



PROJEKTDOKUMENTATION



Nikolaos Papadopoulos, Runa Roth

Inhaltsverzeichnis

Informieren	3
Aufgabenstellung	3
Anforderungen	3
Informationsquellen	3
Benötigte Software	3
Planen	4
Zeitplanung.....	4
Vorschläge Datenbankdesign	5
ERM Runa	5
ERM Niko	6
Vorschläge Seitenaufbau.....	7
Seitenaufbau Runa	7
Seitenaufbau Niko.....	7
Vorschläge Seitengestaltung	8
Mockup Runa	8
Mockup Niko.....	10
Entscheiden	13
Datenbankdesign auswählen	13
Weitere Normalisierung	13
Relationenmodell	14
Seitenaufbau entscheiden.....	14
Seitengestaltung entscheiden	14
Realisieren.....	15
Datenbank umsetzen	15
Datenbankbenutzer	15
Views	15
Stored Procedures.....	15
Webseite gestalten	15
Model	15
Controller	16
Datenbankverbindung.....	16
Views.....	16
Kontrollieren.....	17
Funktionalität überprüfen	17
Anforderungen überprüfen.....	17

Funktionale Anforderungen	17
Zusätzliche Anforderungen.....	17
Auswerten	18
Fazit.....	18
Runa	18
Niko	18
Einführung des Systems	18
Voraussetzungen.....	18
Inbetriebnahme	18

Informieren

Aufgabenstellung

Unsere Aufgabenstellung ist die Webseite zu erweitern. Wir müssen: neue Dienstleistungen hinzufügen, Benutzer mit verschiedenen Berechtigungsstufen in der Applikation erstellen, Admins können Benutzerstammdaten und Mietartikeldaten zupflegen mit CRUD, Ski- und Snowboard Angeboten werden angezeigt mit Reservationsmöglichkeiten und noch die Verwaltung der Reservierungsaufträge (CRUD). Weil wir die Dienstleistungen von Ski- und Snowboardvermietung haben, werden wir neue Informationen von unseren Kunden das Geschlecht, Alter und Körpergröße in cm brauchen. Der Mietartikel soll inkl. Ein Foto, Kurzbeschreibung mit Preis in CHF haben. Eine Zusatz Aufgabenstellung wäre die Rechnung (PDF) einer Miete automatisch Generieren.

Anforderungen

Aus der Aufgabenstellung ergeben sich die folgenden Anforderungen:

- Unterscheidung zwischen Miete und Service
- Benutzer mit verschiedenen Rechten
- Mietobjekte und Services werden nicht statisch im PHP-Code, sondern dynamisch in der Datenbank gespeichert
- Admins können Benutzer und Service- bzw. Mietobjekte bearbeiten
- Testplan

Zusätzlich dazu haben wir von uns aus Anforderungen erstellt:

- Körperdaten auf Pro-Auftrag-Basis, nicht auf Pro-Benutzer-Basis
- Mehrere Mieten in einem Auftrag

Informationsquellen

Als Informationsquellen steht primär das Wissen aus diesem und den vorherigen Modulen zur Verfügung.

Falls dieses nicht ausreicht, ist für Fragen bezüglich Codestrukturierung und Methodenverwendung die offizielle PHP-Dokumentation sowie die HTML-, PHP- und SQL-Kategorien auf der Webseite «W3Schools» aufzusuchen.

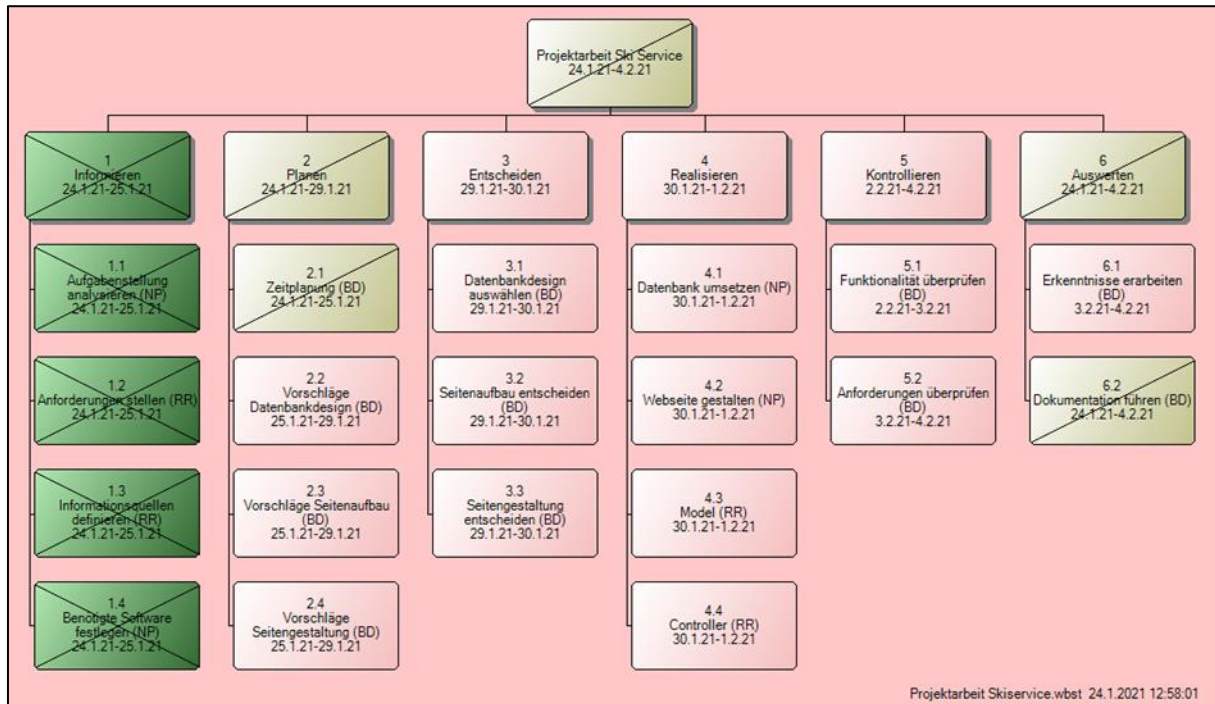
Wenn Hilfestellung bezüglich Codebeispielen nötig ist (Stichwort «Warum funktioniert dieser Codeabschnitt nicht?»), können auf StackOverflow Fragen gestellt werden. Diese Hilfestellung ist allerdings nur im Notfall beizuziehen.

