# Teilnehmer/innen des Teams:

|  |  |
| --- | --- |
| Klasse:  BI17b | Team: Vasan & Jenia  Beispiel Dokumentation M226B LB2 |

# Anforderungsdefinition (Meilenstein A)

|  |  |
| --- | --- |
| „Greenfoot Schiffe versenken“ | |
| **Auftrag:**  (Allgemeine Beschreibung) | Nutzen: Es Wird ein Schiffe versenken Spiel in Greenfoot Programmiert. Auf Zwei Spielfeldern, werden Schiffe in diversen Grössen auf einem Raster mit der Grösse 10x10 platziert. Gegenseitig sollten die Spieler ihre Felder nicht sehen. Abwechslungsweise kann ein Spieler eine Bombe auf dem Raster des gegnerischen Spielers platzieren. Wenn alle Schiffe auf dem Gegnerischen Raster getroffen wurden, ist das Spiel fertig und man hat gewonnen. **Szenario:**  * Zwei Raster werden nebeneinander generiert. * Auf beiden Raster Werden Schiffe erstellt.   **Details:**   * Wird ein Feld bombardiert, wird dieses Markiert * Getroffene Schiffe werden als solches markiert. * Anzahl Verbleibenden Schiffe werden angezeigt.   **Mockup:** |
| **MUSS**  **Kriterien:**  (Konkrete Features, die umzusetzen sind) | **Folgende Features sollen implementiert werden (Funktionalität):**   * Zwei Felder müssen erstellt werden. * Schiffe müssen in einem Feld ersichtlich sein. * Im Gegner Feld kann man mit einem Klick ein Raster auswählen und ein Schiff versenken. * Wenn Alle Schiffe versunken wurden, hat man Gewonnen und dies Wird so angezeigt. * Wird ein leeres Feld getroffen, wird dieses Markiert * Spieldaten werden in einem Array gespeichert. |

|  |  |
| --- | --- |
| **KANN**  **Kriterien:**  (Konkrete Features, die optional sind) | **Folgende Features können zusätzlich implementiert werden: (Kreativität)**   * Das Spiel sollte als Multiplayer möglich sein * Daten sollten über das Netzwerk kommunizierbar sein * Anzahl Verbleibenden Schiffsarten werden angezeigt. * Array wird im Netzwerk hin und her, via Sockets und TCP Pakete verschickt. |

