

Projekt: “KlarschiffSN”



Bearbeiter: Jonas, Oliver, Oleksandra

Kunde: T.Schott

Inhaltsverzeichnis

1. Projektbeschreibung
2. Vorgehensmodell
3. Anforderungsspezifikationen
4. Architekturmuster
5. Wirtschaftlichkeit beurteilen
6. Bewertungsmatrix zu Frameworks
7. Bewertungsmatrix
8. Usability
9. Anwendungsfalldiagramm
10. Wireframe

1. Projektbeschreibung

Die Aufgabe dieses Projekts ist die Entwicklung einer Webanwendung. Die Anwendung soll Informationen über die Stadt Schwerin sammeln und in einer Datenbank speichern. Die Daten werden über eine API-Schnittstelle von verschiedenen Quellen bezogen.

Die gesammelten Daten sollen dann in der Datenbank aggregiert und als Statistik für die Kunden der Anwendung ausgegeben werden. Die Kunden können auf diese Weise Einblicke in verschiedene Aspekte der Stadt Schwerin gewinnen, wie z.B. Straßenschäden, Kriminalitätsraten, Umweltverschmutzung, etc. Die Statistiken sollen in Form von Diagrammen und Tabellen präsentiert werden, damit die Kunden die Daten leicht interpretieren und vergleichen können.

Die Webanwendung soll mit modernen Technologien wie HTML und JavaScript entwickelt werden. Die Datenbank wird mit MySQL implementiert. Die Anwendung soll responsive und benutzerfreundlich sein und auf verschiedenen Geräten laufen.

Das **Ziel** des Projekts ist es, eine innovative Webanwendung zu entwickeln, die es Kunden ermöglicht, fundierte Entscheidungen über verschiedene Aspekte der Stadt Schwerin zu treffen. Die Anwendung wird den Kunden einen wertvollen Einblick in die Daten geben und ihnen helfen, besser informierte Entscheidungen zu treffen.

2. Vorgehensmodell

Das Ziel von Kanban ist es, die Effizienz und Produktivität zu steigern, indem man den Fluss der Arbeit verbessert, Engpässe identifiziert und Abweichungen schnell erkennt und korrigiert.

Im Kern basiert Kanban auf einem visuellen Board, das in Spalten und Zeilen unterteilt ist. Jede Spalte repräsentiert einen Schritt im Arbeitsprozess, während jede Zeile eine Aufgabe oder ein Arbeitspaket darstellt. Jede Aufgabe wird auf einer Karte notiert, die auch als "Kanban-Karte" bezeichnet wird. Diese Karte enthält Informationen wie den Titel der Aufgabe, eine Beschreibung, den Zuständigen und den Status.

Indem die Kanban-Karten von einer Spalte zur nächsten bewegt werden, können Teammitglieder den Fortschritt der Arbeit auf einen Blick verfolgen. Kanban erlaubt es, den Workflow ständig anzupassen und zu optimieren, indem man auf Engpässe und Verzögerungen reagiert und sie ausmerzt.

Kanban ist sehr flexibel und kann auf verschiedene Arten und in verschiedenen Umgebungen angepasst werden. Es kann für die Verwaltung von Software-Entwicklungsprojekten, Marketing-Kampagnen, Wartungsaufgaben und vielen anderen Arten von Aufgaben genutzt werden.

3. Anforderungsspezifikationen

funktionale Anforderungen:

Es soll eine Webseite erstellt werden, die die Statistik von Klarschiff Schwerin darstellt und in verschiedenen Diagrammen grafisch aufbereitet. Mindestens drei Diagramme sollen dabei zum Einsatz kommen.

Die Statistiken sollen nach Typen sortiert und gefiltert werden.

Der Benutzer soll in der Lage sein, die Daten nach Typen und Services auszuwählen und zu sortieren, die er benötigt.

Admins können die Datenbank aktualisieren.

Webanwendung, soll allgemein für alle Browser gestaltet werden. Jedoch soll es kein extra Layout für mobile Geräte geben.

nicht funktionale Anforderungen:

Das Design und die Farben der Statistik sollen dem auf der Startseite von Schwerin bs-technik-schwerin.de gleichen.

Zudem soll eine Admin-Seite erstellt werden, auf der man sich anmelden kann.

Sehr Zuverlässig und sehr geringe Fehlertoleranz, da bereits auf "geprüfte Meldungen" eingegangen wird.

3D Animationen werden nicht benötigt. Das Design soll schlicht und minimalistisch sein.