Benötigte Software

Wir werden verschiedene Software benötigen. Wir brauchen Visual Studio Code für die Programmierung unserer Webseite. MySQL Workbench hilft uns unserer Datenbank einfach zu erstellen. Mit XAMPP können wir unsere Webseite auf Chrome öffnen und eine Verbindung zu unserer Datenbank machen. Einen Browser zum Testen brauchen wir auch.

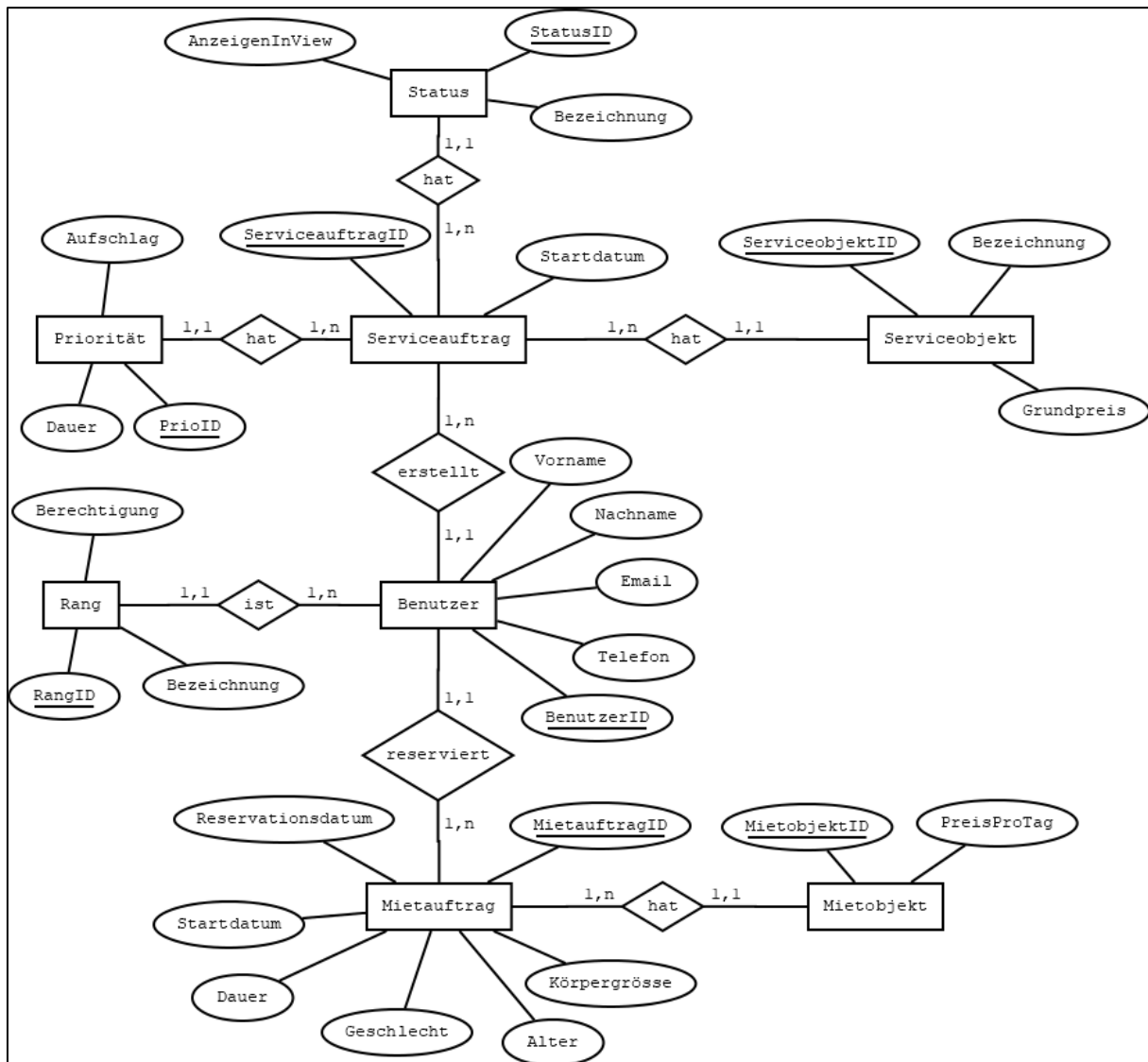
Planen

Zeitplanung



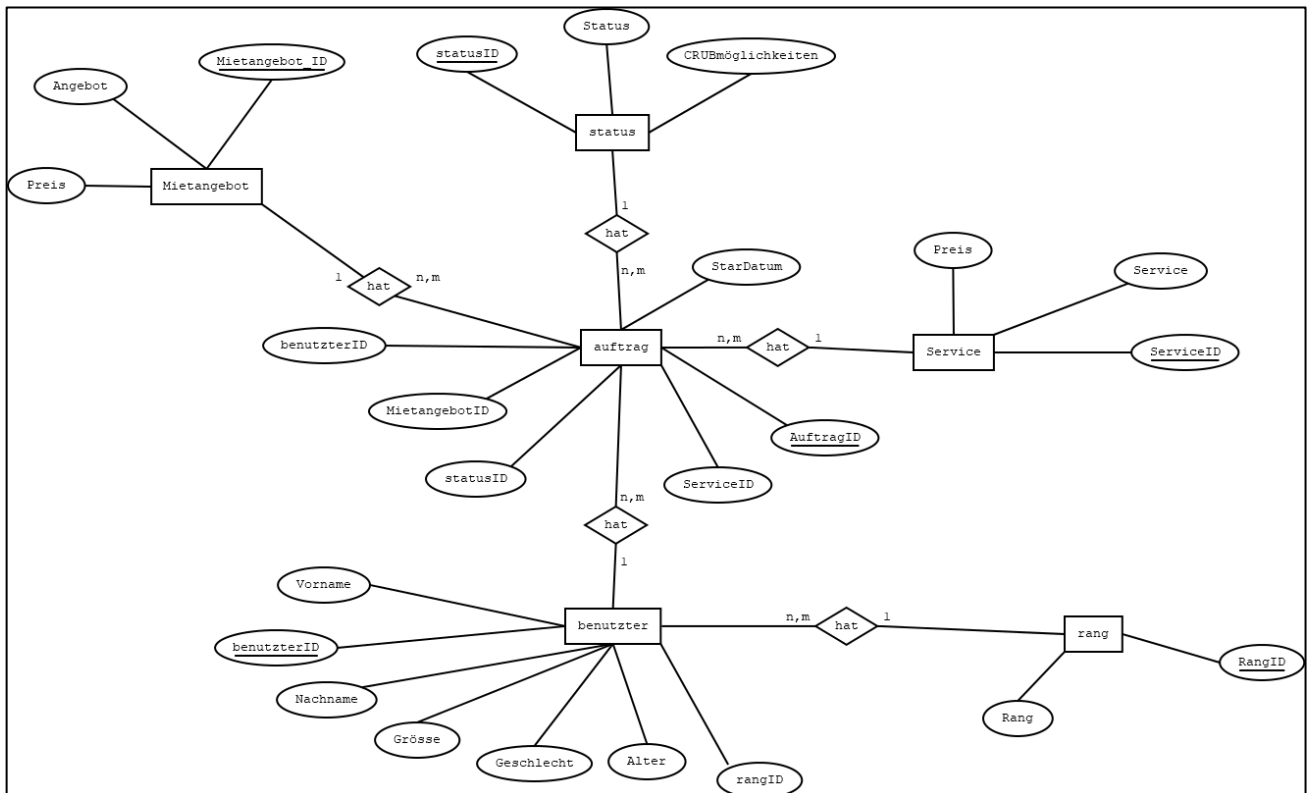
Vorschläge Datenbankdesign

ERM Runa

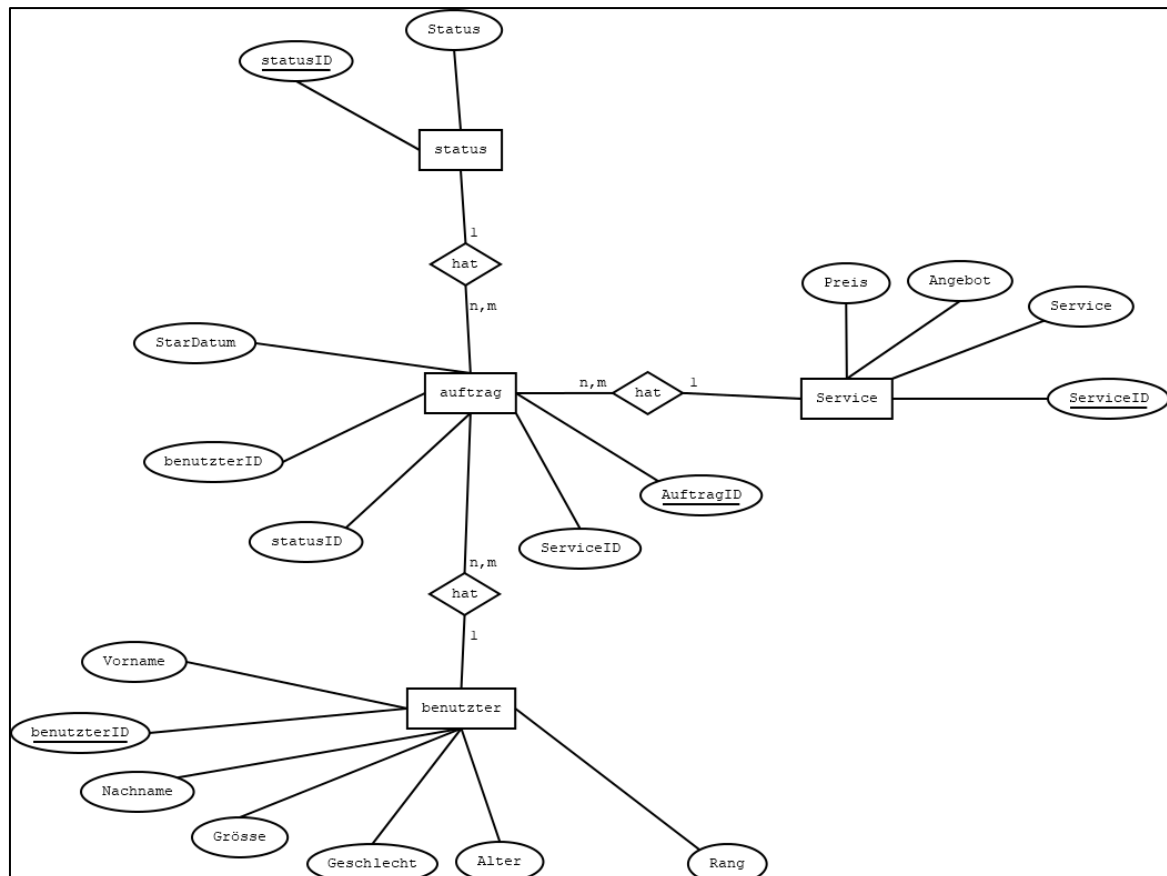


ERM Niko

V1

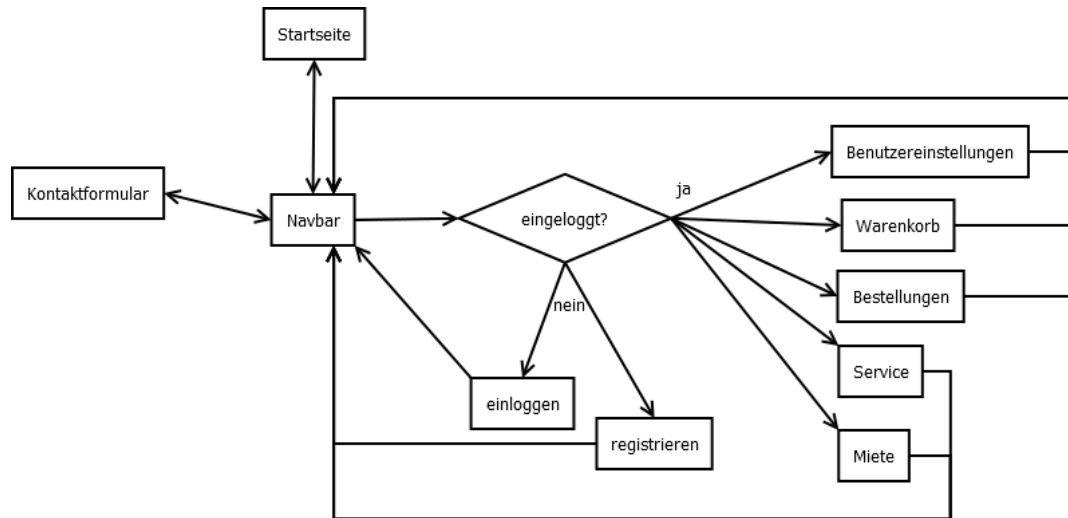


V2



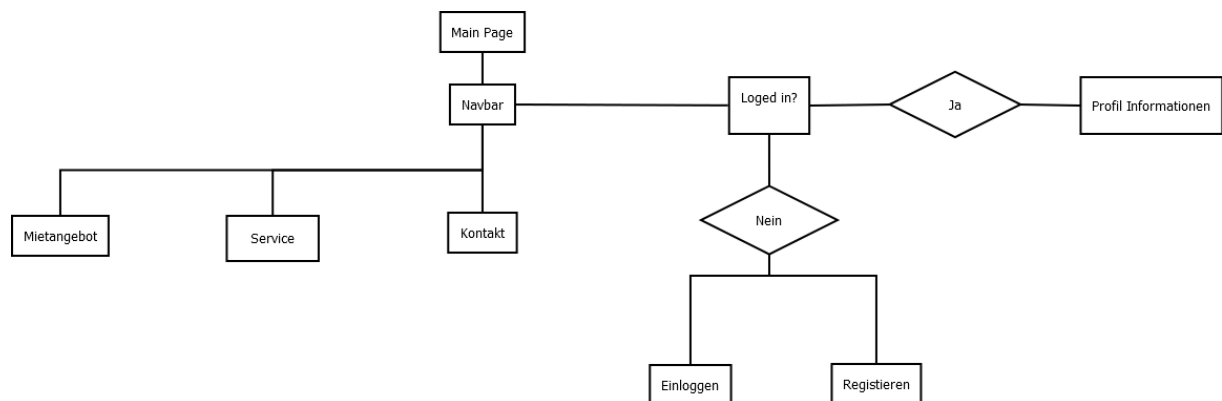
Vorschläge Seitenaufbau

Seitenaufbau Runa

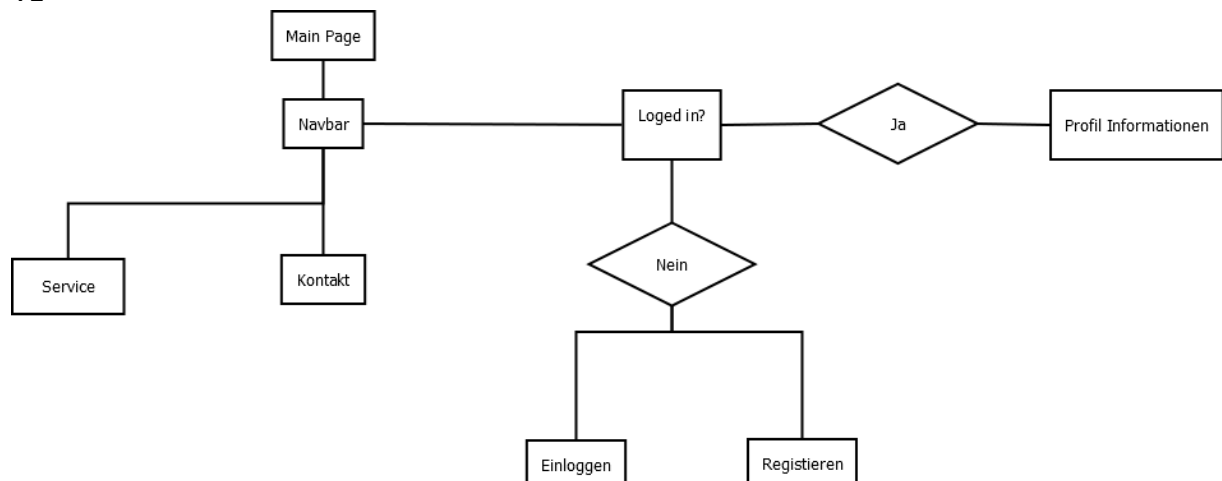


Seitenaufbau Niko

V1



V2



Vorschläge Seitengestaltung

Mockup Runa

Wenn abgemeldet:

[Logo] Startseite Kontakt Anmelden Registrieren

Wenn angemeldet:

