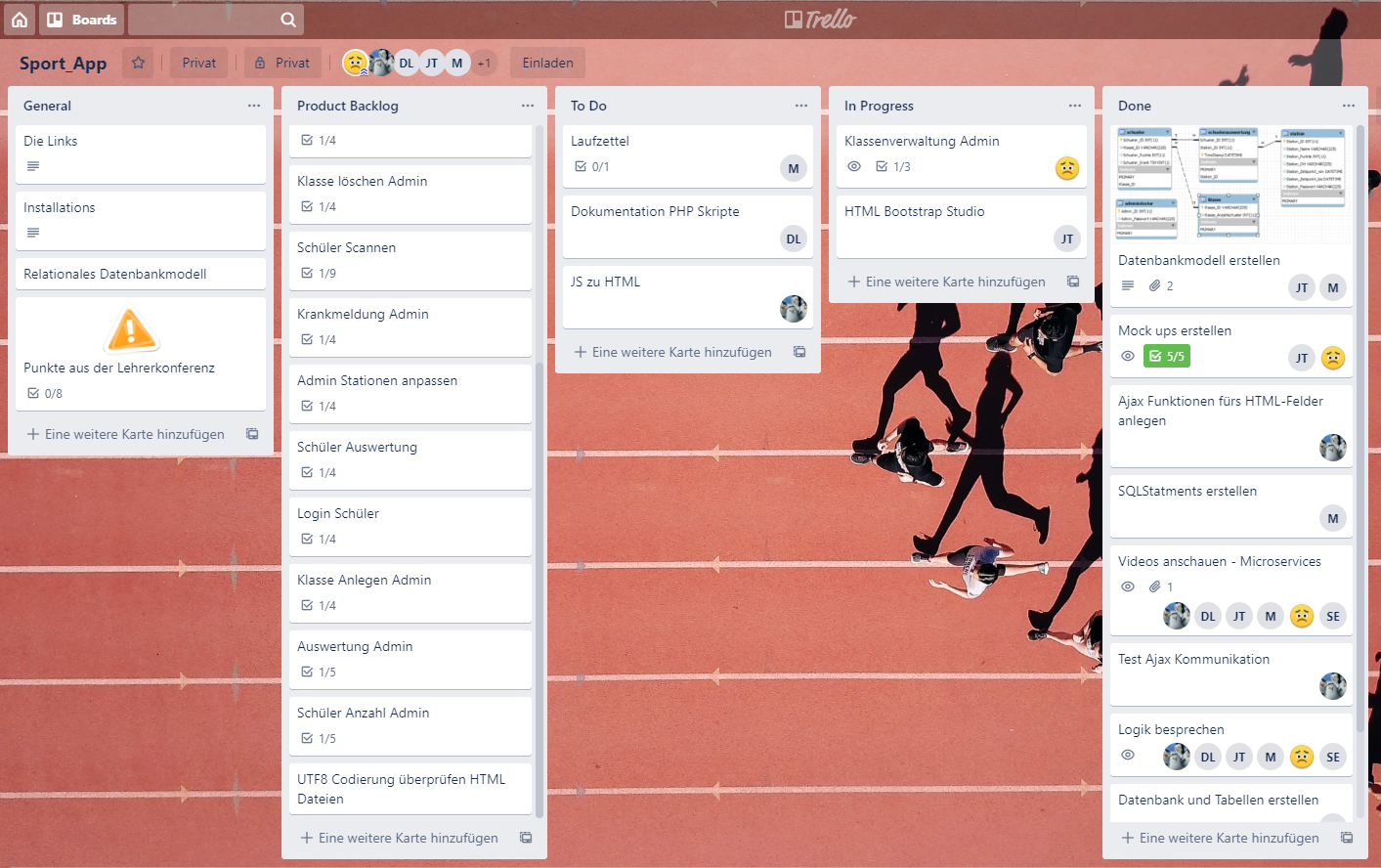
### Sprint 2



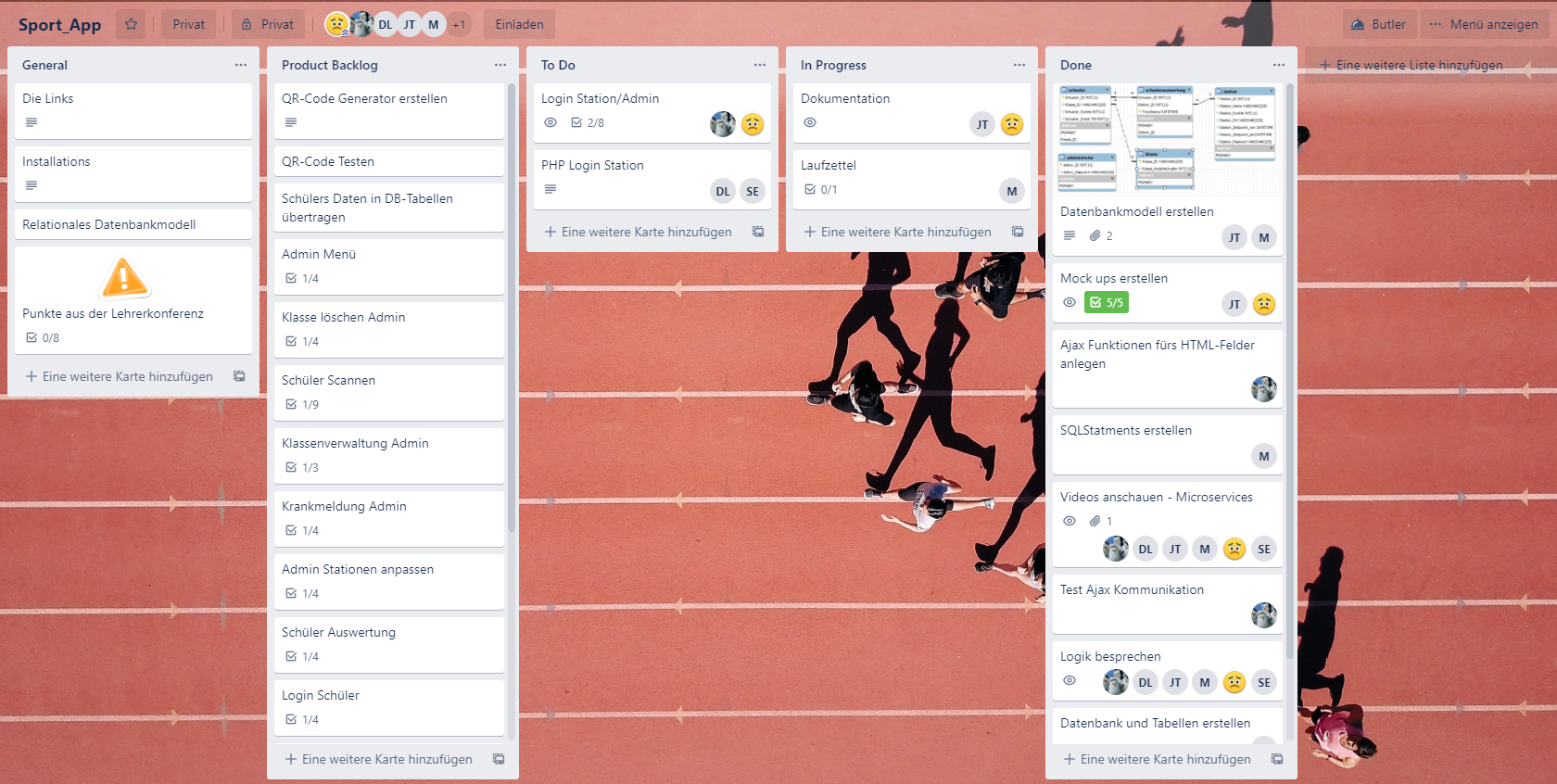
### Sprint 3



