



Abiturjahrgang 2023

Seminararbeit

Leitfach: Informatik

Rahmenthema des Seminars: Robotik

Thema der Arbeit:

Smarter Wecker

Verfasser: Bernd Storath

Kursleiter: StD Christian Joachim

Abgabetermin: 08. November 2022

Bewertung	Note	Notenstufe in Worten	Punkte		Punkte
Schriftliche Arbeit				x 3	
Abschlusspräsentation				x 1	
Summe:					
Gesamtleistung nach § 29 (7) GSO = Summe : 2 (gerundet)					

Unterschrift Kursleiter

eingegangen am

Inhaltsverzeichnis

1 Einleitung	3
2 Aufbau	4
2.1 Kosten der eingesetzten Komponenten	4
2.2 Relation der Komponenten	5
3 Funktionen	5
4 Quelloffenheit	6
5 Einstellungen	6
6 Frontend	7
7 Backend	8
8 Android App	9
9 Kommunikation	9
9.1 Kommunikationsaufbau	9
9.2 Android App mit Backend	11
9.3 Frontend mit Backend	11
9 Design	12
10 Right to Repair	12
11 Fazit und Ausblick	13
12 Anhang	14
13 Abkürzungsverzeichnis	14
14 Literaturverzeichnis	14
14.1 Internetadressen	14
14.2 Bildquellen	16
15 Eidesstattliche Erklärung	17

1 Einleitung

Ich habe bereits viele Wecker kaufen müssen, da bisher alle aufgrund schlecht verklebter Teile im Inneren beim Bewegen geraschelt haben oder wegen ihres leichten Gewichts umgefallen und zu Bruch gegangen sind. Außerdem ist das Einstellen des Weckers durch die geringe Anzahl an Knöpfen zeitaufwendig und ist zudem meist nicht auf jeden Tag individuell anpassbar. Eine Alternative zum klassischen Wecker ist das Smartphone, welches mittlerweile die meisten Menschen unter 34 nutzen¹, jedoch lädt dieses dazu ein noch kurz vor dem Schlaf seine Nachrichten zu lesen, wodurch wegen des ausgestrahlten Blaulichts der Schlaf geschädigt wird².

Es sind schon Produkte vorgestellt worden, die dieses Problem beheben sollen, eines davon ist das „Google Nest Hub“, welches als „[m]ehr als ein fancy Wecker“³ beschrieben wird. Doch auch diese Lösung hat einige Nachteile, wie zum Beispiel ein Sicherheits- bzw. Privatsphärenrisiko durch die direkte Anbindung an Google LLC⁴. Auch die Anpassungs- und Erweiterungsmöglichkeiten sind nicht so stark vertreten wie die des im Folgenden vorgestellten Smarten Weckers⁵. Weiter sind die meisten Wecker aus Kostengründen aus Plastik⁶, das allerdings sehr schlecht zersetzbar ist und daher eine Gefahr für die Umwelt darstellt⁷. Eine gute Alternative wäre ein Wecker aus Holz, der zum einen besser für die Umwelt ist⁸ und zum anderen auch noch „[...] ein Stück Natur nach Hause [bringt]“⁹.

¹ Schmidt, M., Wenn der Wecker zweimal klingelt, in <https://yougov.de/news/2016/05/28/wenn-der-wecker-zweimal-klingelt/>, zuletzt aufgerufen am 30.08.2022.

² „3 Gründe, das Handy nicht als Wecker zu nutzen“, in <https://www.ongnamo-versand.de/blog/3-gruende-das-handy-nicht-als-wecker-zu-nutzen/>, zuletzt aufgerufen am 30.08.2022.

³ Hasse, K., „Mehr als ein fancy Wecker: Der Google Nest Hub im Test“, in: <https://www.annabelle.ch/stil/mehr-als-ein-fancy-wecker-der-google-nest-hub-im-test/>, zuletzt aufgerufen am 30.08.2022.

⁴ „Can Your Google Home Or Google Nest Be Hacked? Here's How“, in: <https://robotpowered-home.com/can-a-google-home-be-hacked/>, zuletzt aufgerufen am 22.03.2022.

⁵ „Nest Hub Max - home screen customization“, in: <https://www.googlenestcommunity.com/t5/Speakers-and-Displays/Nest-Hub-Max-home-screen-customization/td-p/83948>, zuletzt aufgerufen am 22.03.2022.

⁶ „Plastik ist zu billig: Behörde warnt vor Recycling-Problemen“, in: <https://www.rnd.de/wirtschaft/plastik-ist-zu-billig-behorde-warnt-vor-recycling-problemen-CFNJXXBVM7HHZOLK5ME-CACDXQM.html>, zuletzt aufgerufen am 30.08.2022.

⁷ „Plastikmüll“, in: <https://de.statista.com/themen/4645/plastikmuell/#dossierKeyfigures>, zuletzt aufgerufen am 30.08.2022.

⁸ Durant, T., „Ist Holz nachhaltiger als Plastik?“, in: <https://citizensustainable.com/de/holz-nachhaltiger-plastik/>, zuletzt aufgerufen am 30.08.2022.

⁹ Rüsche, A., „Umwelt schonen und trotzdem schöner Wohnen mit Holz“, in: <https://www.bauen-wohnen-aktuell.de/wohnen/umwelt-schonen-und-trotzdem-schoener-wohnen-mit-holz/>, zuletzt aufgerufen am 30.08.2022.