[Logo] Startseite Anmelden ▼ Kontakt [Benutzername] ▼
 Service Warenkorb
 Miete Bestellungen
 Einstellungen

Kontakt

e-mail@provider.ch

Adresszeile 1
 Adresszeile 2
 Adresszeile 3

Telefonnummer

[Zum Kontaktformular](#)

Header mit Navigationsleiste

-- Navbar --

Willkommen beim Jetstream-Service!

Lorem ipsum dolor sit amet

Bild

Angebotsliste

Footer mit Kontaktdaten

-- Navbar --

Kontaktformular

Vorname Nachname

E-Mail

Nachricht

Absenden!

Startseite mit Bild als visuellem Aufhänger

-- Navbar --

Registrierung

Vorname Nachname

Email

Telefon (optional)

Passwort Passwort wiederholen

Registrieren!

Registrierungsformular

Kontaktformular

-- Navbar --

Login

E-Mail

Passwort

Einloggen!

Erneute Anmeldung

-- Navbar --

Service in Auftrag geben

Priorität

Servicetyp

Absenden!

Service in Auftrag geben

-- Navbar --

Warenkorb

Tabelle mit bestellten Inhalten

Warenkorbanzeige (ähnlich:
Bestellungsanzeige, Benutzerübersicht)

-- Navbar --

Service in Auftrag geben

Priorität

Servicetyp

Stornieren Absenden!

Wollen Sie wirklich stornieren?

Abbrechen Stornieren

Service bearbeiten

-- Navbar --

Schneesportgeräte mieten

Körpergröße

Geschlecht

Altersgruppe

Startdatum Enddatum

Reservieren!

Mietformular

-- Navbar --

Einstellungen

Vorsicht! Sie bearbeiten gerade ihr Profil! Gegebenenfalls können Sie sich nach Bestätigung der Änderungen nicht mehr anmelden!

Vorname Nachname

Email

Telefon (optional)

Neues Passwort Neues Passwort bestätigen

Altes Passwort erneut eingeben

Profil löschen Änderungen bestätigen

Benutzereinstellungen

-- Navbar --

Schneesportgeräte mieten

Körpergröße

Geschlecht

Altersgruppe

Startdatum Enddatum

Stornieren Absenden!

Wollen Sie wirklich stornieren?

Abbrechen Stornieren

Miete bearbeiten

Mockup Niko