## Planung LB2

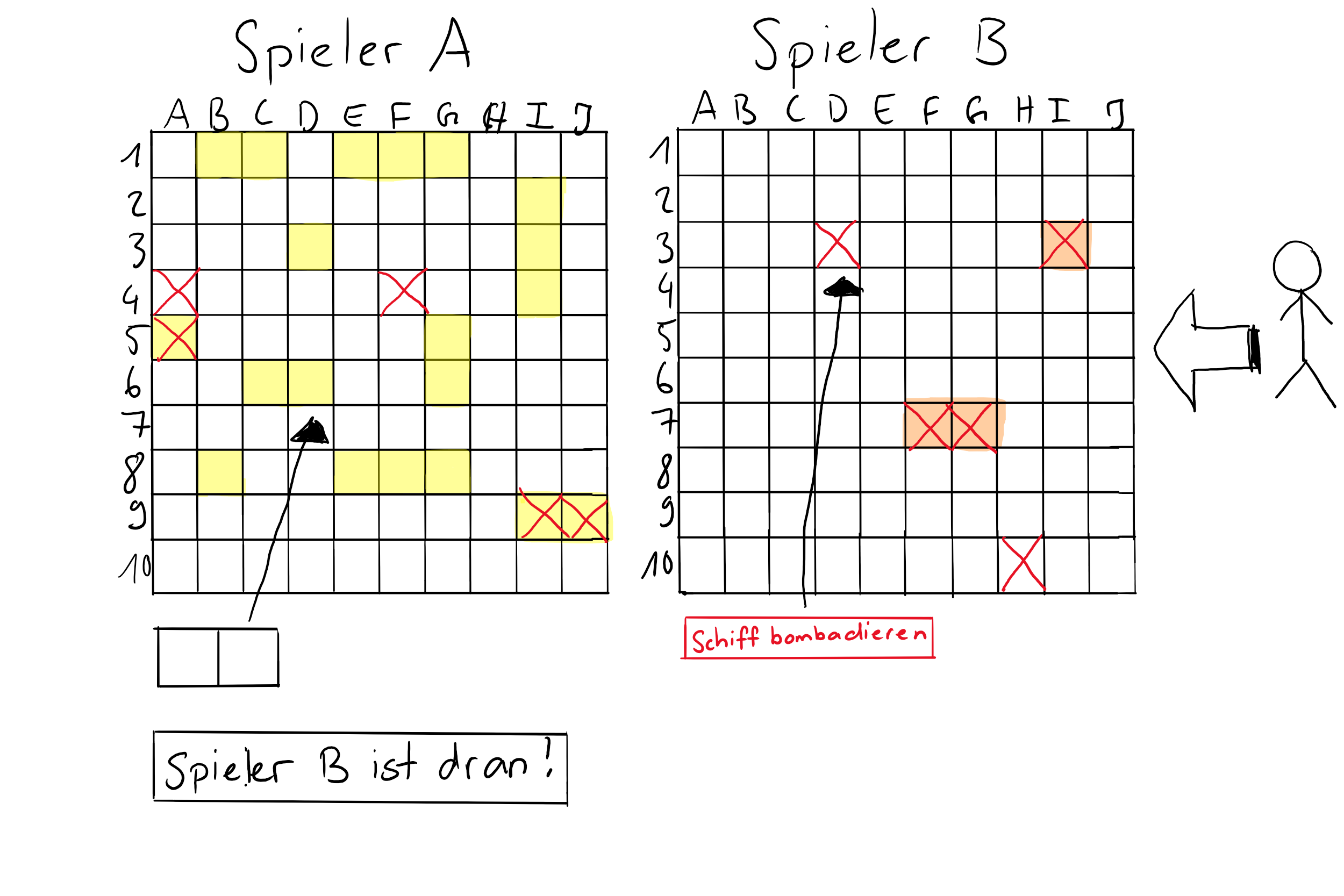
|  |  |  |  |
| --- | --- | --- | --- |
| *MS* | *Tätigkeit / Abgabe* | *Soll-Datum* | *Ist-Datum* |
| A | ProjektstartTeam BildungWahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson | 29.11.2020 |  |
| B | Teamaufgabe 1:Abgabe: Lösungsdesign  (Analyse, Design: Funktionsmodell, UseCase, GUI, Storyboard) | 01.12.2020 |  |
| B2 | Teamaufgabe 2:Abgabe: Testvorschrift und Testfälle | 19.01.2021 |  |
| C | Einzelaufgabe 3:Abgabe Szenario (.zip) mit Inline-Dokumentation, Systemdokumentation (UML Klassen-, Sequenzdiagramm)Fachgespräch Projektabnahme | 26.01.2021 |  |
| C2 | Einzelaufgabe 4:Abgabe: Ausgefüllter Systemtest | 21.01.2021 |  |