2 Aufbau

2.1 Kosten der eingesetzten Komponenten

Um alles zu steuern, wird als Herzstück der Raspberry Pi 4 benutzt, welcher im Zentrum steht und den Bildschirm, den Lautsprecher und die Anbindung zur App steuert. Er wird verwendet, da er ein vielseitig einsetzbarer, kleiner und billiger Einplatinencomputer mit Anschaffungskosten von 89,28€¹⁰ ist.

Angezeigt wird der Wecker wird auf einem 7-Zoll 480p Monitor mit Touch Funktionen. Durch die Touch Funktion kann der Nutzer den Wecker interaktiv steuern und die Auflösung von 800 auf 480 Pixeln sorgt für die gute Erkennbarkeit des Inhalts. Dieses Display steht für umgerechnet 48,95€ zum Verkauf¹¹.

Um den Klingelton abzuspielen und den Nutzer aufzuwecken, wird ein 1,3-Zoll 4 Ohm Lautsprecher der Firma Visaton verwendet. Man kann einen solchen für 9,05€ kaufen¹²

Da der Lautsprecher mit der nötigen Spannung versorgt werden muss wird ein Verstärker verwendet, der für 4,90€ käuflich ist¹³.

Alle aufgelisteten Komponenten sind in einem aus Holz gefertigten Gehäuse (s. Abb. 9) mit den Maßen Länge 29 cm x Breite 11 cm x Höhe 16 cm eingebaut und sind so genug geschützt, um lange ihre Funktion zu erhalten. Da das Holz nicht speziell für die Herstellung gekauft werden musste, kann der Preis vernachlässigt werden.

Die Materialkosten für die Herstellung eines solchen Smarten Weckers belaufen sich somit auf 152,18€.

¹⁰ „Raspberry Pi 4 Modell B 4GB SDRAM“ in: <https://www.rasppishop.de/Raspberry-Pi-4-Modell-B-4GB-SDRAM?src=raspberrypi>, zuletzt aufgerufen am 30.08.2022.

¹¹ „7inch Capacitive Touch Screen LCD (B), 800×480, HDMI, Low Power“, in: <https://www.waveshare.com/7inch-hdmi-lcd-b.htm>, zuletzt aufgerufen am 30.08.2022.

¹² „VIS 2243 Breitbandlautsprecher BF 32, 2 W, 4 Ohm“, in: <https://www.reichelt.de/de/de/breitbandlautsprecher-bf-32-2-w-4-ohm-vis-2243-p276918.html?r=1>, zuletzt aufgerufen am 30.08.2022.

¹³ „DEBO SOUND AMP1 Entwicklerboards - Audioverstärker, Klasse D, PAM8302A“, in: <https://www.reichelt.de/entwicklerboards-audioverstaerker-klasse-d-pam8302a-debo-sound-amp1-p235505.html>, zuletzt aufgerufen am 30.08.2022.

2.2 Relation der Komponenten

Der Smarte Wecker ist in drei logische Einheiten gegliedert (s. Abb. 1). Das Frontend leitet alles, was den Nutzer direkt betrifft, also Bild und Ton. Das Backend verwaltet die Datenbank und die Kommunikation bzw. Synchronisation zwischen App und Frontend. Die App steuert die Anpassung, das heißt sie passt die Weckzeiten an, lädt Klingeltöne hoch und ändert den Ort, von welchem das Wetter gezeigt werden soll.

Der Lautsprecher ist durch 2 Kabel an den Verstärker verlötet. Der Verstärker wiederum ist an die 2 Kontakte für den Klinke Ausgang des Raspberry Pi verlötet

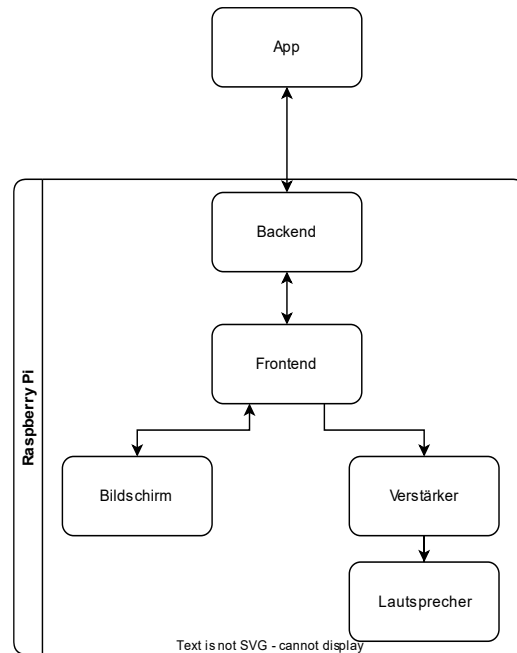


Abb. 1: Interaktion zwischen den Komponenten, eigene Abbildung vom 30.08.2022.

3 Funktionen

Der Smarte Wecker hat Funktionen, wie Uhrzeit, Wetter, Weckfunktion und eigene Klingeltöne, um sich von anderen Produkten abzuheben. Diese sind vom Nutzer nach Belieben anpassbar. Die Benutzeroberfläche auf dem Wecker wird in zwei Module aufgeteilt: Wecker und Wetter.

Das Weckermodul (s. Abb. 2) zeigt die Uhrzeit, das Datum und das Wetter für die nächsten 7 Tage mithilfe von Symbolen an. Drückt man auf eines dieser Symbole wird man zu einem genaueren Wetterbericht weitergeleitet.