Die Webanwendung sollte robust sein und soll auch einen starken Andrang von Usern aushalten.

IT-Sicherheit: Die Software blockiert nach mehreren Fehlversuchen die Anmeldung für 30 Sekunden.

Datenmodellierung:

Design der Datenbank und Definition der Entitäten, Attribute und Beziehungen, die für die Speicherung der Daten benötigt werden.

Web Anwendungsentwicklung:

Implementierung der Benutzeroberfläche der Webanwendung, die die Daten aus der Datenbank abrufen und als Statistik präsentiert.

Testen und Qualitätssicherung:

Durchführung von Unit- und Integrationstests, um sicherzustellen, dass die Anwendung einwandfrei funktioniert und den Anforderungen entspricht.

4. Architekturmuster

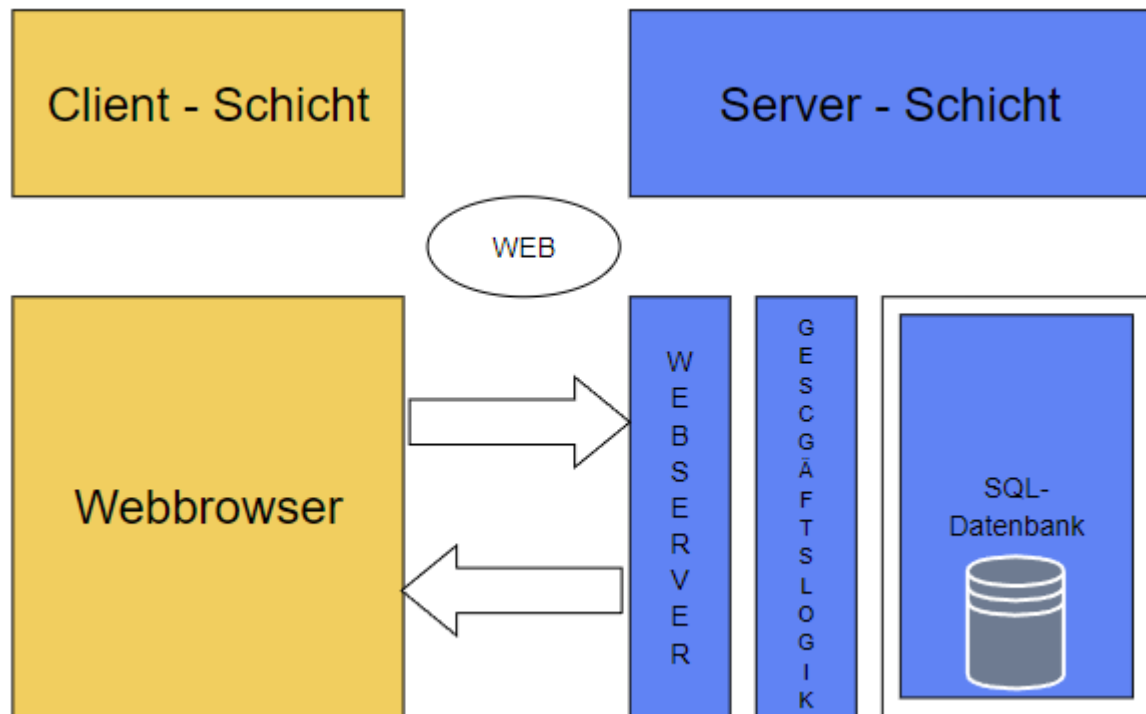
Die Server-Client-Architektur ist ein klassisches Konzept in der Informatik, das sich auf die Struktur von Netzwerkanwendungen bezieht. Diese Architektur besteht aus zwei einfachen Komponenten: dem Server und dem Client, welche in der Regel auf verschiedenen Rechnern laufen. Beispielsweise wird ein Client-Server-System mit einem zentralen Datenbankserver betrachtet. Das Client-Server-System bildet eine Netzwerkstruktur, bestehend aus dem zentralen Datenbankserver als Server-Komponente und mehreren Benutzer-Clients als Client-Komponenten. Der Client bildet das Anwendungsprogramm, über dessen Benutzerschnittstelle der Benutzer über das Netzwerk auf Ressourcen des Datenbankservers zugreift. Der Server stellt hierfür verschiedene Dienste zur Verfügung, die dann von unterschiedlichen Clients genutzt werden können.

Client-Schicht: Template View HTML/CSS, JavaScript

In der Präsentationsschicht werden für den Benutzer die verschiedenen Statistiken als Diagramme dargestellt. Zudem ist es hier möglich, die Daten gefiltert darzustellen.

