Jonathan Roth, Nikolaos Papadopoulos

Ski-Service

Partnerarbeit im Modul 133

Inhaltsverzeichnis

[Informieren 2](#_Toc59095704)

[Aufgabenstellung 2](#_Toc59095705)

[Zusätzliche Aufgaben 2](#_Toc59095706)

[Hilfestellung 2](#_Toc59095707)

[Planen 2](#_Toc59095708)

[Datenbankdiagramme 2](#_Toc59095709)

[Nikolaos’ ERM 3](#_Toc59095710)

[Runas ERM 3](#_Toc59095711)

[Aussehen 4](#_Toc59095712)

[Entscheiden 5](#_Toc59095713)

[Datenbankdesign entscheiden 5](#_Toc59095714)

[Aussehen entscheiden 5](#_Toc59095715)

[Lösungsweg entscheiden 5](#_Toc59095716)

[Realisieren 5](#_Toc59095717)

[SQL-Datenbank 5](#_Toc59095718)

[Seiten erweitern 5](#_Toc59095719)

[Kontrollieren 6](#_Toc59095720)

[Funktion überprüfen 6](#_Toc59095721)

[Anforderungen erfüllt? 6](#_Toc59095722)

[Auswerten 6](#_Toc59095723)

[Realisierung 6](#_Toc59095724)

[Planung 6](#_Toc59095725)

# Informieren

## Aufgabenstellung

* Logik von die Jetstream-Service Webseite erweiterten.
* Auf der Datenbank die Informationen des Serviceauftrags und die Login Daten der Mitarbeiter speichern.
* Serviceaufträge anzeigen und nach Priorität sortieren.
* Ein Login Dialog mit Passwort codieren.
* Mitarbeiter können die Status eines Auftrags ändern
* Sicherheitsrelevanten Aspekte Berücksichtigen
* Datenbankstruktur soll in der 3.NF normalisiert sein inkl. Referenzieller Integrität
* Kein Benutzer root benutzten. (DML)
* Die Web Anwendung soll nach dem MVC Design Pattern strukturiert sein.

## Zusätzliche Aufgaben

* Eingeloggte Mitarbeiter können neue Benutzer eröffnen
* Gelöschte Aufträge werden nicht aus der Datenbank entfernt, sondern nur als gelöscht markiert.

## Hilfestellung

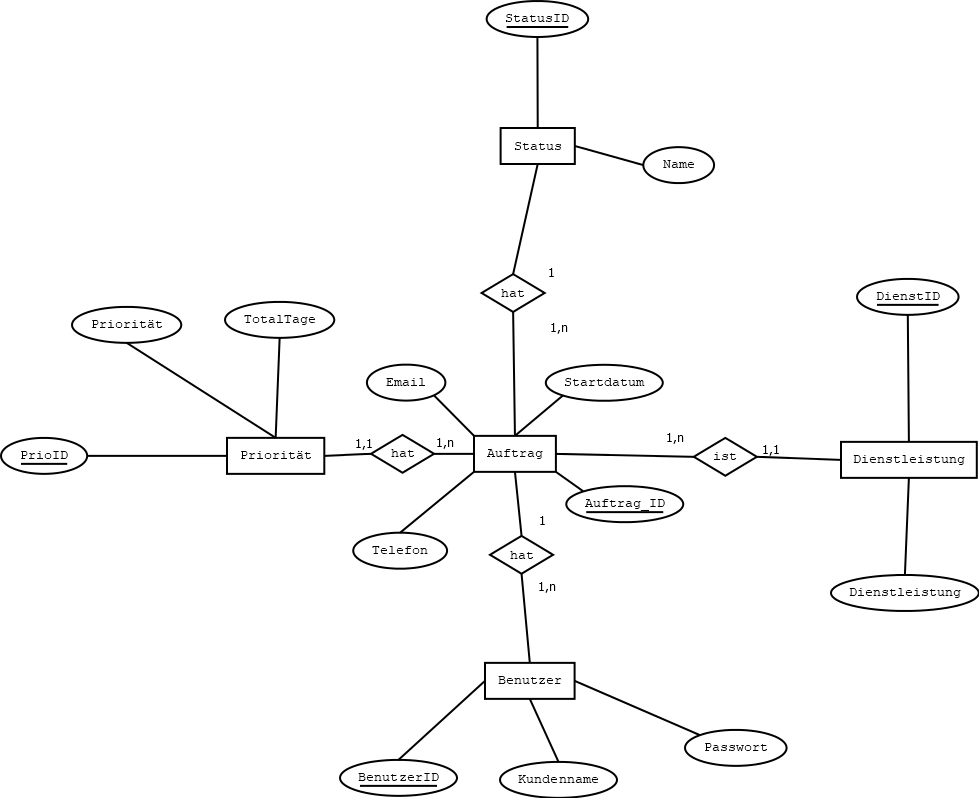
* Modulwissen
* OneNote
* W3School
* Google
* Klassenkameraden
* Herr Müller