Abb. 2: Wecker Modul, eigene Abbildung, vom 16.10.2022.

Das in Abbildung 3 gezeigte Wettermodul zeigt das jetzige Wetter mit Bild, Höchst- und Tiefsttemperatur und Information zu Wind, Regen und Sonnenaufgang bzw. Sonnenuntergang. Darunter ist eine stündliche Auflistung der Wettervorhersagen für die nächsten 24 Stunden. Am Ende sieht man das Wetter für die nächsten 7 Tage. Wenn man auf eines der 7-tägigen Vorhersagen drückt, wird der Temperaturverlauf über den Tag angezeigt.

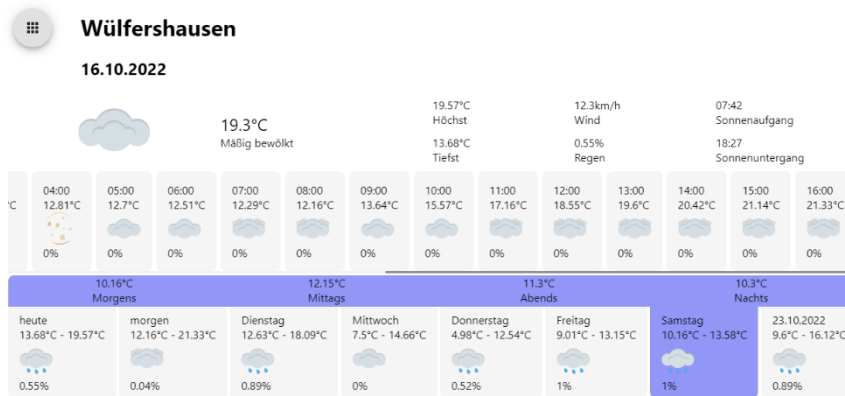


Abb. 3: Wetter Modul, eigene Abbildung vom 16.10.2022.

4 Quelloffenheit

Da das Projekt in jeglicher Hinsicht samt Quellcode frei verfügbar ist¹⁴, kann es auch in Zukunft durch den modularen Aufbau erweitert werden. So ist eine Anpassung weitab von den vorgegebenen Modulen möglich. Ein grundlegender Vorteil einer solchen Open-Source-Software ist das „peer-review“-Prinzip¹⁵. Bei diesem wird der Quellcode durch sogenannte Peers untersucht und anschließend bewertet, ob dieser Schwachstellen oder unsichere Muster verfolgt. Somit können Schwachstellen schneller entdeckt und das Produkt sicherer gemacht werden, wie Nowak zeigt: „For example, bug fixes have been shown to come out about 6 times faster for OSS than for proprietary equivalents“¹⁶. Außerdem ist man für eine fortlaufende Ausstattung des Produkts mit Sicherheitsupdates nicht auf eine Firma oder Betreiber angewiesen, da es andere Programmierer oder auch man selbst machen kann.

5 Einstellungen

Da die jeweiligen Nutzer den Smarten Wecker auf sich zuschneiden wollen, sind verschiedene Einstellungen verfügbar

Eine wichtige Eigenschaft des Weckers ist das er um die richtige Uhrzeit weckt. Da der Nutzer aber in verschiedenen Zeitzonen lebt oder reist, muss diese einstellbar

¹⁴ Storath, B., „Smart Alarm Clock“, in: <https://github.com/kaaax0815/smart-alarm-clock>, zuletzt aufgerufen am 30.08.2022.

¹⁵ Hapke, T., „Was ist eigentlich Peer Review?“, in: <https://www.tub.tuhh.de/wissenschaftliches-arbeiten/2018/02/14/was-ist-eigentlich-peer-review/>, zuletzt aufgerufen am 30.09.2022.

¹⁶ Nowak, P., „The Powerful Economic Underpinnings of OSS“, in: <https://web.archive.org/web/20031204125550/http://www.linuxworld.com/story/34293.htm>, zuletzt aufgerufen am 30.09.2022.

sein. Daher kann man durch eine sogenannte „Dropdown-Liste“¹⁷ die richtige Zeitzone auswählen.

Die im Wecker- und Wettermodul angezeigten Werte sind auch vom Nutzer anpassbar. So kann dieser den Ort eingeben in dem er sich gerade befindet und die Werte für den jeweiligen Ort werden angezeigt.

6 Frontend

Um ein umfangreiches und komplexes Frontend zu erstellen, wird React von Meta Platforms, Inc. verwendet, da dieses durch die vorhandene Flexibilität alle Arten von Anwendungen erstellen lässt. React wurde allerdings für die Entwicklung von Webanwendungen im Browser angefertigt, weswegen zusätzlich auch Electron genutzt wird, um eine minimale Browserumgebung auf Linux zu erstellen.

Die im Wettermodul angezeigten Werte werden durch die One Call API 2.5 von OpenWeather¹⁸ bereitgestellt. Die kostenlose API kann eine Millionen Mal im Monat, aber nur maximal 60-mal in der Minute, aufgerufen werden. Da die Werte aber nur alle 5 Minuten im Frontend aktualisiert werden, wird der maximale Wert nicht erreicht. Dadurch muss kein kostenpflichtiges Abonnement abgeschlossen werden, was sich negativ auf den Nutzer auswirken würde.

