|  |
| --- |
| Modul 153 LB02 |
| Lagerverwaltungssystem |
| Alan Parmaksiz und Phearum Svay |

Inhalt

[1. Anforderungsanalyse 2](#_Toc61254500)

[1.1 Probleme 2](#_Toc61254501)

[2.2 Funktionale Anforderungen 2](#_Toc61254502)

[2. Projektmanagementmethode 2](#_Toc61254503)

[3. Projektablauf 2](#_Toc61254504)

[4. Terminplanung 2](#_Toc61254505)

[5. Datenmodelle 2](#_Toc61254506)

[6. Entstandene Probleme 2](#_Toc61254507)

[7. Testing 2](#_Toc61254508)

# Anforderungsanalyse

* 1. **Vorhandenes Problem**

Ein Online-Shop möchte seine komplizierte Lagerverwaltung verbessern. Da er noch mit einer riesigen Excel-Tabelle arbeitet, möchte er nun auf eine richtige Datenbank umsteigen.

* 1. **Funktionale Anforderungen**

Die Datenbank beinhaltet und kann…

* … alle Artikel anzeigen.
* … alle Mitarbeiter anzeeigen.
* … die Produktkategorie anzeigen.
* … die Positionen der Artikel anzeigen.
  + die Position wird als Stockwerk, Regal und Ablage angezeigt.
* … die Anzahl an Artikel anzeigen.
* … den Hersteller der Artikel anzeigen.
* … der Standort des Lagers vom Artikel anzeigen.
* … Bewegungen bzw. Ein- und Ausgang anzeigen.
* … Backups anzeigen.
  1. **Nicht funktionale Anforderungen**

Bei der Implementation der Datenbank wird die Entwicklungsumgebung *DataGrip* und *Docker* verwendet. Als Datenbankmanagementsystem wurde *MariaDB* gewählt. Für die Dokumentation die Bewegungen bzw. Ein- und Ausgänge werden *Trigger* verwendet. Der Trigger reagiert, wenn Artikel das Lager betreten, verlassen oder ihre Position im Lager verändern. Zusätzlich gibt es zwei *stored Procedures*. Eine *Procedure* kreiert ein Backup des ganzen Lagers und die andere *Procedure* sollte von einem Event aufgerufen werden. Das Event ruft jede fünf Minuten die Procedure auf, die alle Artikel auf ihre Verfügbarkeit überprüft.

* 1. **Randbedingungen**
  2. **Ermittlungstechniken**

# Projektmanagementmethode

# Projektablauf

# Terminplanung

# Datenmodelle

# Entstandene Probleme

# Testing