

# Projekt Softwaretechnik / Medieninformatik

## Meilenstein 1

18. Oktober 2023



Projekt

**eLunch**

Firma: Steinbeis Embedded Systems Technologies  
GmbH

M. Kaan Asik	766381
Esra Balci	766561
Selim Cetin	766916
Vivian Schmiss	766236
Imran-Nur-Reyhan Sevinc	766668

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort .....</b>	<b>1</b>
<b>2</b>	<b>Generell .....</b>	<b>2</b>
2.1	Zielgruppe.....	2
2.2	Probleme .....	2
2.3	Eigenschaften.....	2
<b>3</b>	<b>Team-Organisation.....</b>	<b>3</b>
<b>4</b>	<b>User Stories .....</b>	<b>4</b>
<b>5</b>	<b>Flow Chart .....</b>	<b>5</b>
<b>6</b>	<b>Funktionsumfang .....</b>	<b>6</b>
<b>7</b>	<b>User Interface .....</b>	<b>7</b>
<b>8</b>	<b>Technologien .....</b>	<b>9</b>
8.1	Frontend.....	9
8.1.1	React.js.....	9
8.1.2	Material UI.....	9
8.2	Backend .....	9
8.2.1	Node.js.....	10
8.2.2	Express.js .....	10
8.2.3	MongoDB .....	10
8.2.4	API .....	11
<b>9</b>	<b>Quellenverzeichnis.....</b>	<b>12</b>
9.1	Online-Quellen.....	12

# **1 Vorwort**

In einer Zeit, in der die Bedeutung von gesunder Ernährung und ausgewogenen Mahlzeiten immer mehr an Bedeutung gewinnt, haben wir uns das Ziel gesetzt, die Essensplanung zu revolutionieren und den Mitarbeitern der Steinbeis Embedded Systems Technologies GmbH dabei zu helfen, sich gesünder und energiegeladener zu fühlen.

Wir haben uns für die Firma Steinbeis, die sich das Projekt eLunch, ein Tool zur Planung eines Speiseplans überlegt hat, entschieden. Dafür hat die Firma bereits ein Produkt, das nun durch unseres abgelöst werden soll.

## **2 Generell**

### **2.1 Zielgruppe**

Grundsätzlich ist unsere Zielgruppe durch den Kunden des Projektes definiert. Unser Speiseplan-Tool richtet sich exklusiv an die Mitarbeiter der Firma und wird entwickelt, um deren Essenserlebnis zu verbessern und einen gesunden Lebensstil zu fördern. Diese Software wird maßgeschneidert für die Bedürfnisse und Anforderungen der Mitarbeiter. Unser Speiseplan-Tool wird für alle Mitarbeiter der Steinbeis GmbH zugänglich sein. Spezifischer ist unsere Zielgruppe also durch die Mitarbeiter und den Admin des Unternehmens definiert.

### **2.2 Probleme**

Als grundsätzliches Problem stellt sich uns die Übersichtlichkeit der gesamten Oberfläche. Die Veranschaulichung der Gerichte innerhalb des Monats, das Eintragen bzw. das Ändern von Gerichten und die Eingabe der Zutaten in eine einfach lesbare Form, stellt also unsere größte Herausforderung dar. Hinzuzufügen ist, dass wir keine Experten für die von uns verwendeten Technologien sind. Zweifelsohne haben wir uns im Vorfeld für Technologien und Programmiersprachen entschieden, die uns liegen und bei deren Verwendung wir uns sicher fühlen. Dennoch können wir nicht mit Sicherheit voraussagen, dass wir im Projekt nicht Problemen gegenüberstehen werden, die unser aktuelles Können herausfordern und deren Lösung mehr Zeit als geplant in Anspruch nehmen wird.

### **2.3 Eigenschaften**

Wir schreiben eine Web-Applikation, die das gemeinsame Kochen und dessen Planung in einer Firma simplifizieren soll. In einer MongoDB werden alle Daten gespeichert, welche mit Node.js implementiert werden. Die genaue Darstellung der Webseite werden wir anhand HTML, CSS und React.js veranschaulichen. Es gibt genau zwei Nutzergruppen, darunter sind die Mitarbeiter und der Admin. Deren Rechte und Anwendungsfälle werden unter Kapitel 4 User Stories genau erläutert.