Um Werte für den Ort zu erhalten an dem der Nutzer wohnt, kann dieser in der App die Stadt angeben. Da die oben genannte API aber Längen- und Breitengrad benötigt, wird hierfür die Geocoding API von OpenWeather¹⁹ genutzt. Durch diese wird der angegebene Ort in Koordinaten umgewandelt, welche dann in der Datenbank (s. Punkt 7) gespeichert werden, um unnötige API-Aufrufe zu vermeiden.

¹⁷ „Dropdown-Liste“, in: <https://de.wikipedia.org/wiki/Dropdown-Liste>, zuletzt aufgerufen am 01.11.2022.

¹⁸ „One Call API 2.5“, in: <https://openweathermap.org/api/one-call-api>, zuletzt aufgerufen am 01.11.2022.

¹⁹ „Geocoding API“, in: <https://openweathermap.org/api/geocoding-api>, zuletzt aufgerufen am 01.11.2022.

OpenWeather hat zwischen Mai 2020 und Juni 2020 eine Statistik veröffentlicht, welche ihre künstliche Intelligenz, welche durch globale Wetterdienste und von Nutzern bereitgestellten Wetterstationen eine Vorhersage berechnet, mit anderen Quellen vergleicht. Das Ergebnis war, dass die Reliabilität ungefähr 95 % beträgt, und die Wetterwerte die angezeigt werden somit aussagekräftig sind

7 Backend

Für das Backend wird mit Express.js ein schneller und responsiver Server erstellt, der als Anwendungsschnittstelle (API) jegliche Kommunikation steuert. Da das HTTP-Protokoll für jede Antwort des Backend eine Anfrage des Frontend bzw. der App benötigt, läuft zusätzlich zu Express.js auch Socket.IO, welches durch das WebSocket-Protokoll eine bidirektionale Kommunikation ermöglicht.

Damit die Daten bei einem Neustart nicht verloren gehen, wird eine dateibasierte JSON-Datenbank (s. Abb. 4) verwendet. JSON ist ein Datenformat, welches sowohl Computer als auch Menschen einfach verstehen können. Durch node-json-db²⁰ wird die angegebene Datei ausgelesen und zur Verfügung gestellt.

```
{
  "settings": {
    "timezone": "Europe/Berlin",
    "location": {
      "city": "Wülfershausen",
      "countryCode": "DE",
      "lat": 50.0540866,
      "lon": 9.9995626
    }
  },
  "ringtones": [
    {
      "name": "Alarm",
      "location": "/ringtones/Alarm.mp3"
    }
  ],
  "alarms": [
    {
      "ringtone": "Alarm",
      "time": "06:18",
      "days": [
        true,
        true,
        true,
        true,
        true,
        true,
        false,
        false
      ],
      "enabled": true,
      "name": "Schule"
    }
  ],
  "initialized": true
}
```

Abb. 4: Komplette Datenbank in JSON-Format, eigene Abbildung vom 11.10.2022.

Um den Wecker zur richtigen Zeit klingeln zu lassen wird ein Weckservice beim Starten des Backend gestartet. Dieser überprüft jede Minute, ob ein Wecker klingeln soll. Falls dies der Fall ist, wird der Wecker dem Frontend übermittelt.

Dem Nutzer wird die Möglichkeit bereitgestellt eigene Klingeltöne für Wecker einzurichten. Diese müssen jedoch gespeichert werden, damit das Frontend beim Klingeln auch den entsprechenden Klingelton abspielen kann. Dies ist durch eine einfache Ordnerstruktur implementiert. Wenn der Nutzer über die App einen Klingelton hochlädt, wird dieser mit dem gewünschten Namen in dem Ordner „Ringtones“ gespeichert. Genannter Ordner ist auch über die API erreichbar, sodass alle Klingeltöne ohne Probleme abgespielt werden können.

²⁰ Aflalo, A., „node-json-db“, in: <https://github.com/BelpheMur/node-json-db>, zuletzt aufgerufen am 16.10.2022.

Da Daten, die vom Nutzer eingegeben werden, beim Übergeben an die API validiert werden müssen, das heißt es muss für alle Parameter überprüft werden, ob sie vorhanden und im richtigen Format sind, wird `express-zod-api`²¹ benutzt. Diese Bibliothek vereinfacht das Bauen von APIs mithilfe der Serverbibliothek `Express.js` und der Validierungsbibliothek `Zod`. Man definiert das Schema, also wie die Daten aussehen sollen, und `express-zod-api` validiert automatisch alle Daten.

8 Android App

Bei nativen App-Entwicklungslösungen muss man für jede Plattform die App in der plattformspezifischen Programmiersprache neu schreiben. Um dieses Problem zu umgehen, wird `React Native` verwendet, welches native Elemente auf Android, iOS und Windows erstellt. Um die sogenannte „Bridge“²² zwischen JavaScript und nativem Code zu bilden, muss mit Leistungseinbußen gerechnet werden²³, in der echten Welt sind diese Einbußen jedoch zu vernachlässigen.

Das asynchrone State Management in `React` wird durch `react-query` implementiert. Durch `Optimistic Updates` wird eine sehr einfache und schnelle Nutzererfahrung erstellt. Wenn der Nutzer ein Wecker entfernt, wird der Wecker von der App schon gelöscht, bevor das Backend den Wecker aus der Datenbank entfernt hat und eine Bestätigung gesendet hat. Dadurch fühlt sich die App deutlich schneller an.

Insgesamt wird durch diese Android App die Steuerung aller Funktionen vereinfacht und von überall im Heimnetz ermöglicht. Über eine einfache Benutzeroberfläche kann der Nutzer Wecker und Klingeltöne löschen, bearbeiten und hinzufügen. Auch diverse Einstellungen, wie z.B. der Ort, von dem das Wetter angezeigt wird, sind in der App verfügbar.

