



PROJEKTARBEIT

03 KONZERTTICKETS



Noah Buchs, Dawid Kapka, Leonie Roth
CSS-VERSICHERUNG

Inhaltsverzeichnis

<u>EINLEITUNG</u>	<u>2</u>
<u>SITEMAP</u>	<u>2</u>
<u>FORMULARE</u>	<u>2</u>
PLANNING (SKIZZE)	2
1-TICKETS (/OVERVIEW)	3
2-CREATE TICKET (/CREATE)	3
3-TICKET PREVIEW (/CREATE)	3
<u>VALIDIERUNG</u>	<u>4</u>
1-TICKETS (/OVERVIEW) & 2-CREATE TICKET (/CREATE)	4
3-TICKET PREVIEW (/CREATE)	4
<u>DATENBANK</u>	<u>4</u>
<u>TESTFÄLLE</u>	<u>5</u>
1-PLUS-BUTTON AUF TICKETS WIRD GEDRÜCKT	5
2-IM FORMULAR WERDEN FEHLERHAFTE ODER FALSCH E DATEN ANGE GEBEN	5
3-TICKET KANN IN TICKET PREVIEW BEARBEITET WERDEN.....	5
4-BENUTZERDATEN VOM FORMULAR WERDEN IN DIE DATENBANK GESPEICHERT	5
5-NACH DEM CONFIRM-BUTTON WEITERLEITEN AUF TICKETS-SEITE.....	5
6-TICKET AUF DER TICKETS-SEITE BEARBEITEN	5
7-DRÜCKEN AUF DISCARD-BUTTON ALLE ÄNDERUNGEN VERWERFEN.....	5
8-DER STATUS DES TICKETS KANN BEARBEITEN WERDEN	6
9-TICKETS MIT DEM STATUS BEZAHLT WERDEN NICHT MEHR AUF DER TICKETS-SEITE ANGEZEIGT	6
10-ALS NUTZER MÖCHTE ICH DAS ÄLTESTE TICKET ZU OBERST SEHEN.....	6
<u>TESTBERICHT.....</u>	<u>6</u>
<u>ROADMAP</u>	<u>7</u>
<u>BEKANNTE FEHLER</u>	<u>7</u>

Einleitung

Dieses Projekt wurde im Rahmen des ÜK 307 umgesetzt. Unsere Gruppe hat sich für das Projekt Konzerttickets entschieden, welches die Gruppennummer 3 hat. Das Projekt wurde mit Angular, NestJS und MySQL umgesetzt. Am Ende des ÜK haben wir unser Projekt auf unserem Lehnringsserver zur Verfügung gestellt.

Sitemap

Titel	Route (URL)	Inhalt/Zweck
Tickets	http://css-appli20.com/overview	Die Anzeige aller unbezahlten Tickets.
Create Ticket	http://css-appli20.com/create	Das Formular, um ein neues Ticket zu erstellen.

Formulare

Planning (Skizze)

