

# Projektplanung - Kundendatenbank für Bestellungen

## Brainstorming

### Hauptfunktionen:

- **Bestellungen hinzufügen:** Der Benutzer kann eine Bestellung mit Kunden- und Bestellinformationen eingeben.
- **Bestellungen löschen:** Bestellungen können durch Eingabe der Bestell-ID gelöscht werden.
- **Bestellungen anzeigen:** Alle gespeicherten Bestellungen werden angezeigt.
- **Persistente Speicherung:** Alle Bestellungen werden in einer **CSV-Datei** gespeichert und beim Neustart des Programms geladen.

## Sortieren des Brainstormings

### Hauptanforderungen:

- **Benutzerinteraktion:** Der Benutzer kann Bestellungen hinzufügen, löschen und anzeigen.
- **Datenmanagement:** Bestellungen werden in einer **CSV-Datei** gespeichert.
- **Benutzeroberfläche:** Eine textbasierte **CLI** (Command-Line Interface).
- **Datenmodell:** Eine **Order-Klasse**, die sowohl Bestell- als auch Kundendaten enthält.

### Technische Anforderungen:

- Verwendung von **Java** für die Implementierung.
- Speicherung der Bestellungen in einer **CSV-Datei**.

## Allgemeine Beschreibung des Projektziel

Das Ziel des Projekts ist es, ein einfaches System zur Verwaltung von Bestellungen zu entwickeln, bei dem jede Bestellung sowohl **Kunden-** als auch **Bestellinformationen** enthält. Bestellungen können hinzugefügt, angezeigt und gelöscht werden. Die Bestellungen werden in einer **CSV-Datei** gespeichert, um Daten auch nach einem Neustart der Anwendung beizubehalten. Die Anwendung wird eine **CLI** bieten.

## Ist-Analyse

Es gibt keine automatisierte Verwaltung von Bestellungen. Alle verwendeten Daten gehen nach Beenden der Anwendung verloren. Es gibt keine Möglichkeit, Bestellungen anzuzeigen oder zu bearbeiten.

## Soll-Analyse

Das Projekt zielt darauf ab, eine Anwendung zur Verwaltung von Bestellungen zu entwickeln, die Bestelldaten in einer CSV-Datei speichert und über eine textbasierte Benutzeroberfläche (CLI) verwaltet wird. Die zentrale Order-Klasse speichert Bestell- und Kundendaten, während die OrderManager-Klasse die Geschäftslogik übernimmt, einschließlich Hinzufügen, Löschen und Abrufen von Bestellungen. Das System ermöglicht die persistente Speicherung von

Bestellungen und gewährleistet deren Wiederherstellung bei jedem Start der Anwendung. Die CLI stellt Funktionen zur Eingabe neuer Bestellungen, zum Löschen von Bestellungen und zum Anzeigen aller Bestellungen zur Verfügung. Zudem wird auf Fehlerbehandlung und Eingabevalidierung geachtet, um eine stabile und benutzerfreundliche Anwendung zu gewährleisten.

## Aufgabenaufteilung - Milestones

### Tag 1 (7 Stunden) - Kernklassen und CSV-Handling implementiert:

- **Projektstruktur** einrichten, `Order`-Klasse erstellen, CSV-Handling vorbereiten.
- **Testen** der `Order`-Klasse und der CSV-Methoden.
- **Skizzieren** der `OrderManager`-Klasse zur Bestellverwaltung.

### Tag 2 (6 Stunden) - Geschäftslogik und Benutzerinteraktion funktionsfähig:

- **Implementierung der `OrderManager`-Klasse** zur Verwaltung von Bestellungen.
- **Erstellung der CLI** zur Interaktion mit dem Benutzer.
- **Testen** der Anwendung, Eingabevalidierung und Fehlerbehandlung abschließen.