9 Kommunikation

9.1 Kommunikationsaufbau

Um die Programmierung zu vereinfachen, wird jegliche Verschlüsselung oder Verwendung von API-Schlüsseln außer Acht gelassen. 100 % der Top-100-Webseiten

²¹ Bocharova, A., „`express-zod-api`“, in: <https://github.com/RobinTail/express-zod-api>, zuletzt aufgerufen am 16.10.2022.

²² Ruda, N. S., „What is a bridge in `React Native` ?“, in: <https://www.geeksforgeeks.org/what-is-a-bridge-in-react-native/>, zuletzt aufgerufen am 30.10.2022.

²³ Gunalan, C., „Android App's Performance – Native vs Flutter vs `React Native`“, in: <https://blogs.perficient.com/2020/11/02/android-app-native-vs-flutter-vs-react-native/>, zuletzt aufgerufen am 30.08.2022.

benutzen SSL-Verschlüsselung²⁴, um sensitive Kundendaten sicher übertragen zu können. Da diese Benutzung des Smarten Weckers allerdings nur innerhalb des eigenen Heimnetz empfohlen wird und keine sensitiven Daten, außer der Position, übermittelt werden, kann dies vernachlässigt werden. Die andere Möglichkeit ist das Schützen der APIs vor Angreifen über einen Schlüssel, weil nur Nutzern mit dem richtigen Schlüssel Zugriff gewährt wird. So eine Implementierung einer Authentifizierung durch einen API-Schlüssel führt bei diesem Projekt zu zu hoher Komplexität, weswegen auch dieses Sicherheitskonzept nicht infrage kommt.

Um die Datenbank zu steuern, muss lediglich eine HTTP-Anfrage an das Backend im richtigen Format gesendet werden. Das Backend hört auf dem Port 3535 auf Anfragen. Alle Routen sind in diesem Format anzusprechen: „(Request-Typ) (IP des Backend):3535/api/(Methode)“. Die Methoden entsprechen den Einträgen in der Datenbank, die Request-Typen entsprechen HTTP-Methoden. Die HTTP-Methode „GET“ „dient zur Anforderung eines Dokuments oder einer anderen Quelle“²⁵. Weitere Methoden sind „DELETE“, „PATCH“ und „POST“, die jeweils für das Löschen, Ändern bzw. Erstellen von Ressourcen zuständig sind. Um also alle Klingeltöne aufzulisten, muss eine Anfrage mit der Methode „ringtones“ und dem Request-Typ „GET“ gesendet werden.

Um in Echtzeit Informationen über den Stand der Datenbank oder ob der Wecker klingeln soll zu erhalten, muss eine stetige WebSocket-Verbindung mit dem Backend aufrechterhalten werden. Diese Verbindung wird mithilfe der Socket.IO-Bibliothek deutlich vereinfacht, da Verbindungsabbrüche und jegliche Übertragungen von der Bibliothek übernommen bzw. vereinfacht werden. Somit muss nur die Nachricht und deren Inhalt definiert werden. Dass das Backend zwischen Frontend und App unterscheiden kann muss ein „type“-Parameter bei der Initialisierung mit entweder dem Wert „frontend“ oder „client“ übergeben werden, diese Information speichert das Backend ab. Durch diese Isolation können Nachrichten aufgeteilt werden, somit werden Nachrichten die nur für das Frontend bestimmt werden, z.B. „alarm“ (s. Punkt 9.3), auch nur an dieses gesendet.

²⁴ „HTTPS-Verschlüsselung im Web“, in: <https://transparencyreport.google.com/https/overview>, zuletzt aufgerufen am 11.10.2022.

²⁵ Reibold, H., „Hypertext Transfer Protocol“, in: HTTP-Methoden - HTTP-Grundlagen: Hypertext Transfer Protocol - TecChannel Workshop, zuletzt aufgerufen am 11.10.2022.

9.2 Android App mit Backend

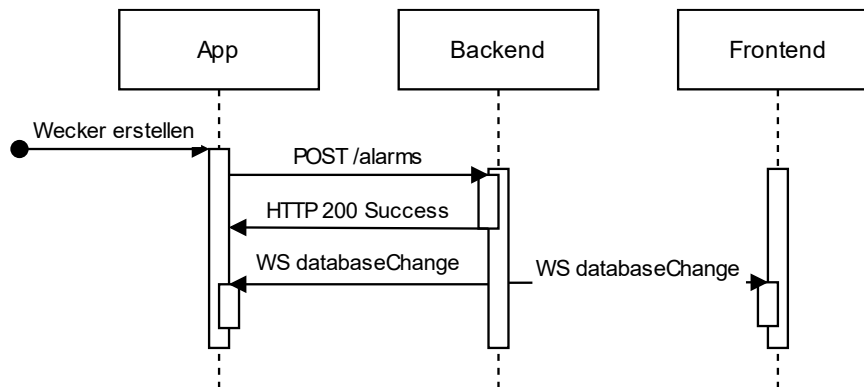


Abb. 5: Sequenzdiagramm eines Erstellvorgangs eines Weckers, eigene Abbildung vom 28.10.2022.