Server-Schicht

Mit der Server-Schicht werden die angeforderten Statistiken für den Server bereitgestellt. Zusammen bilden der Data-Manager und der Controller eine wichtige Schicht in der Architektur der Anwendung. Der Data-Manager ist für die Verwaltung von Datenbankverbindungen und Transaktionen verantwortlich und bietet eine Schnittstelle zum Abrufen und Manipulieren von Daten. Der Controller ist für die Implementierung der Geschäftslogik verantwortlich und koordiniert die Datenbankoperationen mit Hilfe des Data-Managers. Durch die Trennung der Geschäftslogik von der Datenbankverarbeitung kann die Anwendung flexibler und leichter wartbar sein.



5. Wirtschaftlichkeit beurteilen:

Kostenberechnung: Bei einem Stundenlohn von 25 € und einem Zeitrahmen von 40 Stunden betragen die Personalkosten für das Projekt 1.000€. Dazu kommt noch ein Gewinnzuschlag von 20 % und 19 % Umsatzsteuer Zuschlag. Insgesamt entstehen dadurch Kosten in Höhe von 1.428€. Dadurch haben wir Kosten in Höhe von 1000€ und einen Ertrag von 1428. Dadurch entsteht ein Gewinn von 200€. Dadurch entsteht eine Umsatzrentabilität von 14%. Diese wird wie folgt berechnet: $\text{Umsatzrentabilität} = (\text{Gewinn}/\text{Umsatz}) \cdot 100$.

Kosten-Nutzen-Verhältnis (K/N-Verhältnis): Das K/N-Verhältnis gibt an, wie viel Nutzen man für jede ausgegebene Geldeinheit erhält. In diesem Fall haben wir keine genauen Zahlen für den Nutzen, aber wenn wir davon ausgehen, dass die neuen Funktionen des Programms dazu beitragen, die Arbeit effizienter zu gestalten und möglicherweise Zeit und Ressourcen zu sparen, kann man davon ausgehen, dass der Nutzen höher ist als die Kosten. Unter der Annahme, dass der Nutzen etwa 14 % höher ist als die Kosten, beträgt das K/N-Verhältnis etwa 1,14. (Kosten < Nutzen). Dadurch dass die Umsatzrentabilität bei 14% liegt, kann man ebenfalls erkennen, dass der Nutzen für uns größer ist als die Kosten.

Zusammenfassend lässt sich sagen, dass das Projekt wirtschaftlich sinnvoll ist, da es wahrscheinlich einen Gewinn für uns ergibt. Dies kann man aus den beiden Prognosen ziehen.

6. Bewertungsmatrix

Es gibt mehrere Gründe, warum Express.js für das Projekt eingesetzt werden soll: von schnelleren I/Os für schnellere Anfragen und Antworten bis hin zu seinem Single-Thread-System und asynchronen Prozessen. Außerdem nutzt es die MVC-Struktur, um Datenmanipulationen und Routing-Systeme zu vereinfachen.

Express.js ist sehr benutzerfreundlich und flexibel und schneller als jedes andere Node.js-Framework. Es ist ein minimalistisches Framework, das eine schnelle Anwendungsentwicklung ermöglicht und viele verschiedene Teile größerer Frameworks beherrscht. Express.js hat sich im Laufe der Jahre als sehr skalierbar erwiesen, da viele große Unternehmen das Framework täglich auf ihren Servern einsetzen.

| | Express.js | ASP.Net | AngularJS | React |
|------------------|------------|---------|-----------|-------|
| Lizenzkosten | 5 | 5 | 5 | 5 |
| Einfache Nutzung | 4 | 4 | 1 | 4 |
| Flexibilität | 5 | 3 | 4 | 3 |
| Dokumentation | 5 | 5 | 5 | 4 |
| Erweiterbarkeit | 5 | 4 | 5 | 3 |
| Kompatibilität | 5 | 4 | 5 | 4 |
| Gesamtpunktzahl | 29 | 25 | 25 | 23 |

7. Risikoanalyse

Es gibt verschiedene Arten von Risiken, welche bei der Durchführung eines Softwareprojektes bestehen. In der nachfolgenden Tabelle werden einige die häufigsten Risiken aufgeführt, die ein Softwareprojekt betreffen können.