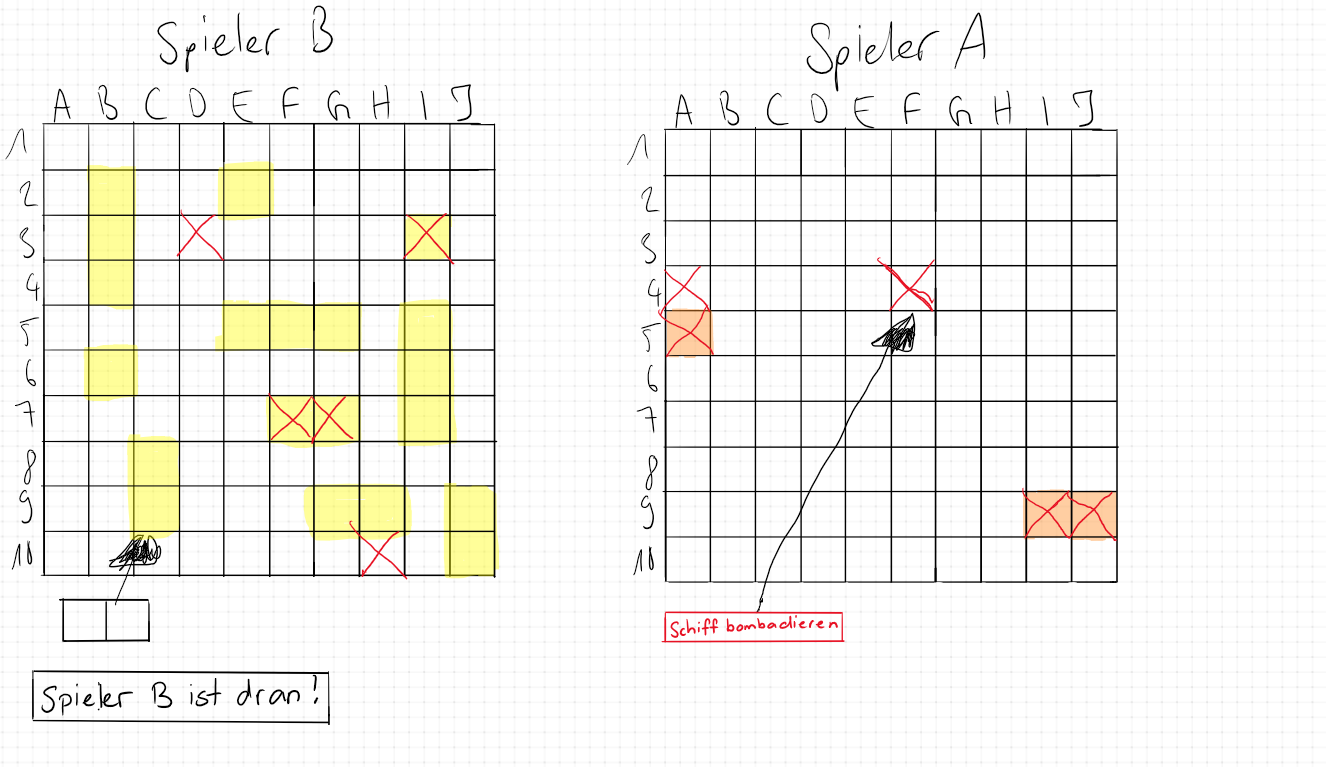
# Lösungsdesign (Meilenstein B: Teamaufgabe 1)

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

## Funktionsmodell

Im Folgenden sind die erwarteten Eingaben und Ausgaben beschrieben / dargestellt:





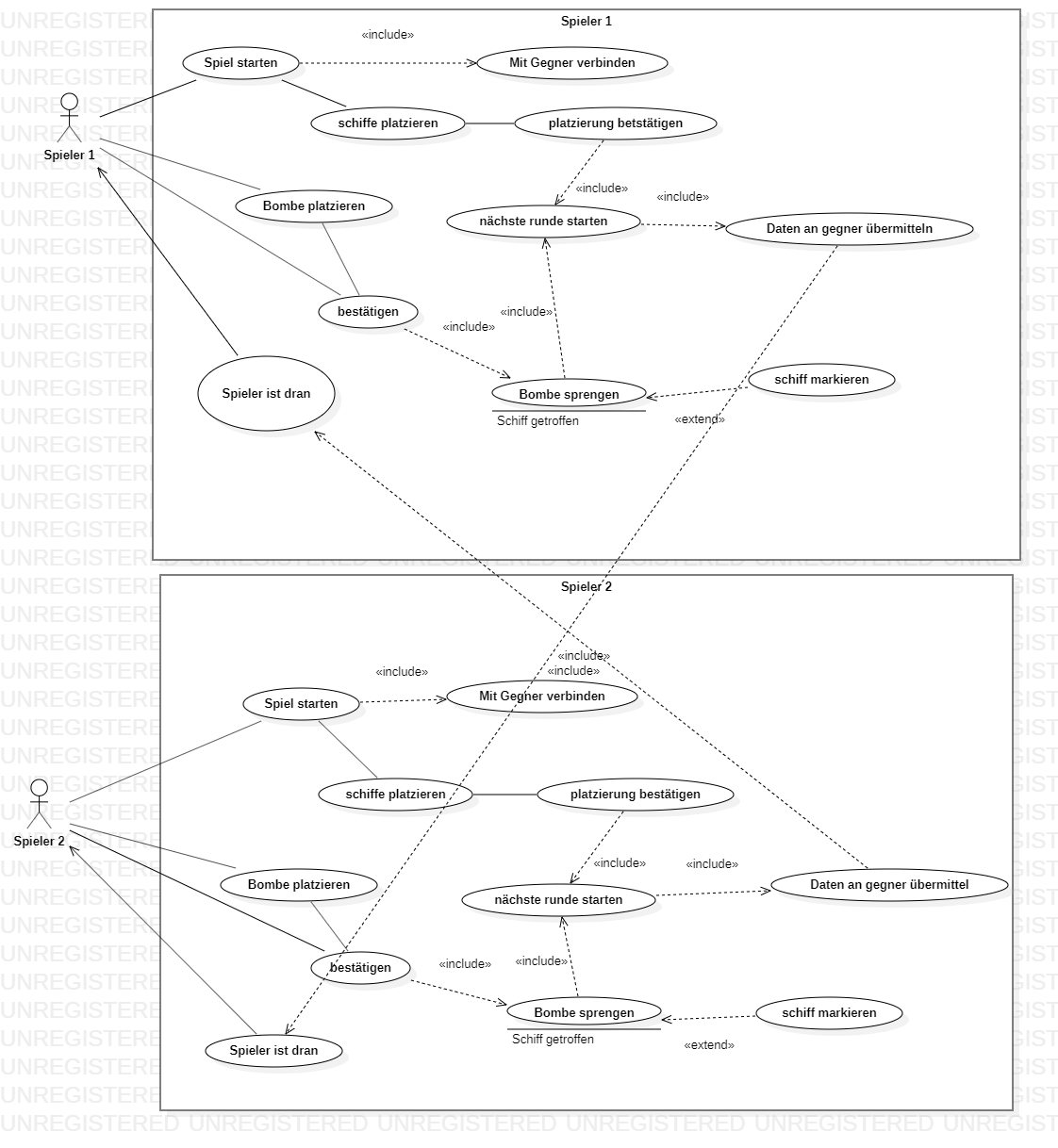
Legende:

