



**«MeNeW»**

**«Clever planen, bewusster essen  
– Menüplanung smart gemacht»**

---

Technische Informationen für die Jury



## Technische Informationen für die Jury

### Aktueller Stand des Sourcecodes

- Link zu Github Repository: <https://github.com/BaernHaeckt-MeNeW/MeNeW>
- Der aktuelle Stand umfasst eine funktionierende Version, welche die zentralen Features enthält. Wir sind iterativ vorgegangen und haben nach und nach implementiert, was in der Zeit möglich war.
- Einzelne Features wurden bereits im Frontend platziert, aber noch nicht funktional implementiert.
- Das Frontend wurde nur für Mobile Viewports implementiert.

### Ausgangslage

- Unser Fokus lag bei der Inspiration von Mahlzeiten.
- Als Basis entwickelten wir ein Haushaltsverwaltung und ein einfachen Planungsinstrument für die drei Hauptmahlzeiten.
- Anschliessend wurde die Inspiration implementiert und laufend erweitert.
- Wir wollten eine Mobile-First Webapplikation erstellen, da der Zugang zur Applikation möglichst einfach sein soll.

### Technischer Aufbau

- Wir verwenden Spring Boot für das Heavy Lifting unserer REST Schnittstelle. Die Logik ist in einer gängigen Struktur organisiert.
- Im Rahmen des Hackathon wird eine H2 Datenbank verwendet und mit Mahlzeiten und Personen geseedet.
- Als GenKI wird GPT-4o-mini per OpenAI API eingebunden.
- Das Userinterface basiert auf React mit Typescript, Tailwind sowie Vite als Build-System. Wiederverwendbare Komponenten sind ausgelagert. Die Repo-Struktur entspricht gängigen Standards.

### Implementation

- Haushaltsverwaltung sowie Mahlzeitplanung sind einfache CRUD-Endpunkte, welche via Spring Data JPA in die H2 Datenbank persistieren.
- Die Inspiration läuft über einen Endpunkt, welcher den Dialog mit der KI sowie den Tag und die Mahlzeit entgegennimmt. Alle anderen Daten bezieht das Backend direkt aus der Datenbank (Diäten, No-Gos, Gewohnheiten).
- Die GenKI wird per Systemprompt instruiert. Wichtige Aspekte wie Ausgabeformat etc. sind entsprechend gehighlighted.
- Die Userprompt enthält alle wichtigen Aspekte für die Menüinspiration wie Diäten, No-Gos, die zuletzt abgelehnten Menüvorschläge etc.
- Das Frontend umfasst mehrere Seiten, welche via react-router navigiert werden können. Es wurden keine UI-Libraries oder Formular-Lösungen verwendet, um flexibel zu bleiben.

### Abgrenzung / Offene Punkte

- Wir haben den Freitagabend hauptsächlich am Whiteboard verbracht, wo wir sämtliche Screens der App konzipiert haben. Relativ schnell wurde uns klar, dass eine Applikation, die das Problem von der Ideenfindung bis hin zur Einkaufsliste löst, in der Zeit des Hackathons nicht in einer Qualität möglich ist, die uns zufriedenstellen würde.
- Aus diesem Grund haben wir uns darauf geeinigt, dass wir uns primär auf die Ideenfindung konzentrieren und dann iterativ die weiteren Features dazunehmen, soweit wir dazu Zeit haben.
- Features, die wir uns für einen zukünftigen Release vorstellen könnten:
  - Einkauf planen → User legt fest, für wie viele Tage er einkaufen möchte. App gibt eine provisorische Einkaufsliste, anhand derer der User sagen kann,

welche Zutaten er bereits zu Hause hat. Dies ist pragmatischer als ein «Stock Management» der Zutaten.

- Hinzufügen von temporären Mitessern → Day-by-Day Hinzufügen von Gästen, Mitbewohnern
- Wochenplanung → Aktuell funktioniert die Planung nur pro Mahlzeit. Wir würden eine Funktion ermöglichen wollen, in der der User sich eine ganze Woche planen kann.
- Ergänzung der Vorschläge mit Bilder und Rezept
- «Kantinenmodus» → Aktuell basieren die Vorschläge auf einem Haushalt, wo sämtliche Essgewohnheiten immer respektiert werden müssen. In einer Kantine wäre es möglich, mehrere Gerichte anzubieten, die nicht alle allen Vorgaben entsprechen müssten (z.B. Vegi-Menu nebst Fleisch-Menu)
- Getränke, Vorspeisen, Desserts usw.
- Ausbau des Frontends auf mehrere Viewports oder sogar Implementierung von nativen Apps für Mobileplattformen.