# Planen

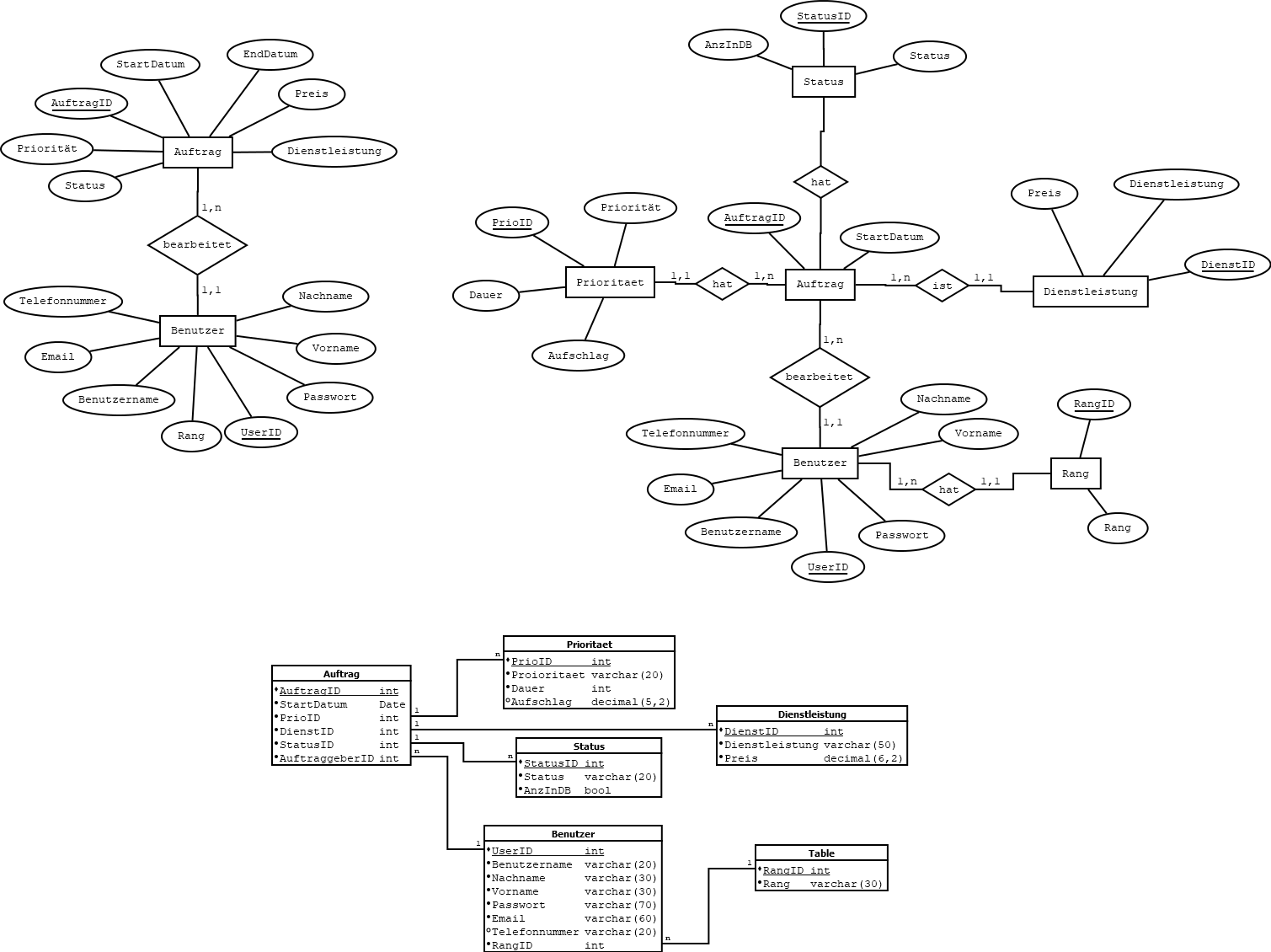
## Datenbankdiagramme

Um die Struktur der Datenbank zu ermitteln, haben wir unabhängig voneinander Datenbankdiagramme erstellt.