### Sprint 4

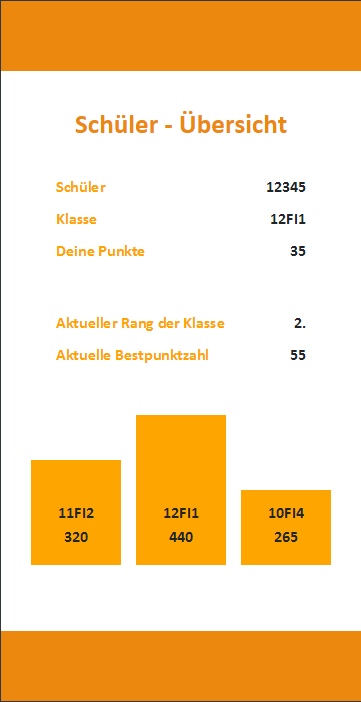


### Sprint 5

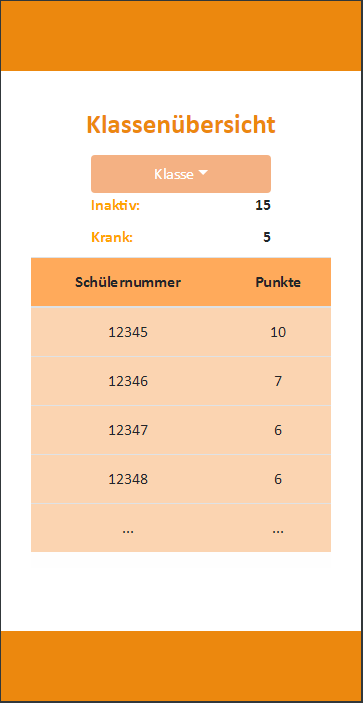