| | Eintrittswahrscheinlichkeit | | |
|--------------------|---|---|----------------------------------|
| | Gering | Mittel | Groß |
| Personelle Risiken | Das Projektteam harmonisiert nicht miteinander. | Mitarbeiter werden längere Zeit krank oder fallen ganz aus. | |
| Zeitliche Risiken | Keine oder unrealistische | | Verzögerung in den Projektseiten |

| | | | |
|--------------------------|--|--|---|
| | Terminpläne und Kostenplanungen. | | fertigstellen. |
| Technologische Risiken | Es werden neue Systemarchitekturen eingesetzt. | Neue Entwicklungsumgebung, mit der die Mitarbeiter noch nicht vertraut sind. | |
| Organisatorische Risiken | Mitarbeiter sind in mehreren Projekten aktiv. | | Unzureichende Kommunikation innerhalb des Projektteams. |

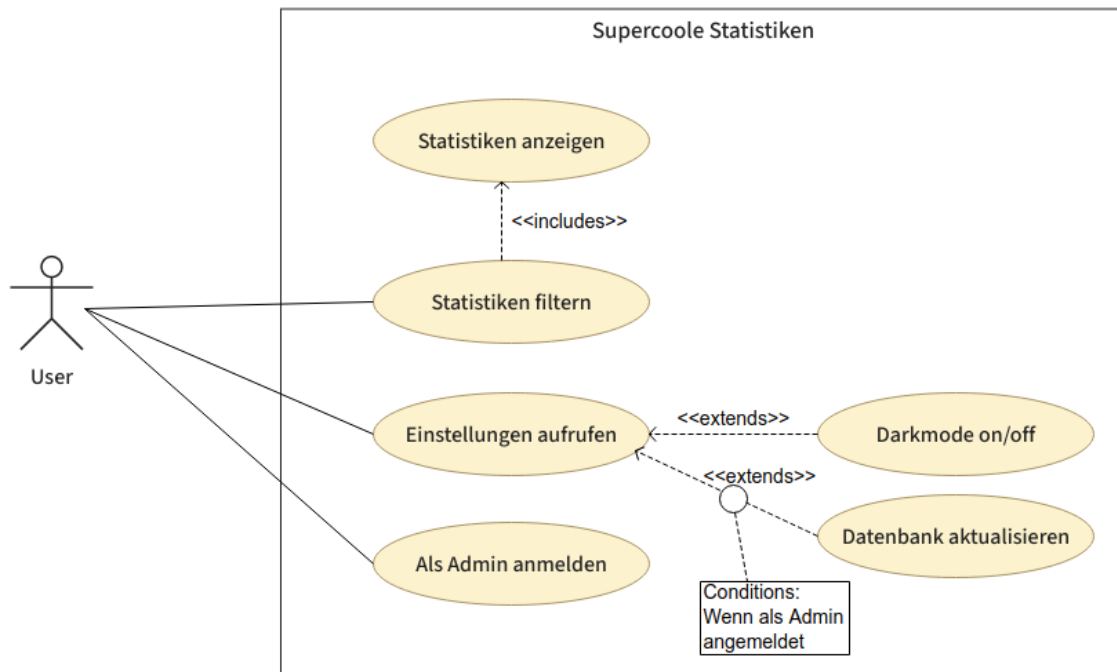
8. Usability

Usability oder Benutzerfreundlichkeit ist ein wichtiger Aspekt bei der Entwicklung von Produkten. Eine gute Usability gewährleistet, dass Benutzer ein Produkt einfach und effektiv nutzen können. In diesem Abschnitt werde ich die Usability-Aspekte unseres Projekts erläutern.

1. Wir haben die Zielgruppen unseres Projekts analysiert, um die Bedürfnisse und Erwartungen besser zu verstehen. Auf diese Weise können wir unser Design und unsere Funktionalitäten auf ihre Bedürfnisse und Anforderungen zuschneiden.
2. Wir werden darauf achten, dass unser Design konsistent ist, um Verwirrung bei den Benutzern zu vermeiden. Wir werden einheitliche Farben, Schriftarten und Symbole verwenden, um die Bedienung unseres Produkts oder Dienstleistung zu erleichtern.
3. Wir werden auch darauf achten, dass unser Design barrierefrei ist und allen Benutzern eine einfache Bedienung ermöglicht. Wir werden beispielsweise eine ausreichend große Schriftart verwenden, um eine gute Lesbarkeit zu gewährleisten, und werden alternative Texte für Bilder und Grafiken zur Verfügung stellen, um sie auch für sehbehinderte Benutzer zugänglich zu machen.