Startseite	Service	Mietangebot	Kontakt		Einloggen	Registrieren
------------	---------	-------------	---------	--	-----------	--------------

Titel

background image
with transbox

Main Page mit Navbar. Ein Hintergrundbild mit einer Trans Box, wo unser Text sein wird. Wenn Mietangebot in Service Formular sein wird, dann wird Mietangebot von die Navbar entfernt.

Benutzername	
Vorname	Nachname
Email	Telefon
Geschlecht	Grösse
Passwort	Passwort wiederholen
Enter	

Registrieren Formular.

Priorität

▲▼

Mietangebot

▲▼

Service

▲▼

Enter

Priorität

▲▼

Service

▲▼

Enter

Service Formular V1, mit Mietangebot.

Angebot

▲▼

Datum

Enter

Service Formular V2, ohne Mietangebot.

Navbar

Warenkorb

Liste
Löschen
Löschen
Löschen

Alles Löschen

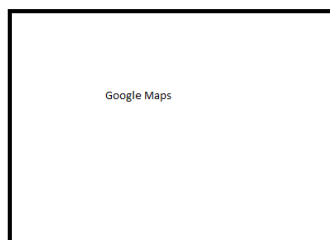
Kaufen/Senden

Mietangebot Formular.

Warenkorb von Service oder Mietangebot.



Einloggen mit schon erstelltem User.