### 3 Team-Organisation

In unserem Team soll jeder bei allem mal dabei gewesen sein und alles mal gesehen haben. Jeder muss Code geschrieben haben, muss wissen, wie unser Backend funktioniert, sollte die Dokumentation kennen und einen Teil davon geschrieben haben und jeder sollte in Figma mitgewirkt haben.

Dennoch gibt es grobe Rollen in unserem Team. Für das Backend sind Selim und Kaan zuständig. Das bedeutet, dass sie dafür verantwortlich sind, diese zu koordinieren und aufzusetzen. Zusätzlich kümmern Sie sich um die API.

Esra ist im Team die Projektleiterin. Sie koordiniert das gesamte Team und ist zuständig für die Kommunikation mit Steinbeis und organisiert die Meetings.

Für das Protokollieren der Meetings werden wir uns jede Woche abwechseln. Die Protokolle müssen immer aktuell und so ausführlich geschrieben sein, dass ein Projektfremder sofort weiß, worum es geht und auf welchem Stand das Projekt aktuell ist. Für die Dokumentation ist jeder zuständig, die einzelnen Dokumentationsteile werden von jedem Teammitglied einzeln zusammengestellt. Esra, Imran und Vivian formatieren, überprüfen und ergänzen diese anschließend.

Für Figma sind Imran, Esra und Vivian zuständig. Die benötigten Mockups werden zu koordinieren, zu erstellen und zu einem einheitlichen Design zusammen zu fügen.

Das Kanban-Board haben wir auf GitHub hinzugefügt. GitHub liegt in Esras Verantwortung. Sie hat das Projekt aufgesetzt und wird es pflegen. Die Zuweisung von Issues, Post-Requests bearbeiten, Merge-Requests bearbeiten bzw. Personen zuweisen und allgemein darauf achten, dass das Projekt strukturiert auf Git gepflegt ist, liegt in Esras Hand.

Beim Coden des Frontends sind ebenfalls Imran, Esra und Vivian zuständig.

Jeder arbeitet an allem mit und leistet prozentual gleich viel für das Projekt.

## 4 User Stories

Der Mitarbeiter möchte...

- ... auf die Wochen- und Monatsansicht des Speiseplans zugreifen können.
- ... in der Wochenansicht neue Gerichte hinzufügen, wenn noch keine eingetragen wurden.
- ... sich bei Gerichten zum Mitessen ein- oder austragen.
- ... auf seine Finanzübersicht zugreifen können.

Der Admin möchte...

- ... eingetragene Gerichte bearbeiten oder neue hinzufügen.
- ... die Ein- und Ausgaben der Mitarbeiter bearbeiten können.
- ... die Statistik- und Mitarbeiteransicht einsehen können.

## 5 Flow Chart

Wir haben ein Flow Chart vorbereitet, siehe Abbildung 1, welches aufführt, wie die verschiedenen Funktionen auf der Webseite mit anderen Screens verknüpft sind.



Abbildung 1: Flow Chart zum Ablauf der Steuerung der Webseite

## 6 Funktionsumfang

Aufgabe	Aufwand in Stunden
API Definition in OpenAPI Format	10 h
Backend Routing implementieren	60 h
Backend Verbindungsaufbau zur DB implementieren	10 h
Backend Tests (Unit Tests)	30 h
Frontend Grundgerüst für UI erstellen	40 h
Frontend UI abschließend implementieren, Buttons, Daten visualisieren...	60 h
UI/Mockup Design	10 h
Frontend Tests	20 h
User Testing	10 h
Projektmanagement in Git mit Branches, Mergekonfliktlösung...	20 h
Dokumentation	20 h
Präsentationsvorbereitung	10 h
<b>Gesamtaufwand</b>	300 h



## 7 User Interface

Im Folgenden sind unsere Mockups in den Abbildungen 2, 3, 4 und 5 zu sehen.