* Kreuz: Bombardierte Felder vom Spieler
* Kasten: Schiffe die eingesetzt werden können
* Roter Knopf: Knop nach Auswahl vom Feld bomadieren
* Info: Screen aktueller Spieler anzeige

## Anwendungsfälle (UseCases)

Folgende Anwendungsfälle sind hier detailliert dokumentiert:

...



## Ablauf

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

...

(Storyboard)

|  |  |
| --- | --- |
| Ein Bild, das Tisch enthält.  Automatisch generierte Beschreibung | Spieler kann am Anfang vom Spiel die Schiffe im Feld einsetzen. Er wird drei  Schiffe im Feld einsetzen die zur Verfügung stehen. |
| Ein Bild, das Tisch enthält.  Automatisch generierte Beschreibung | In diesem Infofeld kann man sehen welcher Spieler dran ist. |
| Ein Bild, das Tisch enthält.  Automatisch generierte Beschreibung | Der Spieler kann mit einem Mausklick einen Feld auswählen und danach mit dem Button unten das Feld bombardieren. |
| Ein Bild, das Tisch enthält.  Automatisch generierte Beschreibung | Wenn der Spieler einen Feld getroffen hat wird das Feld eingefärbt |
|  | Ein Spieler hat gewonnen, wenn alle Schiffe getroffen wurden. Wenn der Spieler gewonnen hat kommt eine Meldung das man gewonnen hat, bzw. verloren. |

# Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument   
***M226B\_LB2\_Testvorschrift\_MS-B2.docx***

# Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

Das erstellte Java-Projekt (Greenfoor-Szenario) ist hier detailliert abgelegt:

***M226B\_ Aufgabe\_3\_Szenario\_IhrName.zip***

## Statisches Design: Klassendiagramm

Folgend die statische Struktur des Szenarios

...

(UML Klassendiagramm mit Assoziationen und Kardinalitäten)

## Umfang / Abgrenzung / Änderungen gegenüber Design

Aufgrund unten beschriebener Umstände sind Anpassungen des ursprünglichen Lösungsdesigns gemacht worden:

...

(Umstände / Anpassungen / Veränderungen)

## Funktionalität der Implementation.

Zusätzlich zu der Inline-Dokumentation sind hier folgende Funktionen detailliert beschrieben:

...

(Ausführliche Beschreibung der internen Funktionen   
oder Verweis zum Inline-Kommentar mit JavaDoc! (/\*\* @param @return \*\*/)

## Dynamische Struktur: Sequenzdiagramm

Ein zentraler Ablauf eines UseCases ist im Folgenden dargestellt:

...

(Darstellung eines zentralen Ablaufs mittels Sequenzdiagramm)

**Trace: ...**

*...*

# Bedienungsanleitung (Meilenstein C: individuelle Aufgabe 3)

...

# Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument   
***M226B\_LB2\_Testvorschrift\_MS-C2\_Name.docx***