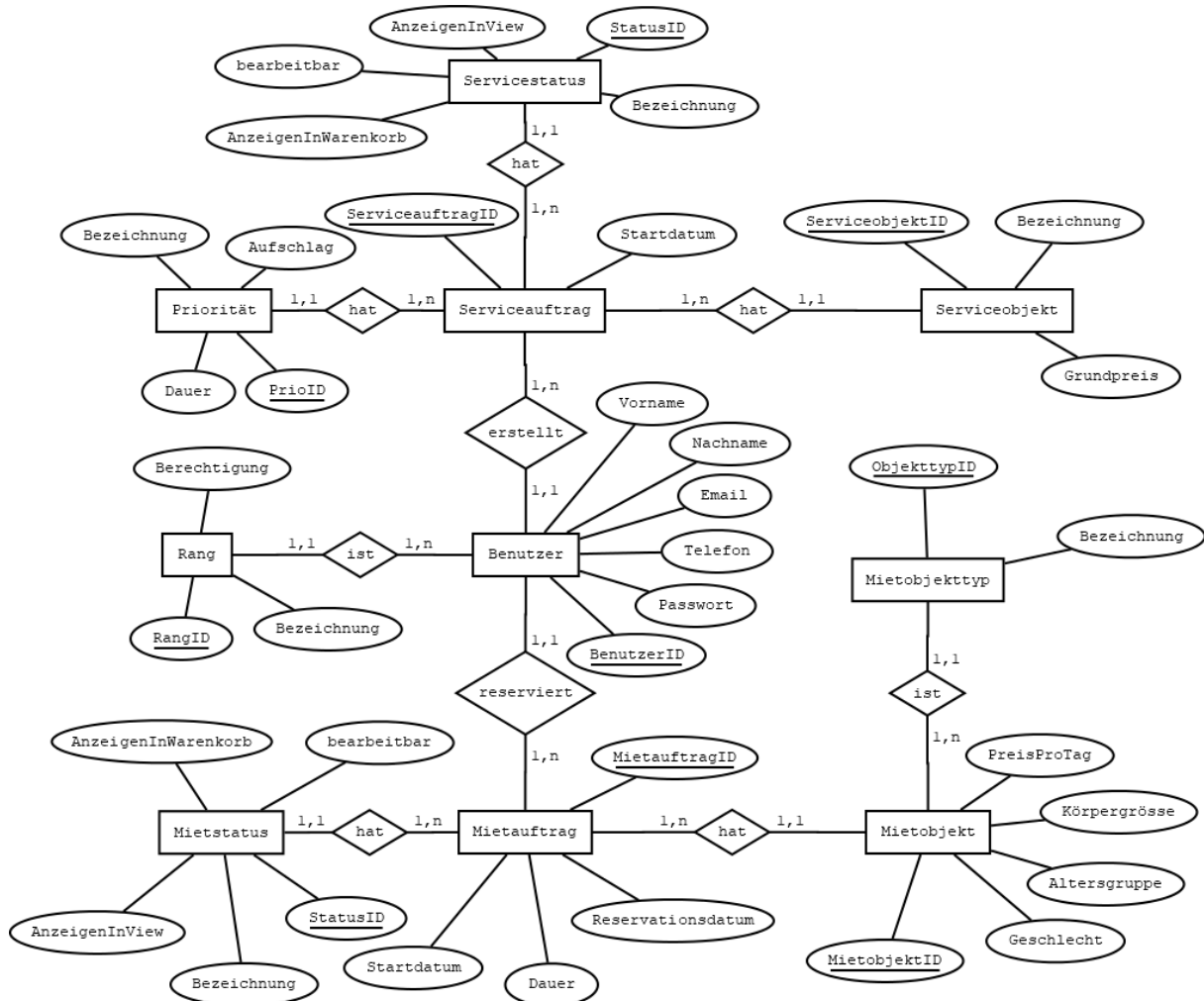
Info Seite mit Google Maps.

Entscheiden

Datenbankdesign auswählen

Weitere Normalisierung

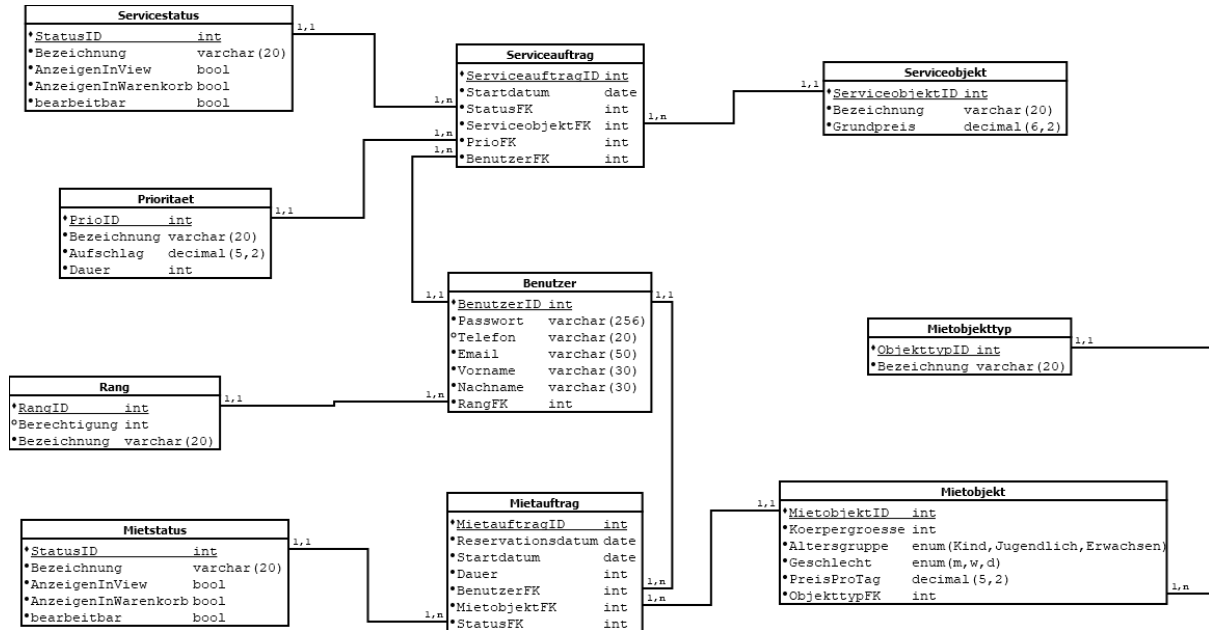
Wir haben uns für Runas ER Datenbankmodell entschieden. Wir haben es auch weiterentwickelt. Neu sieht es so aus:



Grund dafür wäre das Runas ER besser normalisiert ist und weil es flexibler ist für arbeiten.