## Screenshots Frontend

### Schüler

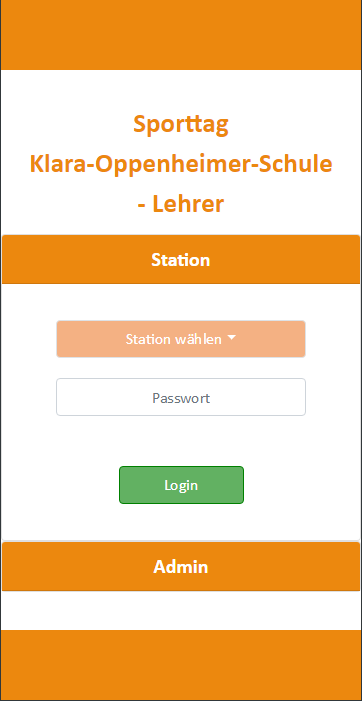
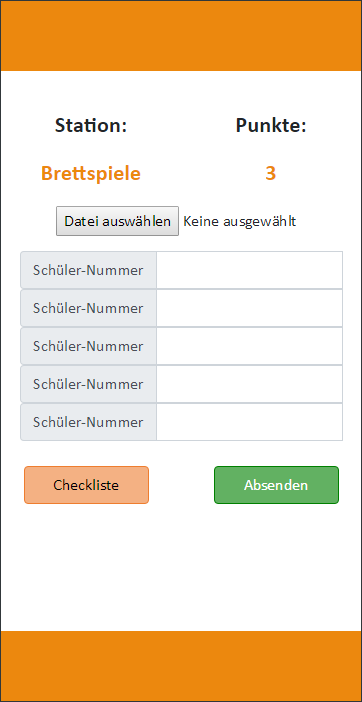
 