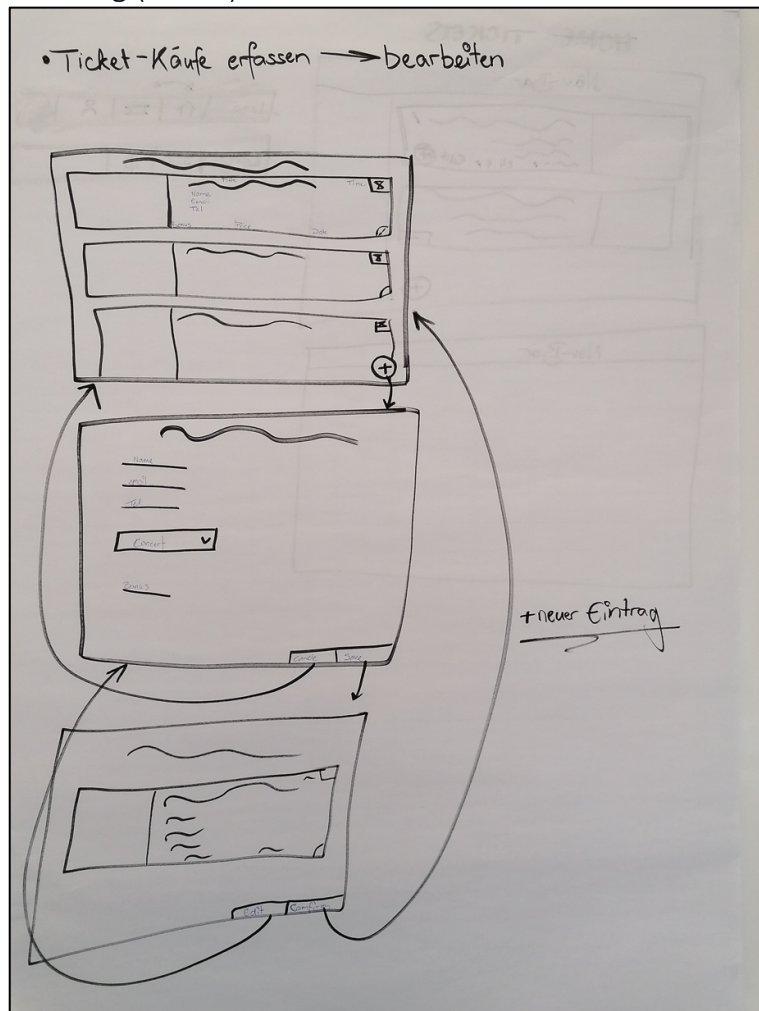


Abbildung 1: erste Skizze der verschiedenen Seiten

1-Tickets (/overview)

Titel

Date

Firstname:
Lastname:
Email:
Telefon:

Price

Date

Titel

Firstname:
Lastname:
Email:
Telefon:

Titel

Firstname:
Lastname:
Email:
Telefon:

+

2-Create Ticket (/create)

New Ticket

Firstname:

Lastname:

Email:

Telefon:

Concert:

Concert

Bonus:

Cancel

Save

3-Ticket Preview (/create)

New Ticket

Titel

Firstname:
Lastname:
Email:
Telefon:

Edit

Save

Validierung

1-Tickets (/overview) & 2-Create Ticket (/create)

Formularfeld: Firstname

Validierung: Nicht leer/mindestens zwei Zeichen

Formularfeld: Lastname

Validierung: Nicht leer/mindestens zwei Zeichen

Formularfeld: Email

Validierung: Enthält @-Zeichen und mindestens einen .

Formularfeld: Telefon

Validierung: Kann leer sein/Falls nicht, enthält nur Nummer, Leerzeichen und +

Formularfeld: Concert

Validierung: Nicht leer/ein Konzert ausgewählt

Formularfeld: Bonus

Validierung: Nicht leer

3-Ticket Preview (/create)

Formularfeld: -

Validierung: -

Datenbank

Feld-Bezeichnungen	Feld-Typ-Definition
id	int auto_increment
firstname	varchar (255)
lastname	varchar (255)
email	varchar (255)
phone	varchar (255)
concertId	int
status	varchar (255)

- PRIMARY KEY -> id
- FOREIGN KEY -> concertId
- REFERENCES -> concerts(id)

Feldbezeichnung vorgegebene DB concerts	Feld-Typ-Definition
id	int (11) NOT NULL
artist	varchar (255) NOT NULL

Testfälle

1-Plus-Button auf Tickets wird gedrückt

GIVEN: Der Nutzer befindet sich auf der Tickets-Seite.

WHEN: Der Nutzer drückt den Plus-Button.

THEN: Der Nutzer wird zur Create Ticket weitergeleitet, um ein Ticket zu erstellen.

2-Im Formular werden fehlerhafte oder falsche Daten angegeben

GIVEN: Der Nutzer gibt falsche oder fehlerhafte Daten im Formular für ein neues Ticket ein.

WHEN: Der Nutzer den Save-Button drückt.

THEN: Erscheinen unten in einer roten Box Meldungen mit den Feldern, welche fehlerhaften Eingaben beinhalten.

3-Ticket kann in Ticket Preview bearbeitet werden

GIVEN: Der Nutzer hat ein Ticket erstellt und auf den save-Button gedrückt.

WHEN: Der Nutzer auf den Stift-Button drückt, kann das Ticket bearbeiten werden. Die Änderungen werden mit dem Save-Button gespeichert.

THEN: Die Änderungen werden in der Tickets-Seite gespeichert, in Form eines neuen Tickets.

4-Benutzerdaten vom Formular werden in die Datenbank gespeichert

GIVEN: Der Nutzer drückt nach dem Erstellen eines Tickets den Save- und anschliessend den Confirm-Button.

WHEN: Der Nutzer das Ticket speichert.