// besser formulieren

Relationenmodell



Seitenaufbau entscheiden

Da es detaillierter und genauer ist, haben wir uns für Runas Aufbaudiagramm entschieden. Die drei Diagramme sind nahezu identisch, was die Entscheidung schwer gemacht hat.

Da wir zufrieden sind mit ihnen, müssen wir sie nicht weiter verbessern.

Wir haben uns für ein MVC Design Pattern entschieden, da dieses zum einen in der Aufgabenstellung verlangt wird, andererseits war beim letzten Mal noch Verbesserungspotenzial, also müssen wir üben, um es zu können.

Seitengestaltung entscheiden

Runa:	Navbar	Bedingte Anzeige
	Footer	Bei Niko nicht vorhanden
	Registrierung	Besser an ERM angepasst
	Mietformular	Besser an ERM angepasst
	Bearbeitungsformulare	Bei Niko nicht vorhanden
Niko:	Startseite	Visuell ansprechender
	Warenkorb	Detaillierter, Leeren/absenden Funktion
Kombination:	Kontakt/Info	
Identisch:	Login	
	Service (bei Niko V2)	

Realisieren

Datenbank umsetzen

Datenbankbenutzer

Es gibt drei Benutzer in der Datenbank: Kunde, Mitarbeiter und Admin.

Sie haben unterschiedliche Privilegien: Der Kunde darf nur die Inhalte der Kunden- und Auftragstabellen mutieren, den Rest nur anschauen. Mitarbeiter dürfen zusätzlich Service- und Mietobjekte bearbeiten, hinzufügen und löschen. Admins dürfen alles.

Views

Views helfen uns grosse Datenbank befehle zu kürzen. So können wir unser PHP code besser lesbar machen. Es gibt verschiedene Views. Views wo man z.B. denn Benutzer mit seine Mietauftrag zeigt.

Alle Views:

- Alle Benützer mit Mieatauftrag
- Alle Benutzer mit Serviceauftrag
- Alle Mietauftraege
- Alle Services
- Alle Mietobjekte
- Benutzer und Berechtigung
- Benutzer und Rang

Stored Procedures

Stored Procedures sind wie Methoden beim PHP. Man kann beim einen Procedure z.B. ein Insert command hinzufügen. Wir haben zwei Wichtige Procedures programmiert

- Createuser
 - Beim Registrieren wird ein neuer Benutzter hinzugefügt
- PMietauftrag
 - Hinzufügt eine neues Mietauftrag

Webseite gestalten

Model

Für jede Tabelle und jedes View in der Datenbank wurde eine Model-Klasse mit nur Feldern erstellt. Die Attribute widerspiegeln genau die der Tabelle bzw. des Views, die Klasse beinhaltet keine Logik.

Diese Klasse wird verwendet, um die Resultate einer PDO-Abfrage nicht in ein Array, sondern in ein Objekt zu speichern. Dies macht den Code absturzsicherer und stabiler.

Controller

Alle Controller sind in demselben Ordner. Dies hat den Vorteil, dass alle Views, Models etc. mit einer Datei abgerufen werden können. Diese heisst `requireAll.inc.php`.

Die Controller sind die Pages, die vom Benutzer aufgerufen werden. Über die oben genannte Datei werden alle Views und Models eingebettet und können bei Bedarf jederzeit aufgerufen werden.

Jede Controller-Datei besteht aus drei Teilen:

1. Kopf: Hier wird die Session gestartet und alle Dateien bereitgestellt.
2. Logik: Hier findet alle Logik statt, die gebraucht wird. Es werden Eingaben überprüft, Weiterleitungen gesetzt, Datenbankbefehle ausgeführt, etc.
3. Aufruf der Views: Standardmässig wird zuerst der Seitenkopf (Titel, CSS, etc.) aufgerufen. Darauf folgt die Navigationsleiste (Navbar). Dann kommt der eigentliche Seiteninhalt, anschliessend der Footer mit Kontaktdaten sowie der Seitenfuss.

Ausnahme davon ist die Logout-Seite, die bloss die Session zerstört und sofort auf die Startseite weiterleitet. Bei dieser fehlt Teil 3.

Datenbankverbindung

Für die Datenbankverbindung wurde eine separate Klasse erstellt, in der die Zugangsdaten zur Datenbank abgespeichert sind. Falls diese nicht mit dem Datenbankserver übereinstimmen, kann der Admin diese einfach ändern.

Die Benutzernamen und Passwörter sind je in einem Array gespeichert und werden über einen optionalen Parameter in der Verbindungsfunktion ausgewählt. Dies soll von der Webseite genutzt werden: Je nach Rang des aktuellen Benutzers wird die Datenbankverbindung mit einem anderen Datenbankbenutzer aufgebaut.