Der Erstellvorgang eines Weckers, wird im obenstehenden Diagramm (s. Abb. 5) anschaulich dargestellt. Sobald der Nutzer das Formular (s. Abb. 8) ausgefüllt hat, sendet die App eine Anfrage mit den Daten des zu erstellenden Weckers an das Backend. Nach erfolgreichem Speichern wird eine Erfolgsantwort zurückgesendet. Da sich nun die Datenbank (s. Punkt 7) um einen Weckereintrag erweitert hat, schickt das Backend eine weitere Nachricht an alle verbundenen Geräte, um sie über den neu hinzugefügten Wecker zu informieren.

9.3 Frontend mit Backend

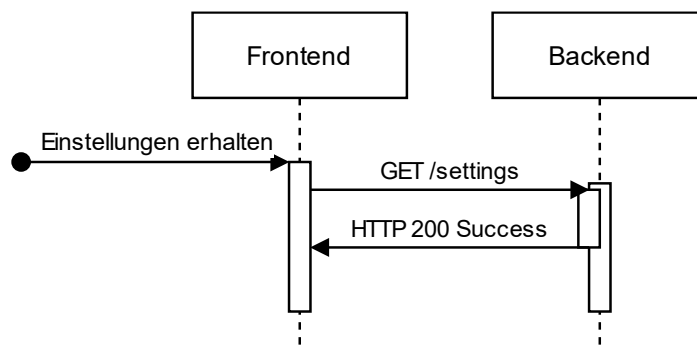


Abb. 6: Sequenzdiagramm des Erhaltungsvorgangs der Einstellungen, eigene Abbildung, vom 28.10.2022.

Das Frontend kommuniziert, wie die App, mit dem Backend über die API um Information, gespeichert in der Datenbank, abzufragen. In Abbildung 6 sieht man das Sequenzdiagramm wie das Frontend die Einstellungen vom Backend erhält. Dieses Prinzip kann man analog auf das Erhalten von den Weckern oder Klingeltönen anwenden.

Um das Frontend zu informieren, wenn ein Wecker klingeln soll, wird vom Backend durch den Weckservice eine Nachricht gesendet. Diese lautet in diesem Fall

„alarm“. Dabei werden Daten über den Wecker, wie z.B. den abzuspielenden Klingelton, übertragen.

9 Design

In einer jährlichen Umfrage von Venngage von 2021 stufen 49 % der Vermarkter visuelles Marketing als sehr wichtig für ihre Marketingstrategie ein, 22 % halten es für wichtig, und 19 % sagen, dass ihre Strategie ohne visuelle Inhalte nichts taugt.²⁶ Daher wurde in Figma²⁷, eine Software zum Erstellen von Prototypen für Oberflächen, ein Designprototyp für die App angelegt (s. Abb. 7), um eine gutaussehende und intuitiv nutzbare Oberfläche zu erstellen.

10 Right to Repair

Right to Repair ist eine Bewegung die fordert, „dass Produkte länger halten müssen und deshalb repariert werden sollten, wenn sie kaputt gehen“²⁸. Diese Bewegung gibt es, da „Elektroschrott [...] der am schnellsten wachsende Abfallstrom weltweit [ist].“²⁹.

Um Reparaturen so einfach wie möglich zu gestalten, sollte man sein Gerät reparieren können ohne seine

Garantie zu verlieren. Außerdem sollten Ersatzteile verfügbar sein, die man einfach einbauen kann. Um das Problem des Elektroschrotts zu umgehen ist der Smarte Wecker komplett reparierbar. Alle Komponenten (s. Punkt 2.1) sind käuflich zu erwerben bzw. einfach herzustellen. Außerdem ist das Gehäuse nicht verklebt, sondern verschraubt, somit ist das Ersetzen ohne Bestandteile zu beschädigen möglich.

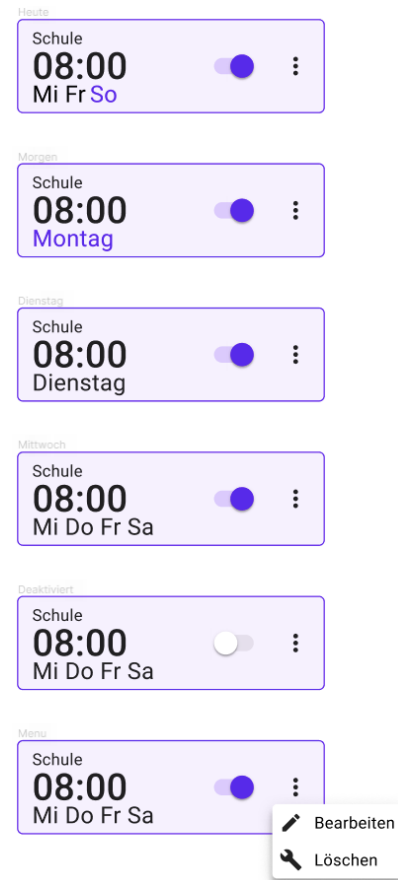


Abb. 7: Designprototyp der Wecker-Kacheln, eigene Abbildung, vom 01.11.2022.

²⁶ Khoja, N., „16 Visual Content Marketing Statistics to Know for 2022“, in: <https://venngage.com/blog/visual-content-marketing-statistics/>, zuletzt aufgerufen am 01.11.2022.

²⁷ „Figma: the collaborative interface design tool.“, in: <https://www.figma.com/>, zuletzt aufgerufen am 01.11.2022.

²⁸ „Right to Repair Europe“, in: <https://repair.eu/de/>, zuletzt aufgerufen am 01.11.2022

²⁹ Ebd.