### Nikolaos’ ERM

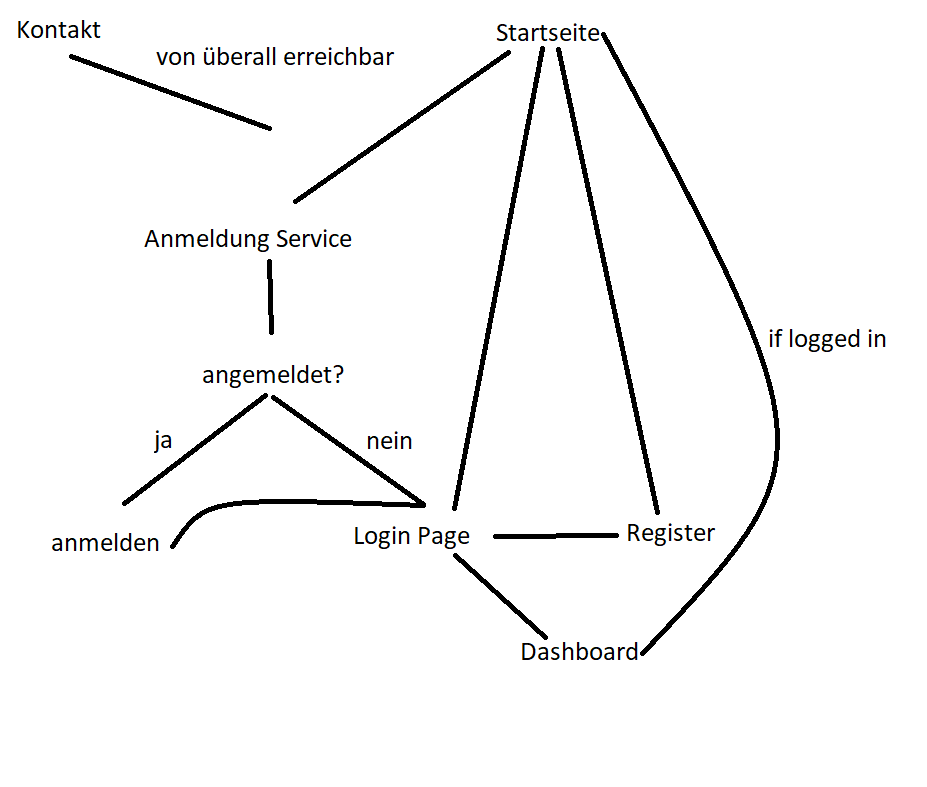
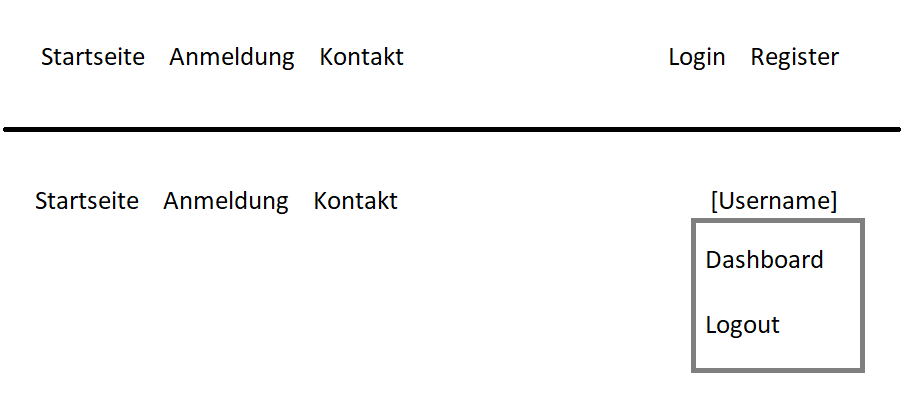