9. Anwendungsfalldiagramm

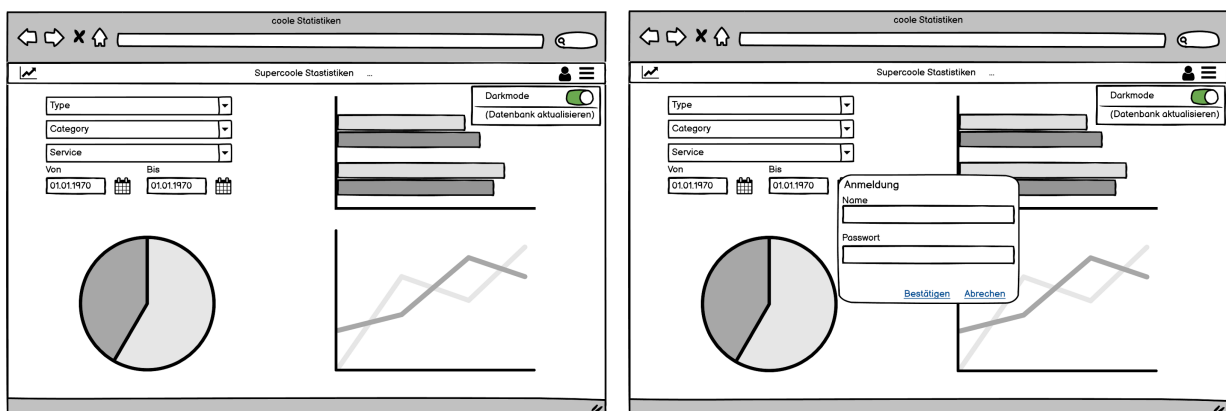
Ein Anwendungsfalldiagramm ist ein Diagramm, das die Interaktion zwischen Benutzern und einem System modelliert. Es zeigt die verschiedenen Anwendungsfälle oder Aufgaben, die von Benutzern ausgeführt werden können, sowie die Interaktionen zwischen Benutzern und dem System. Wir haben dieses Diagramm verwendet, um die Funktionalität des Systems zu modellieren und zu kommunizieren.



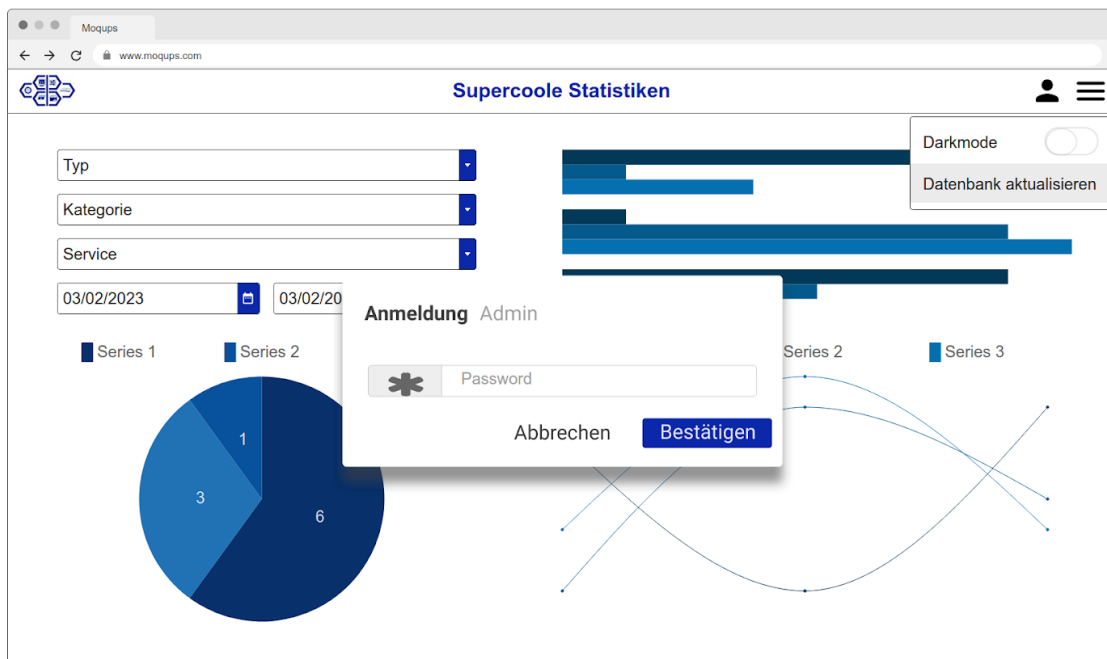
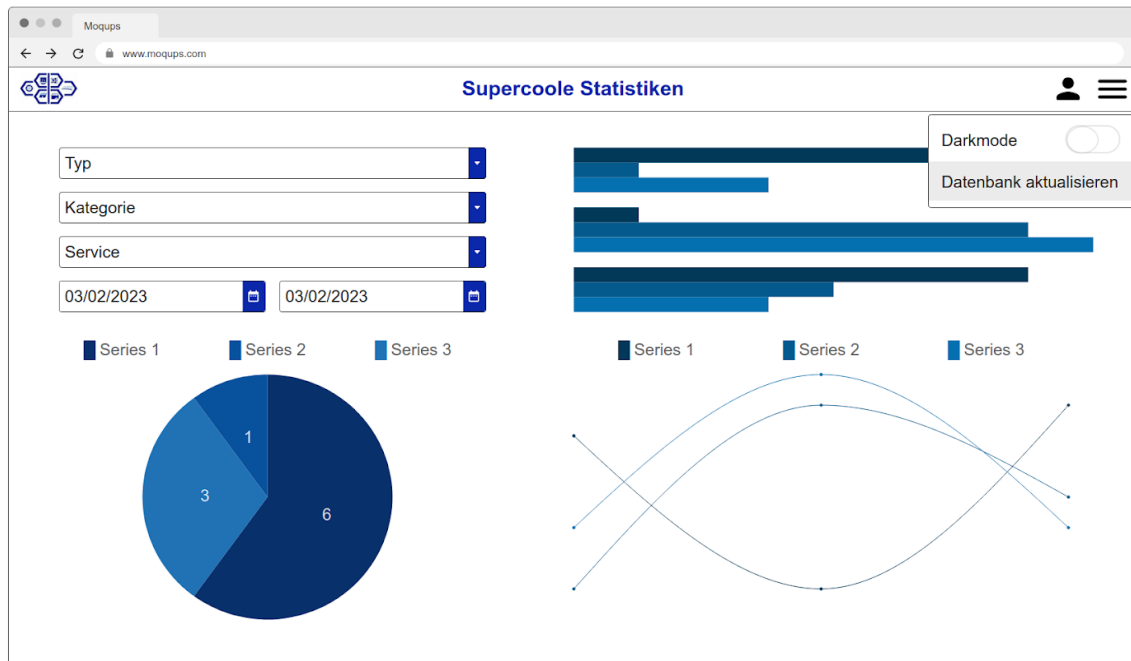
10. Wireframe und Moqups