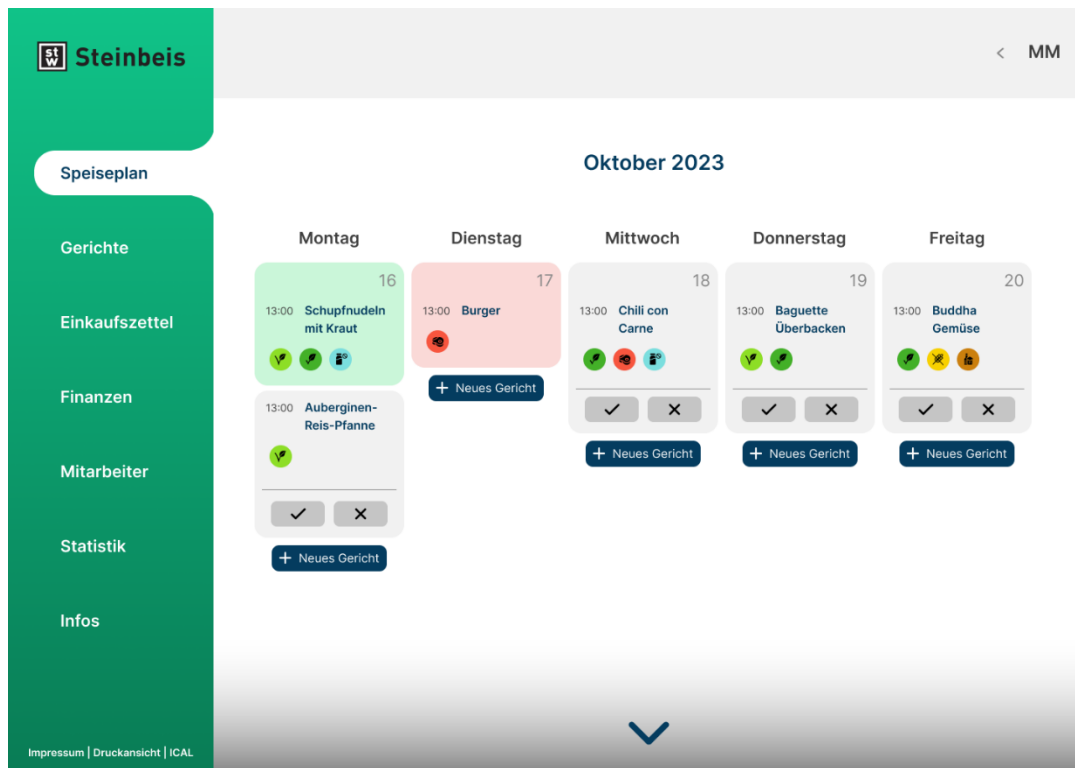


Abbildung 2: Wochenansicht vom Speiseplan

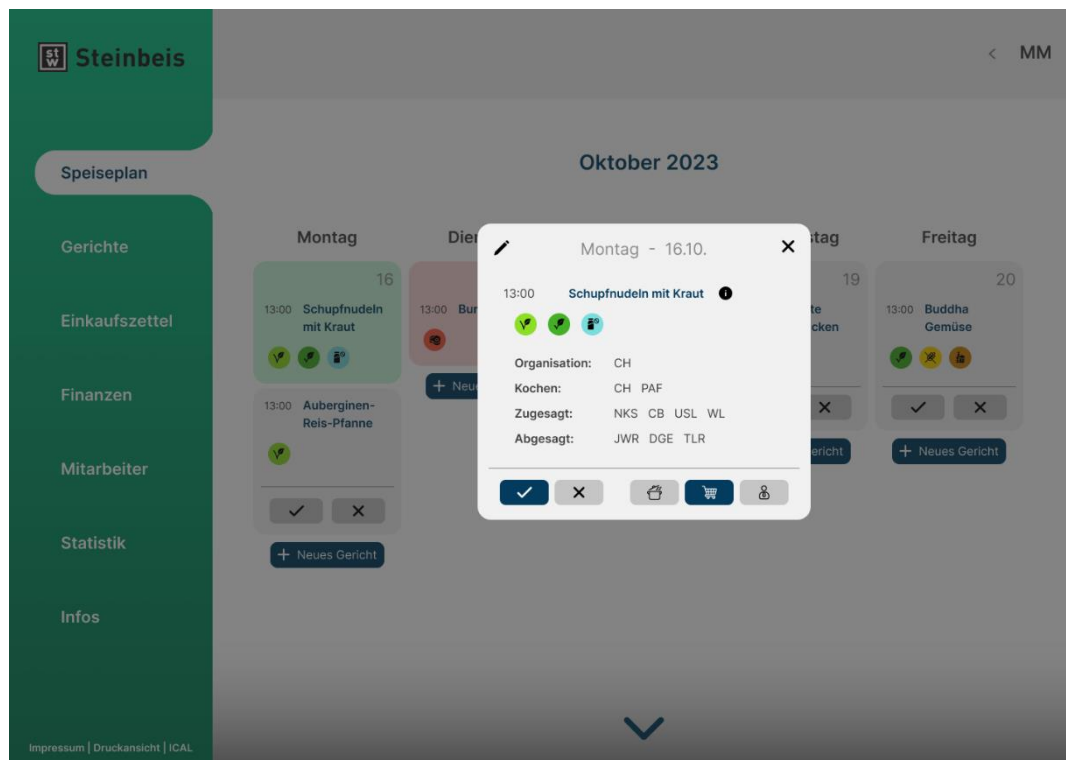


Abbildung 3: Detailansicht von einem Tagesgericht

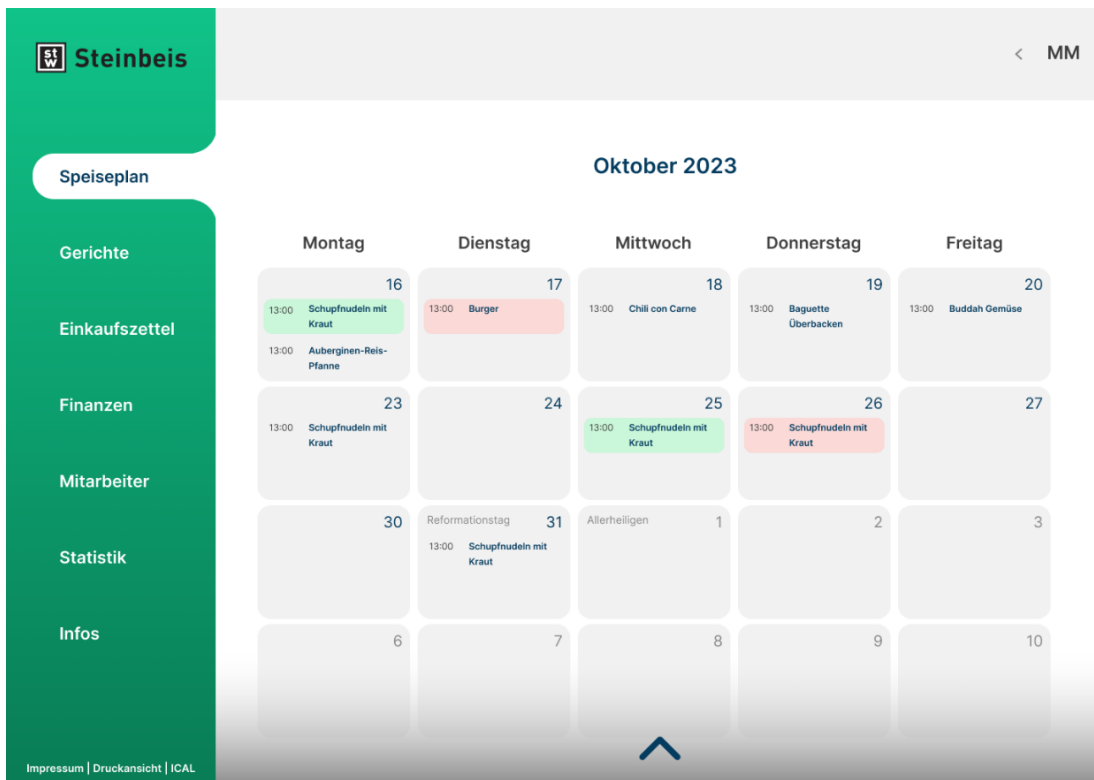


Abbildung 4: Monatsansicht vom Speiseplan

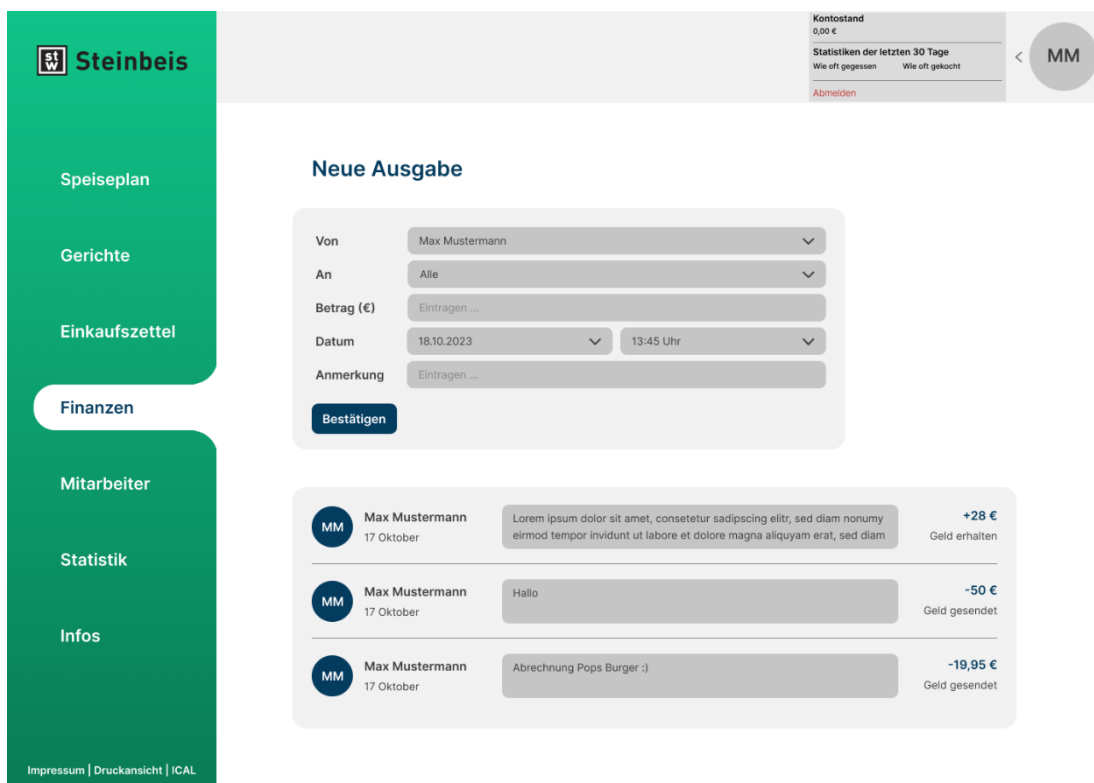


Abbildung 5: Finanzansicht

## 8 Technologien

Für die Architektur bzw. Technologien möchten wir sowohl für das Backend als auch für das Frontend JavaScript als Programmiersprache nutzen. Dabei nutzen wir Node.js als Runtime Environment für unser Backend und React.js als Frontend User Interface. Für die Datenhaltung möchten wir die MongoDB in Kombination mit der Mongoose Library nutzen.

### 8.1 Frontend

#### 8.1.1 React.js

Für das Frontend haben wir uns für React.js entschieden. React.js ist eine JavaScript Library, die von Facebook entwickelt wurde. Mit ihr ist es möglich, interaktive, dynamische und komplexe Webseiten zu gestalten. Sie ist bekannt für das Rendering-Prinzip, welches dynamische Seiten ermöglicht, ohne die gesamte Seite neu laden zu müssen.