### Runas ERM

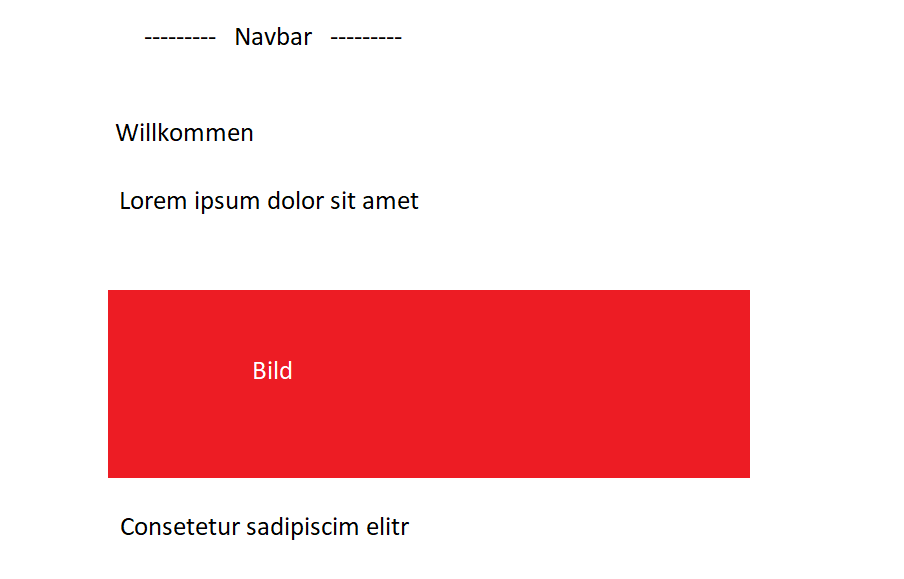
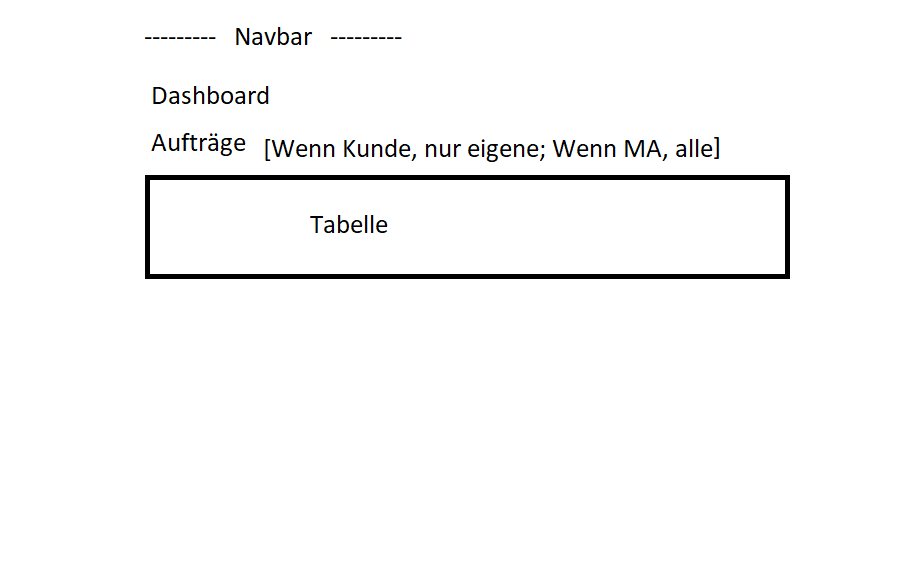