11 Fazit und Ausblick

Zusammenfassend lässt sich feststellen, dass durch kleine Änderungen die Nützlichkeit des Smarten Weckers deutlich erhöht werden könnte.

Ein besseres Display mit OLED oder einer abschaltbaren Hintergrundbeleuchtung würde das Problem beseitigen, dass der Smarte Wecker in der Nacht das Zimmer ungewünscht erhellt.

Durch einen zweiten Lautsprecher und einem passenden Verstärker könnte ein deutlich besserer Stereosound erzeugt werden³⁰. Durch den besseren Klang würde sich ein Musikmodul anbieten, sodass man das Projekt auch als Lautsprecher nutzen kann.

Das Gehäuse auf dem Nachtschrank sieht durch die unnötig große Bauweise nicht besonders ansprechend aus. Durch kleinere Bildschirmränder und einem leicht angewinkelten Display würde der Smarte Wecker schöner aussehen und leichter benutzbar.

Außerdem kann das Projekt noch durch Funktionen, wie eine Anbindung an Google Assistant oder einen Sprachassistenten mit eingebautem Mikrofon, erweitert werden.

Sobald der Wecker klingelt könnte man auch eine Lampe einschalten sodass der Sonnenaufgang simuliert wird, welches „automatisch den Impuls [...] aufzuwachen und aktiv den Tag zu beginnen [fördert]“³¹.

Außerdem wären weitere Module, wie Staumeldung auf dem Weg zur Arbeit oder kurze Zusammenfassung der anstehenden Aufgaben oder Termine, denkbar.

³⁰ Lyn, G., „Mono vs Stereo: The Complete Guide“, in: <https://www.hifireport.com/mono-vs-stereo-the-complete-guide/>, zuletzt aufgerufen am 05.11.2022

³¹ Mikula, T., „Lichtwecker: Kaufberatung mit allen wichtigen Informationen“, in: <https://schlafwissen.com/lichtwecker>, zuletzt aufgerufen am 05.11.2022

12 Anhang

Weitere Bilder sind auf dem USB-Stick zu finden.

12.1 Screenshots der App

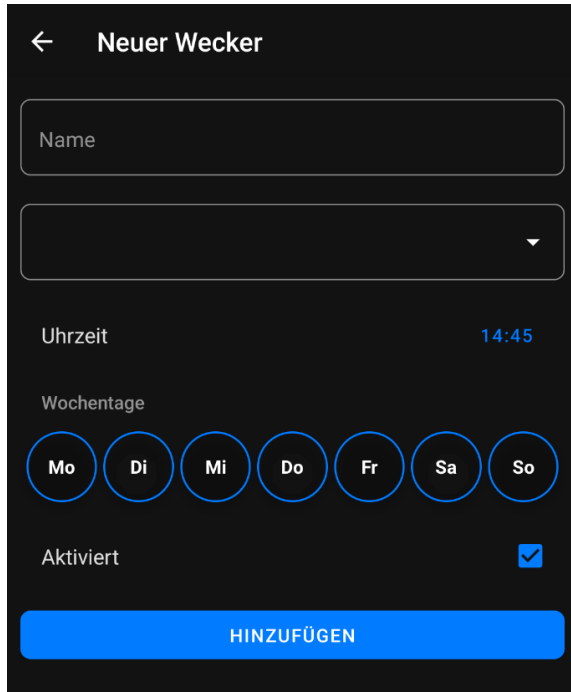


Abb. 8: Formular zum Erstellen eines Weckers, eigene Abbildung vom 05.11.2022.

12.2 Bilder des Gehäuses



Abb. 9: Schräge Ansicht des Gehäuses, eigene Abbildung vom 05.11.2022.

13 Abkürzungsverzeichnis

API	Anwendungsprogrammierschnittstelle
-----	------------------------------------

14 Literaturverzeichnis

14.1 Internetadressen

„3 Gründe, das Handy nicht als Wecker zu nutzen“, in <https://www.ongnamo-versand.de/blog/3-gruende-das-handy-nicht-als-wecker-zu-nutzen/>, zuletzt aufgerufen am 30.08.2022.

„7inch Capacitive Touch Screen LCD (B), 800×480, HDMI, Low Power“, in: <https://www.waveshare.com/7inch-hdmi-lcd-b.htm>, zuletzt aufgerufen am 30.08.2022.

Aflalo, Antoine, „node-json-db“, in: <https://github.com/Belphemur/node-json-db>, zuletzt aufgerufen am 16.10.2022.

Bocharova, Anna, „express-zod-api“, in: <https://github.com/RobinTail/express-zod-api>, zuletzt aufgerufen am 16.10.2022.

„Can Your Google Home Or Google Nest Be Hacked? Here's How“, in: <https://robotpowered-home.com/can-a-google-home-be-hacked/>, zuletzt aufgerufen am 22.03.2022.

„DEBO SOUND AMP1 Entwicklerboards - Audioverstärker, Klasse D, PAM8302A“, in: <https://www.reichelt.de/entwicklerboards-audioverstaerker-klasse-d-pam8302a-debo-sound-amp1-p235505.html>, zuletzt aufgerufen am 30.08.2022.

„Dropdown-Liste“, in: <https://de.wikipedia.org/wiki/Dropdown-Liste>, zuletzt aufgerufen am 01.11.2022.

Durant, Tobias, „Ist Holz nachhaltiger als Plastik“, in: <https://citizensustainable.com/de/holz-nachhaltiger-plastik/>, zuletzt aufgerufen am 30.08.2022.