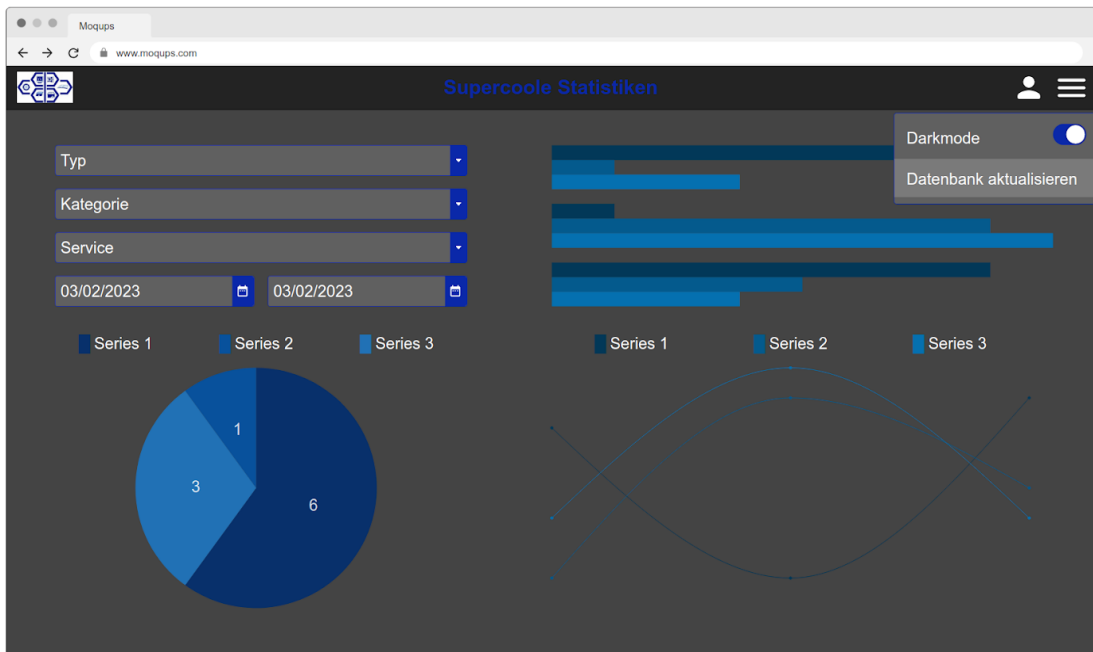
Beim Abrufen der Webseite von Klarschiff SN werden erstmal die Statistiken mit Standard-Werten an dem Diagramm dargestellt. Möchte man Statistik in der verschiedenen Kategorien , Typen oder Services sehen, gibt es die Möglichkeit mit Select Box und dazu noch die Zeitraum auszuwählen. Oben rechts in Einstellungen man darf Darkmodus aktivieren (unabhängig davon ob man sich angemeldet) und sich mit Benutzernamen/Password einloggen um zu Datenbank aktualisieren.