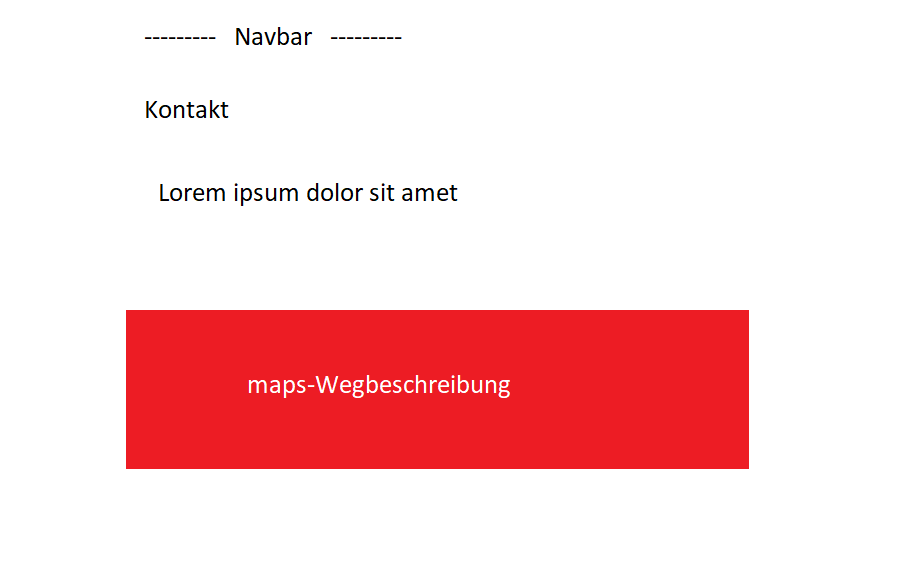
## Aussehen

Das Aussehen der Webseite haben wir gemeinsam definiert.

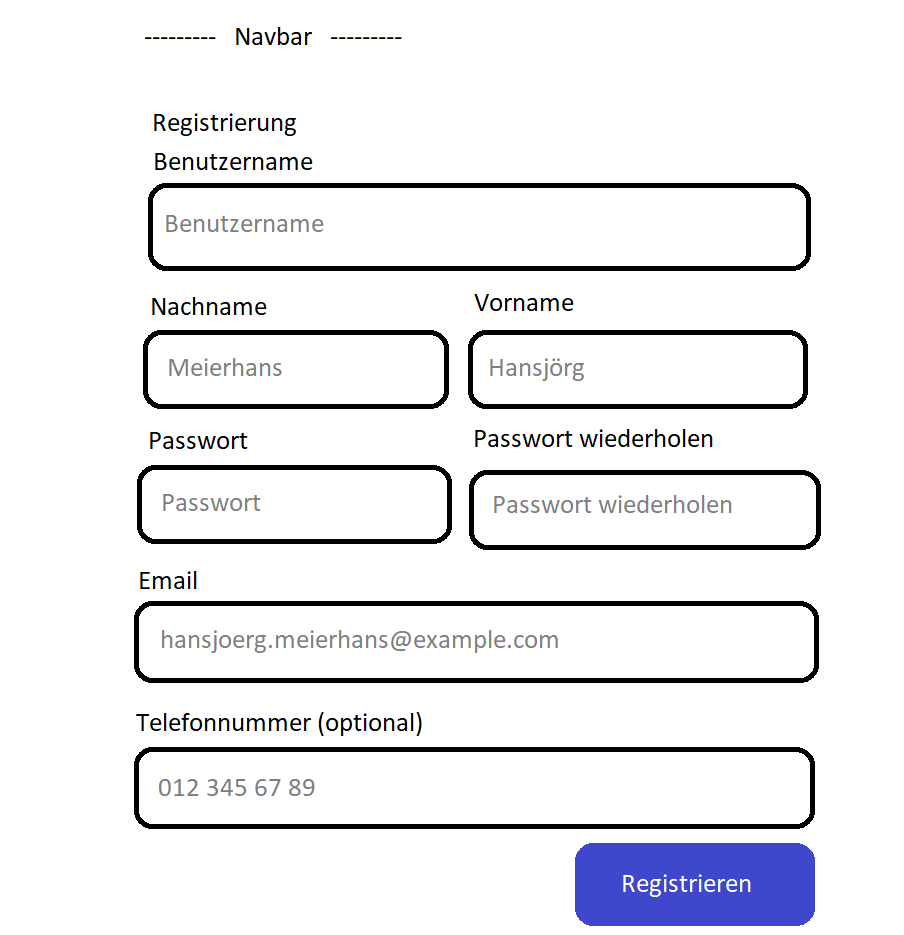
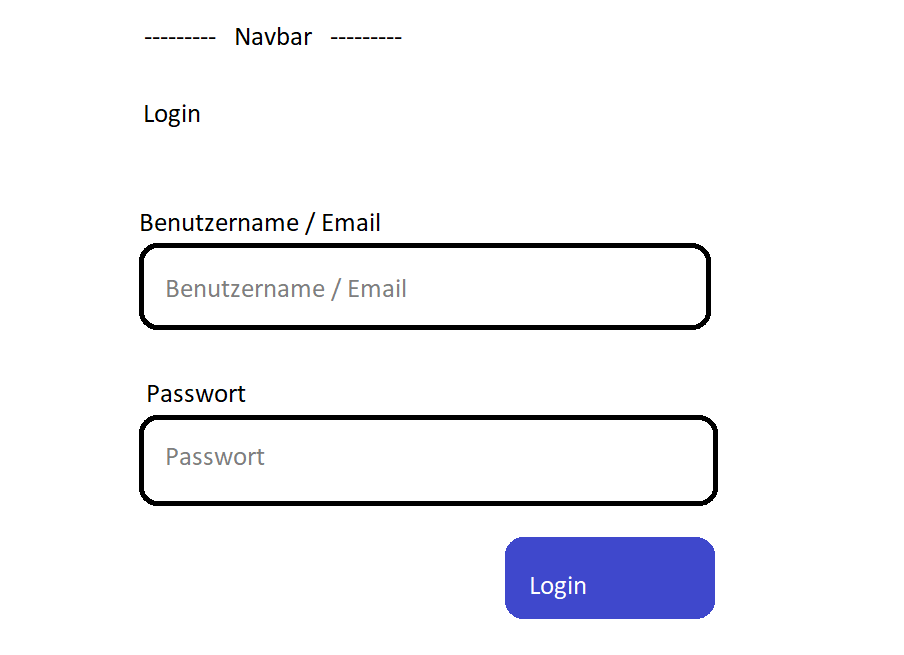
 