THEN: Werden die Ticketinformationen in der Datenbank gespeichert und anschliessend auf der Tickets-Seite ausgegeben.

5-Nach dem Confirm-button weiterleiten auf Tickets-Seite

GIVEN: Der Nutzer hat ein neues Ticket erstellt und ist auf der Ticket Preview-Seite.

WHEN: Der Nutzer den Confirm-Button drückt.

THEN: Wird er auf die Tickets-Seite weitergeleitet und sieht die anderen Tickets.

6-Ticket auf der Tickets-Seite bearbeiten

GIVEN: Der Nutzer befindet sich auf der Tickets-Seite und es besteht bereits ein Ticket.

WHEN: Der Nutzer auf den Stift-Button drückt, kann er das Ticket bearbeiten. Anschliessend wird der Save-Button gedrückt, um die Änderungen zu speichern.

THEN: Das Ticket wird auf der Datenbank angepasst und das bearbeitete Ticket erscheint auf der Tickets-Seite.

7-Drücken auf Discard-Button alle Änderungen verwerfen

GIVEN: Der Nutzer bearbeitet das Ticket in der Ticket Preview oder in Tickets.

WHEN: Der Nutzer nach dem Drücken des Stift-Button und dem Ändern der Informationen den Discard-Button drückt.

THEN: Werden alle Änderungen verworfen.

8-Der Status des Tickets kann bearbeitet werden

GIVEN: Der Nutzer hat bereits das Formular für ein Ticket erstellt und befindet sich auf der Ticket Preview-Seite.

WHEN: Der Nutzer den Stift drückt, kann er den Status der Zahlung anpassen. Anschliessend werden die Änderungen gespeichert.

THEN: Der Status der Sanduhr auf dem Ticket wird geändert.

9-Tickets mit dem Status bezahlt werden nicht mehr auf der Tickets-Seite angezeigt

GIVEN: Es besteht ein Ticket, bei dem man den Status in Ticket Preview oder Tickets bearbeiten kann.

WHEN: Der Nutzer den Status auf 'paid' wechselt.

THEN: Wird das Ticket auf der Tickets-Seite nicht mehr angezeigt.

10-Als Nutzer möchte ich das älteste Ticket zuoberst sehen

GIVEN: Es sind mehrere Tickets erstellt und der Nutzer befindet sich auf der Tickets-Seite.

WHEN: Der Nutzer mehrere Tickets, welche noch nicht gezahlt wurden, sieht.

THEN: Ist das älteste Ticket zuoberst.

Testbericht

Title Test	Datum/Zeit	Status
1-Plus-Button auf Tickets wird gedrückt	11.05.2022/11:58	✓
2-Im Formular werden fehlerhafte oder falsche Daten angegeben	11.05.2022/11:58	✓
3-Ticket kann in Ticket Preview bearbeitet werden	11.05.2022/11:59	✓
4-Benutzerdaten vom Formular werden in die Datenbank gespeichert	11.05.2022/11:59	✓
5-Nach dem Confirm-button weiterleiten auf Tickets-Seite	11.05.2022/12:00	✓
6-Ticket auf der Tickets-Seite bearbeiten	11.05.2022/12:00	✓
7-Drücken auf Discard-Button alle Änderungen verwerfen	11.05.2022/12:00	✓
8-Der Status des Tickets kann bearbeitet werden	11.05.2022/12:01	✓
9- Tickets mit dem Status bezahlt werden nicht mehr auf der Tickets-Seite angezeigt	11.05.2022/12:01	✓
10- Als Nutzer möchte ich das älteste Ticket zuoberst sehen	11.05.2022/12:02	✓

Welche Funktionen konnten nicht integriert werden (inkl. Zusatzfunktionen)?

➔ Ebenfalls wurden automatisierte Tests erstellt und erfolgreich durchgeführt.

Roadmap

	09.05.2022	10.05.2022	11.05.2022
Vormittag	-	Frontend, Datenbank	Validierung/Cleanup
Nachmittag	Aufsetzen, Planung Projekt	Daten anzeigen ->Frontend, Backend verbinden	Präsentation

Bekannte Fehler

Bei der Bearbeitung eines Tickets, kann man beim Konzert und dem Datum teilweise die Schrift nicht lesen. Dies ist jedoch Gerät und Browser abhängig, weshalb wir diesen «Fehler» nicht beheben konnten.