schueler\_login.php schuelerUebersicht.php

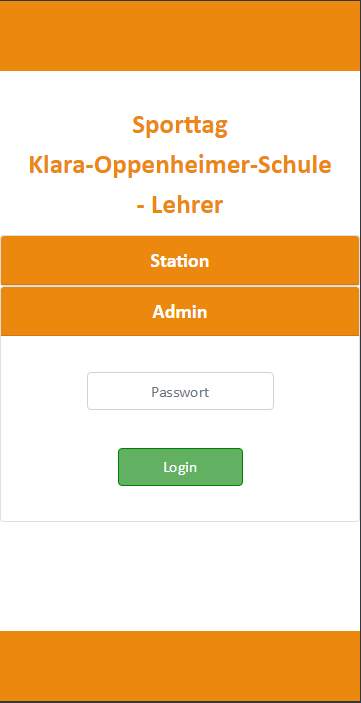
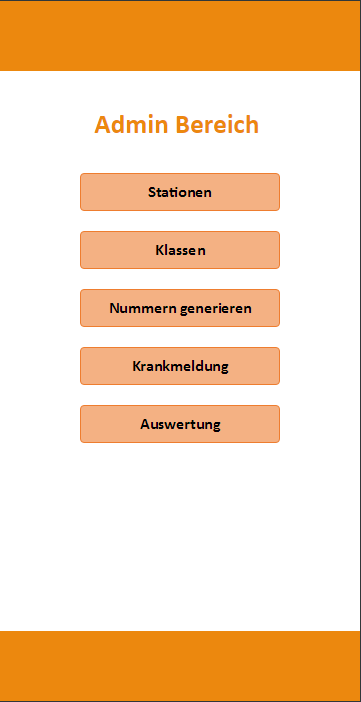
klasseUebersicht.php stationenUebersicht.php

### Station

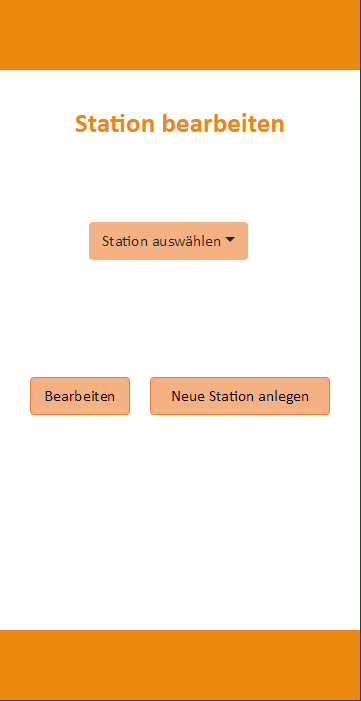
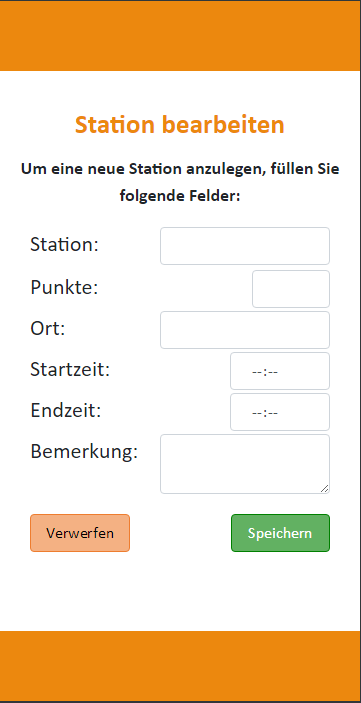
admin\_login.php stationscannen.php

### Admin

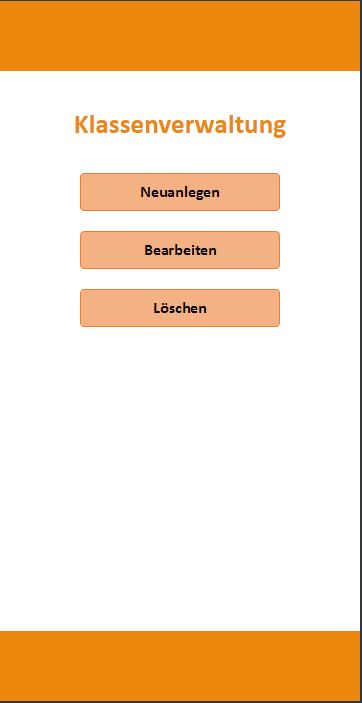
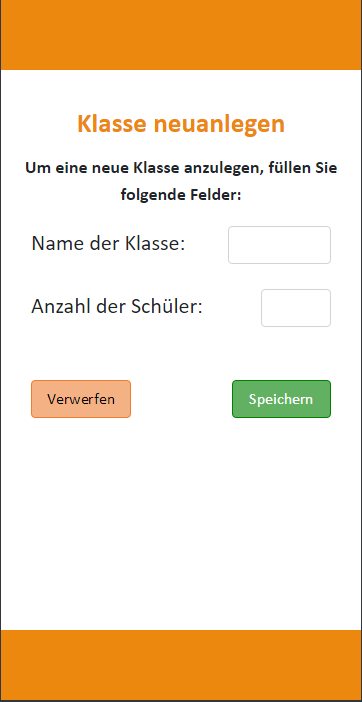
admin\_login.php admin\_menu.php