Seitennavigation Navbar (oben: nicht eingeloggt, unten: eingeloggt)

  Landing Page Dashboard (Auftragsanzeige)

Kontaktseite Anmeldung

Registrierung für Kunden und Mitarbeiter Login für Kunden und Mitarbeiter

# Entscheiden

## Datenbankdesign entscheiden

Wir haben uns für Runas Datenbankmodell entschieden, da dieses besser normalisierbar war und in der Umsetzung einfacher zu meistern ist.

## Aussehen entscheiden

Unsere Webseite soll einfach und modern aussehen, genau wie früher. Deswegen haben wir uns bei den neuen Eigenschaften für das gleiche Design entschieden.

## Lösungsweg entscheiden

Wir haben für den Lösungsweg entschieden der mehr Sinn für uns macht. Das wäre z.B. das wir zuerst die HTML Seite machen und dann die Logik dahinter. Als wir das hatten haben wir nach dem Prinzip «Trial-and-Error» gearbeitet.

# Realisieren

## SQL-Datenbank

In einem ersten Schritt wurde die SQL-Datei zum Erstellen der Datenbank so abgeändert, dass sie den Anforderungen aus dem ERM entspricht.

## Seiten erweitern

Die verschiedenen Seiten unserer Webseite wurden so abgeändert, dass sie dem neuen Design und den neuen Anforderungen (wie im Abschnitt Planen beschrieben) genügen.

# Kontrollieren

## Funktion überprüfen

Zunächst wurde überprüft, ob alles genau so funktioniert wie vorgesehen: ob alle Einträge und Abfragen mit der Datenbank die verlangten Ergebnisse haben und ob alle Seiten einander erreichen können, wenn dies vorgesehen ist. Hatten wir etwas vergessen einzubauen, wurde dies sofort nachgeholt.

## Anforderungen erfüllt?

Wenn alles funktionierte, haben wir geprüft, ob wir die Anforderungen aus der Aufgabenstellung erfüllen. Nichterfüllungen waren nachzuholen.

# Auswerten

## Realisierung

Niko: Tabellen erstellen mit eine Datenbankverbindung ist schwer, aber interessant. Ich soll im Internet nachschauen wie man andere Sachen machen können mit eine Datenbankverbindung

Runa: Es sind viele einfach vermeidbare Fehler passiert, die uns in der Behebung recht viel Zeit gekostet haben. Auch ein wichtiger Denkfehler hat uns recht viel Zeit gekostet: verschachtelte includes/requires gehen immer von der ursprünglichen Datei aus, nicht von der aktuellen.

## Planung

Wir haben von unserem Fehler von die letzte Partnerarbeit im M307 gelernt, dass wir zu wenig geplant haben. Deswegen haben wir diesmal mehr geplant und das hat sich gelohnt, weil wir sofort wussten was wir machen sollten.