„Figma: the collaborative interface design tool.“, in: <https://www.figma.com/>, zuletzt aufgerufen am 01.11.2022.

„Geocoding API“, in: <https://openweathermap.org/api/geocoding-api>, zuletzt aufgerufen am 01.11.2022.

Gunalan, Chellakkannu, „Android App's Performance – Native vs Flutter vs React Native“, in: <https://blogs.perficient.com/2020/11/02/android-app-native-vs-flutter-vs-react-native/>, zuletzt aufgerufen am 30.08.2022.

Hapke, Thomas, „Was ist eigentlich Peer Review?“, in: <https://www.tub.tuhh.de/wissenschaftliches-arbeiten/2018/02/14/was-ist-eigentlich-peer-review/>, zuletzt aufgerufen am 30.09.2022.

Hasse, Kerstin, „Mehr als ein fancy Wecker: Der Google Nest Hub im Test“, in: <https://www.annabelle.ch/stil/mehr-als-ein-fancy-wecker-der-google-nest-hub-im-test/>, zuletzt aufgerufen am 30.08.2022.

„HTTPS-Verschlüsselung im Web“, in: <https://transparencyreport.google.com/https/overview>, zuletzt aufgerufen am 11.10.2022.

Khoja, Nadya, „16 Visual Content Marketing Statistics to Know for 2022“, in: <https://venngage.com/blog/visual-content-marketing-statistics/>, zuletzt aufgerufen am 01.11.2022.

Lyn, Gordon, „Mono vs Stereo: The Complete Guide“, in: <https://www.hifireport.com/mono-vs-stereo-the-complete-guide/>, zuletzt aufgerufen am 05.11.2022

Mikula, Tobias, „Lichtwecker: Kaufberatung mit allen wichtigen Informationen“, in: <https://schlafwissen.com/lichtwecker>, zuletzt aufgerufen am 05.11.2022

„Nest Hub Max - home screen customization“, in: <https://www.googlenestcommunity.com/t5/Speakers-and-Displays/Nest-Hub-Max-home-screen-customization/td-p/83948>, zuletzt aufgerufen am 22.03.2022.

Nowak, Paul, „The Powerful Economic Underpinnings of OSS“, in: <https://web.archive.org/web/20031204125550/http://www.linuxworld.com/story/34293.htm>, zuletzt aufgerufen am 30.09.2022.

„One Call API 2.5“, in: <https://openweathermap.org/api/one-call-api>, zuletzt aufgerufen am 01.11.2022.

„Plastik ist zu billig: Behörde warnt vor Recycling-Problemen“, in: <https://www.rnd.de/wirtschaft/plastik-ist-zu-billig-behorde-warnt-vor-recycling-problemen-CFNJXXBVM7HHZOLK5ME-CACDXQM.html>, zuletzt aufgerufen am 30.08.2022.

„Plastikmüll“, in: <https://de.statista.com/themen/4645/plastikmuell/#dossierKeyfigures>, zuletzt aufgerufen am 30.08.2022.

Rüsche, Alexandra, „Umwelt schonen und trotzdem schöner Wohnen mit Holz“, in: <https://www.bauen-wohnen-aktuell.de/wohnen/umwelt-schonen-und-trotzdem-schoener-wohnen-mit-holz/>, zuletzt aufgerufen am 30.08.2022.

„Raspberry Pi 4 Modell B 4GB SDRAM“ in: <https://www.rasppishop.de/Raspberry-Pi-4-Modell-B-4GB-SDRAM?src=raspberrypi>, zuletzt aufgerufen am 30.08.2022.

Reibold, Holger, „Hypertext Transfer Protocol“, in: <https://www.tecchannel.de/a/hypertext-transfer-protocol,401210,6>, zuletzt aufgerufen am 11.10.2022.

„Right to Repair Europe“, in: <https://repair.eu/de/>, zuletzt aufgerufen am 01.11.2022

Ruda, Neha Shri, „What is a bridge in React Native ?“, in: <https://www.geeksforgeeks.org/what-is-a-bridge-in-react-native/>, zuletzt aufgerufen am 30.10.2022.

Schmidt, Matthias, Wenn der Wecker zweimal klingelt, in <https://yougov.de/news/2016/05/28/wenn-der-wecker-zweimal-klingelt/>, zuletzt aufgerufen am 30.08.2022.

Storath, Bernd, „Smart Alarm Clock“, in: <https://github.com/kaaax0815/smart-alarm-clock>, zuletzt aufgerufen am 30.08.2022.

„VIS 2243 Breitbandlautsprecher BF 32, 2 W, 4 Ohm“, in: <https://www.reichelt.de/de/de/breitband-lautsprecher-bf-32-2-w-4-ohm-vis-2243-p276918.html?r=1>, zuletzt aufgerufen am 30.08.2022.

14.2 Bildquellen

Abbildung 1: eigene Abbildung vom 30.08.2022.

Abbildung 2-3: eigene Abbildung, vom 16.10.2022.

Abbildung 4: eigene Abbildung vom 11.10.2022.

Abbildung 5-6: eigene Abbildung vom 28.10.2022.

Abbildung 7: eigene Abbildung, vom 01.11.2022.

Abbildung 8-9: eigene Abbildung, vom 05.11.2022.

15 Eidesstattliche Erklärung

„Ich erkläre hiermit, dass ich die Seminararbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.“

....., den

Ort

Datum

Unterschrift des Schülers