#### Stationen

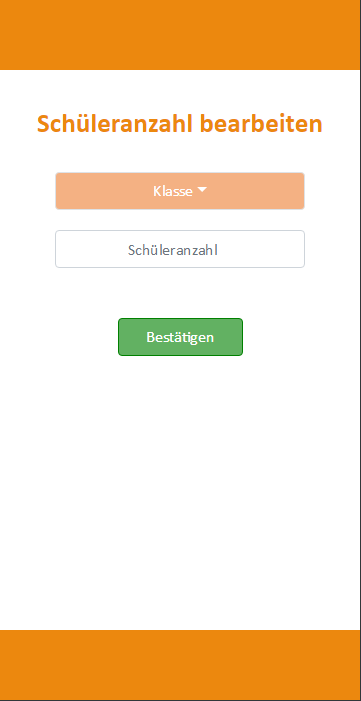
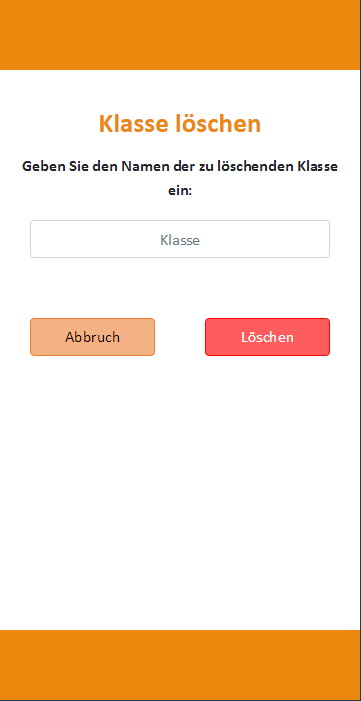
 

stationen.php admin\_station\_erstellen.php

#### Klassen

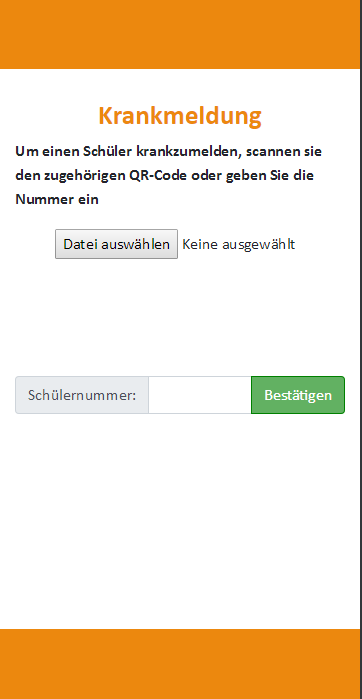
 

admin\_klasse\_verwalten.php adminNeuKlasse.php

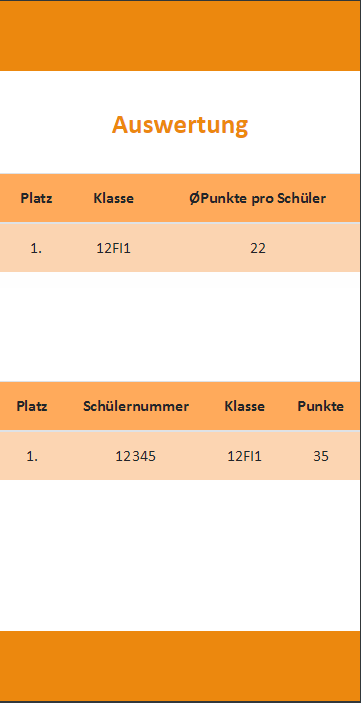
schuelerAnzahlBearbeiten.php klassenLoeschen.php

#### Krankmeldung



krankMeldung.php

#### Auswertung



schuelerAuswertung.php