Die Funktion «connect» wird aufgerufen, um eine Datenbankverbindung aufzubauen. Sie gibt ein neues PDO-Objekt zurück, das für Datenbankabfragen verwendet werden kann.

Views

Bei den Views werden die HTML Files geschrieben, die den Visuellen Teil der Webseite zeigt. Wir wollen das unsere Design Standard ist und gleichzeitig schön/cool. Deswegen haben wir uns entschieden, dass der Standard aussehen der Webseite so sein wird: Ein Hintergrundbild, dass das Thema von unserer Webseite ausspricht. Ein transparentes Texthintergrund, dass den Text mehr lesbar macht. Wir werden auch eine Fusszeile haben, mit die Kontaktinformationen von der Firma zeigt. Views werden mit der Hilfe von Bootstrap geschrieben, aber man wird auch unser eigenes CSS benutzen.

Kontrollieren

Funktionalität überprüfen

Dies haben wir während des Entwickelns laufend gemacht. Alle Daten einmal so falsch wie möglich eingeben und überprüfen ob alle Fehlermeldungen angezeigt werden. Dann einmal alles richtig eintragen. Wenn bei letzterem alles richtig ist und die Webseite die grüne Meldung anzeigt, muss in der Datenbank überprüft werden, ob alles so in der Datenbank vorhanden ist, wie verlangt.

Falls etwas falsch lief in dem Prozess, wird entweder die typische orangene XDebug Fehlermeldung angezeigt. Falls nicht, muss per `var_dump` die Fehlermeldung vom PDO Statement ausgegeben werden.

Anforderungen überprüfen

Funktionale Anforderungen

1. Login mit Registrierungsmöglichkeit Erfüllt
2. Login Dialog mit Passwort für autorisierten Zugang der Mitarbeiter mit Berechtigung
In Nr. 1 integriert, erfüllt
3. Anzeige der Mietartikel inkl. Foto und Kurzbeschreibung und Preis
Erfüllt
4. Auswahlmöglichkeiten eines Mietartikels mit Menge und Reservationsdatum bzw. Dauer
Erfüllt
5. Anzeige der gewählten Artikel im Warenkorb und Speicherung bei Mietabschluss
Erfüllt
6. Anzeige der eingegangenen Mietbestellungen für die internen Mitarbeiter
Erfüllt
7. Berücksichtigung sämtlicher sicherheitsrelevanten Aspekte
Erfüllt
8. Datenbankstruktur 3. Normalform Erfüllt
9. Eigener Datenbankbenutzerzugang Erfüllt: 3 Benutzer: Kunde, Mitarbeiter, Admin
10. Datenbank-Views, Procedures und Transaktionen bei Datenmutation
Erfüllt
11. MVC Design Pattern Erfüllt
12. Dokumentation nach IPERKA in Arbeit

Zusätzliche Anforderungen

1. Annullierung eines Mietauftrages durch Mitarbeiter
Erfüllt
2. Annullierung eines Mietauftrages durch Kunde
Erfüllt
3. Automatische PDF-Rechnung Nicht geplant
4. Statusverwaltung der Mietaufträge Erfüllt
5. Mehrere Artikel pro Bestellung Nicht geplant
6. Warenkorb kann bearbeitet werden Erfüllt

Auswerten

Fazit

Runa

Zeitplanung ist verdammt schwer und zu viele Überstunden tun nicht gut.

Das Projekt war spannend und hat Spass gemacht. Die Aufgabenteilung war sinnvoll.

Dadurch, dass wir komplett von vorne angefangen haben, aber das Serviceangebot weiter beibehalten wollten, hat sich die Arbeit stark erhöht in allen Bereichen.

Niko

Ich habe bemerkt das MySQL Programmieren interessanter geworden ist als früher. Beim HTML programmieren auch.

Das Projekt war sehr interessant und hat Spass gemacht. Ich muss mich aber verbessern beim HTML

Ich habe vieles neues gelernt und das finde ich sehr gut.

Einführung des Systems

Hier folgt eine Kurzanleitung, wie das System in Betrieb genommen werden kann.

Voraussetzungen

Für die Inbetriebnahme werden ein Datenbankserver sowie ein HTTP-Server benötigt. Ein intelligenter Texteditor wie Notepad++ oder Visual Studio Code ist von Vorteil, aber nicht zwingend.

Inbetriebnahme

Für die Inbetriebnahme müssen folgende Schritte ausgeführt werden:

- Inhalt des Ordners "APP" in das Verzeichnis, auf das der Webserver zugreift, entpacken. Ein Unterordner dessen ist auch möglich.
- Mit einem Benutzer, der Datenbanken und Tabellen erstellen und Benutzer bearbeiten kann, die 5 SQL-Skripte im Ordner "APP/database" **in der folgenden Reihenfolge** ausführen:
 1. Ski_service.sql
 2. insert_Mietobjekte.sql
 3. Ski_service_Views.sql
 4. Ski_service_stored_Procedures.sql
 5. Ski_service_User_Access.sql
- Die Datei "APP/repository/Database.class.php" folgendermassen bearbeiten:
 1. Zeile 3: Den Wert hinter "\$host" auf die IP-Adresse des Datenbankservers setzen (localhost lassen, wenn die Server auf derselben Maschine laufen, bspw. über XAMPP)
 2. Zeile 5: Den Wert hinter "\$port" auf den Port des Datenbankservers setzen
- Die Datei "APP/repository/GeneralVariables.class.php" folgendermassen bearbeiten:
 1. Zeile 3: Den Wert hinter "\$MainPageName" auf Ihren Firmennamen setzen

Nun ist das System einsatzbereit!