Wireframe:



Mockup:





Moqups
www.moqups.com

Supercoole Statistiken

Darkmode ☒

Datenbank aktualisieren

Typ
Kategorie
Service
03/02/2023 03/02/2023

Series 1 Series 2 Series 3

| Series | Value |
|----------|-------|
| Series 1 | 1 |
| Series 2 | 3 |
| Series 3 | 6 |

| Series | Value |
|----------|-------|
| Series 1 | 1 |
| Series 2 | 3 |
| Series 3 | 6 |

| Series | Value |
|----------|-------|
| Series 1 | 1 |
| Series 2 | 3 |
| Series 3 | 6 |

Anmeldung Admin

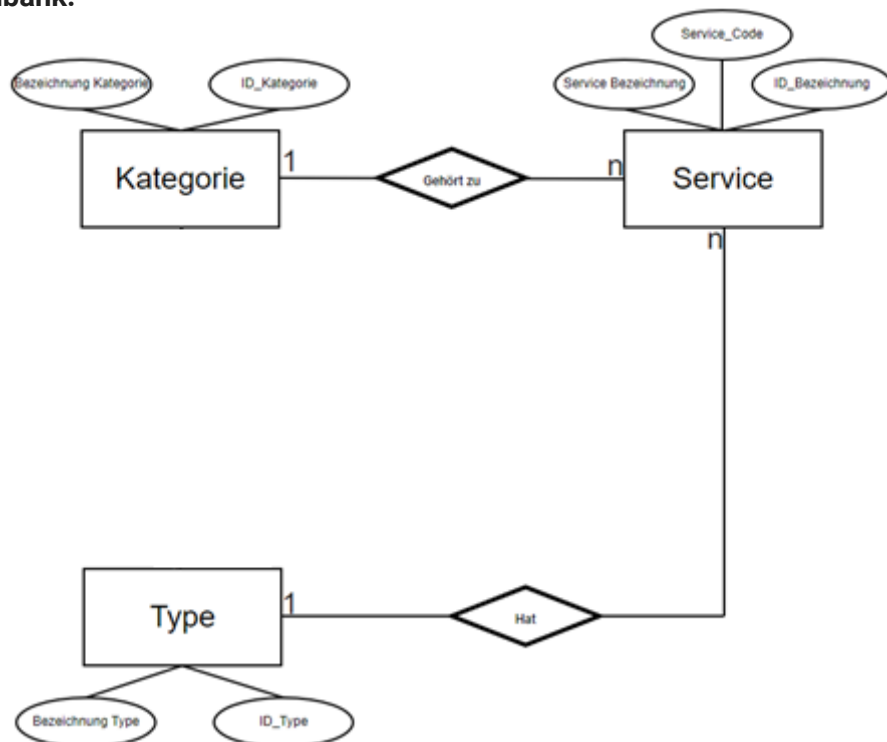
Password

Abbrechen Bestätigen

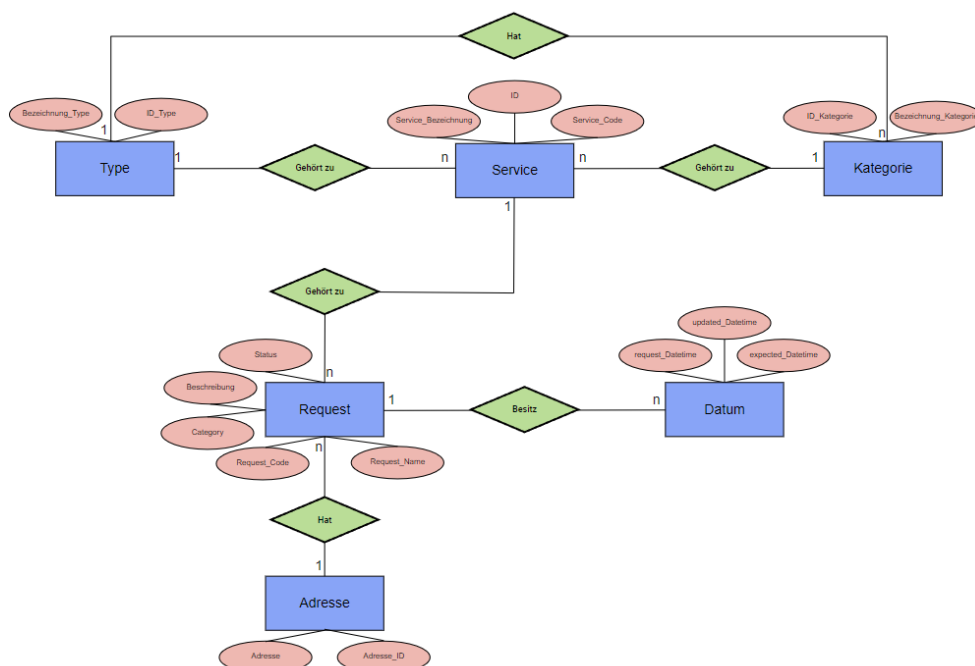
11. Datenbankmodellierung und ERM Modell für API

Unter Datenmodellierung versteht man das Strukturieren und Ordnen einer Menge von Daten, damit diese von einer Datenbank korrekt und ohne großen Aufwand verarbeitet werden können. Im zweiten Bild sieht man das ERM Modell für die API. So könnte sie z.B. Aussehen. Natürlich wäre es auch möglich, dass das ERM Modell anders aussehen kann, da dies nur eine mögliche Version ist.

ERM Datenbank:

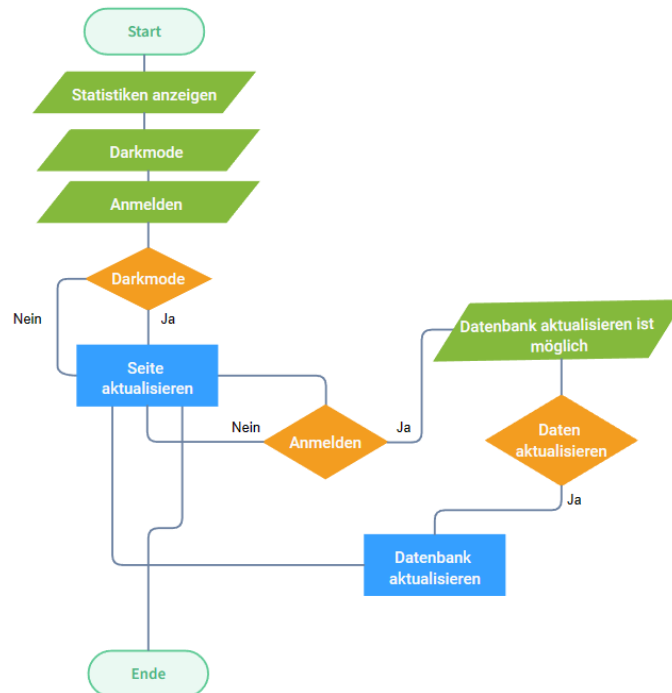


ERM API:



12. Programmablaufplan

Für die Modellierung von Algorithmen können verschiedene Modellierungssprachen verwendet werden. In der Regel werden dafür entweder ein PAP, ein Struktogramm, ein Aktivitätsdiagramm oder auch Pseudocode eingesetzt. Wir haben uns für ein PAP entschieden, um uns einen Überblick zu verschaffen.



13. Fazit und Ausblick

In der vorliegenden Projektarbeit wurde das Thema anhand der Entwicklung einer Web-Anwendung Klarschiff Schwerin untersucht und durchgeführt. Dabei wurden folgende Phasen untersucht: Projektziel, Planung des Projekts und Entwicklungsphase.

Das Ziel des Projektes wurde erreicht und von den Kunden abgenommen. Die offene Frage wurde geklärt und ein paar möglichen Stichpunkten für die Weiterentwicklung erstellt.

14. Eigenständigkeitserklärung

Hiermit erklären wir, dass wir die vorliegende Arbeit eigenständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie die aus fremden Quellen direkt oder indirekt übernommenen Stellen/Gedanken als solche kenntlich gemacht haben.