#### 8.1.2 Material UI

Zudem verwenden wir Material UI, welches uns vorgefertigte Komponenten anbietet, die wir anschließend auf unsere Bedürfnisse anpassen. Dies ermöglicht uns ein schnelles und effektives Erstellen des User-Interfaces. Die umfangreiche Dokumentation und die gute Benutzerfreundlichkeit unterstützt uns dabei

### 8.2 Backend

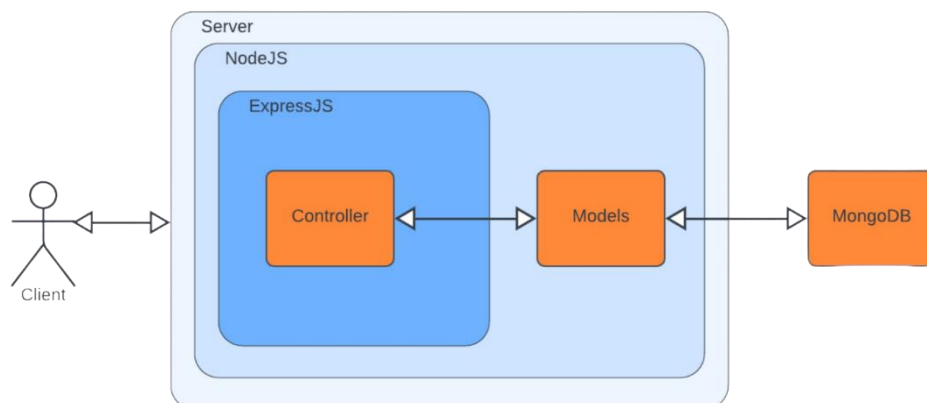


Abbildung 6: Beschreibt den Aufbau unseres Backends

### **8.2.1 Node.js**

Für das Backend verwenden wir Node.js. Es ist ein Open-Source Runtime Environment für JavaScript, die es ermöglicht, serverseitige Anwendungen mit JavaScript zu entwickeln. Node.js verwendet eine ereignisgesteuerte, nicht blockierende I/O-Architektur, die es dem Entwickler ermöglicht, skalierbare und schnelle Web-Applikationen zu erstellen. Es wird meistens für die Erstellungen von Webanwendungen und APIs verwendet.

Wir haben uns für Node.js entschieden, da es sehr verbreitet und beliebt ist. Außerdem gibt es eine große Auswahl an Third-Party-Tools, die sehr einfach zu installieren und nutzen sind.

### **8.2.2 Express.js**

Express ist ein Web-Application-Framework für Node.js. Es ermöglicht den Entwicklern, robuste APIs und Webserver auf saubere und einfachere Weise zu erstellen. Es ist ein lightweight Paket, dass die Hauptfunktionen von Node.js nicht verdeckt.

### **8.2.3 MongoDB**

MongoDB ist eine NoSQL-Datenbank, die als Datenbanktechnologie weit verbreitet ist. Im Gegensatz zu traditionellen Datenbanken, speichert MongoDB Daten in ein dokumentorientiertes Format. Dies ermöglicht mehr Flexibilität und Skalierbarkeit. MongoDB wird oft bei modernen Webanwendungen und mobilen Apps verwendet. Deswegen ist es auch bekannt in einem Stack namens "MERN": MongoDB, Express, React.js, und Node.js.

Als Unterstützung für die Nutzung der MongoDB via Backend werden wir die Mongoose Library verwenden.

Aufgrund der Popularität und somit vorhandener Dokumentation und Support haben wir uns für MongoDB entschieden.

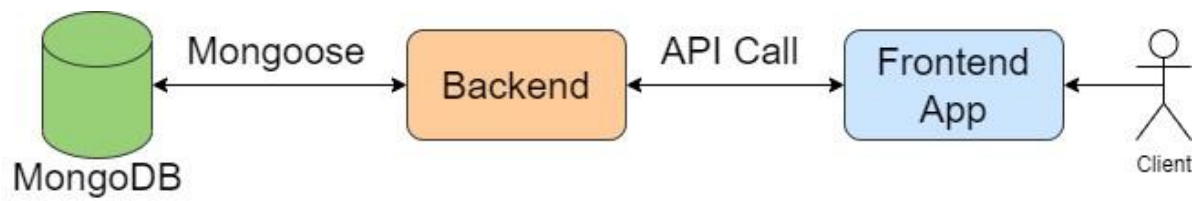


Abbildung 7: MERN Struktur

#### 8.2.4 API

Für die Dokumentation unserer API nutzen wir OpenAPI.

## 9 Quellenverzeichnis

### 9.1 Online-Quellen

- <https://www.hs-esslingen.de>
- <https://www.figma.com>
- <https://app.diagrams.net/>
- <https://www.lucidchart.com>