**Projekt: Mosti**

Anforderungsspezifikation

(Requirement Specification)

[Dokumentstruktur basiert auf RUP „Requirement Specification“]

Inhaltsverzeichnis

[1. Dokumentinformationen 4](#_Toc455612487)

[1.1 Änderungsgeschichte 4](#_Toc455612488)

[2. Einführung (Introduction) 5](#_Toc455612489)

[2.1 Definitionen und Abkürzungen (Definitions, Acronyms, Abbreviations) 5](#_Toc455612490)

[2.2 Referenzen (References) 5](#_Toc455612491)

[2.3 Übersicht (Overview) 5](#_Toc455612492)

[3. Allgemeine Beschreibung (Overall Description) 6](#_Toc455612493)

[3.1 Produkt Perspektive (Product Perspective) 6](#_Toc455612494)

[3.2 Produkt Funktion (Product Functions) 6](#_Toc455612495)

[3.3 Benutzer Charakteristik (User Characteristics) 6](#_Toc455612496)

[3.4 Einschränkungen (Constraints) 7](#_Toc455612497)

[3.5 Annahmen (Assumptions) 7](#_Toc455612498)

[3.7 Use Case Überblick (Use Case Survey) 8](#_Toc455612499)

[4. Spezifische Anforderungen (Specific Requirements) 10](#_Toc455612500)

[4.1 Funktionale Anforderungen (Functionality) 10](#_Toc455612501)

[4.1.1 Funktionale Anforderung F-1: Kassenfunktion 10](#_Toc455612502)

[4.1.2 Funktionale Anforderung F-2: Kalenderfunktionen 10](#_Toc455612503)

[4.1.3 Funktionale Anforderung F-3: Kunden- und Mitarbeiterverwaltung 10](#_Toc455612504)

[4.1.4 Funktionale Anforderung F-4: Lagerbestand 10](#_Toc455612505)

[4.2 Bedienbarkeit (Usability) 10](#_Toc455612506)

[4.2.1 Bedienbarkeitsanforderung U-1: Übersichtlich 10](#_Toc455612507)

[4.2.2 Bedienbarkeitsanforderung U-2: Selbsterklärend 11](#_Toc455612508)

[4.3 Zuverlässigkeit (Reliability) 11](#_Toc455612509)

[4.3.1 Zuverlässigkeitsanforderung R-1: Exakte Berechnung 11](#_Toc455612510)

[4.3.2 Zuverlässigkeitsanforderung R-2: Plausibilität 11](#_Toc455612511)

[4.3.3 Zuverlässigkeitsanforderung R-3: Durchschnittliche Ausfallzeit 11](#_Toc455612512)

[4.3.4 Zuverlässigkeitsanforderung R-4: Fehlersuche 11](#_Toc455612513)

[4.4 Leistung (Performance) 11](#_Toc455612514)

[4.4.1 Leistungsanforderung P-1: Geschwindigkeit 11](#_Toc455612515)

[4.4.2 Leistungsanforderung P-2: Durchsatz 11](#_Toc455612516)

[4.5 Wartbarkeit (Supportability) 12](#_Toc455612517)

[4.5.1 Wartbarkeitsanforderung S-1: Geringe Kopplung 12](#_Toc455612518)

[4.6 Installation 12](#_Toc455612519)

[4.6.1 Installationsanforderung I-1: Download 12](#_Toc455612520)

[4.6.2 Installationsanforderung I-2: Windowskompatibel 12](#_Toc455612521)

[4.7 Schnittstellen (Interfaces) 12](#_Toc455612522)

[4.7.1 Benutzerschnittstelle (User Interface) 12](#_Toc455612523)

[4.7.2 Hardwareschnittstelle (Hardware Interface) 12](#_Toc455612524)

[4.7.3 Softwareschnittstellen (Software Interface) 12](#_Toc455612525)

[4.7.4 Datenbankschnittstelle (Database Interface) 12](#_Toc455612526)

[4.8 Lizenzanforderungen (Licensing Requirement) 12](#_Toc455612527)

[4.9 Verwendete Standards (Applicable Standards) 12](#_Toc455612528)

[5. Use Cases 13](#_Toc455612529)

[5.1. Use Case Diagramm 13](#_Toc455612530)

[5.2. Aktoren & Stakeholders 13](#_Toc455612531)

[5.3. Use Case UC1: Verkauf abwickeln 14](#_Toc455612532)

[5.4. Use Case UC2: Kunde bearbeiten 16](#_Toc455612533)

[5.5. Use Case UC3: Lagerbestand bearbeiten 18](#_Toc455612534)

[5.6. Use Case UC4: Mitarbeiter bearbeiten 19](#_Toc455612535)

[5.7. Use Case UC5: Dienstleistung bearbeiten 20](#_Toc455612536)

[5.8. Use Case UC6: Schichtplan bearbeiten 21](#_Toc455612537)

[5.9. Use Case UC7: Kundentermin bearbeiten 22](#_Toc455612538)

[5.10. Use Case UC8: Admin-Account anlegen 24](#_Toc455612539)

[5.11. Use Case UC9: Passwort ändern 25](#_Toc455612540)

[5.12. Use Case UC10: Systemanmeldung durchführen 26](#_Toc455612541)

[5.13. Use Case UC11: Übersicht über Verkäufe erstellen 27](#_Toc455612542)

[5.14. Use Case UC12: Trester abrechnen 28](#_Toc455612543)

# 1. Dokumentinformationen

## 1.1 Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 09.04.2016 | 1.0 |  | Team 11 |
| 06.07.2016 | 2.0 | kleine Aktualisierungen | Team 11 |

# 2. Einführung (Introduction)

## 2.1 Definitionen und Abkürzungen (Definitions, Acronyms, Abbreviations)

Vgl. separates Dokument (Dateiname: Glossary.docx)

## 2.2 Referenzen (References)

Buch: Craig Larman: UML 2 und Patterns angewendet (2005)

## 2.3 Übersicht (Overview)

Dieses Dokument enthält eine Zusammenfassung aller funktionalen und nicht funktionalen Anforderungen an unsere Mosti-Software. Zusätzlich sind hier die bisher erstellten Use Cases aufgelistet, mit welchen einige Anforderungen spezifiziert worden sind.

# 3. Allgemeine Beschreibung (Overall Description)

## 3.1 Produkt Perspektive (Product Perspective)

Mosti soll als eine Anwendung zur Unterstützung von Mosterei-Betrieben entwickelt werden. Dem Mosterei-Inhaber soll ein Programm zur Verfügung gestellt werden, mit welchem er die einzelnen Bereiche der Mosterei verwalten kann. Dadurch sollen zum einen organisatorische Aufgaben erleichtert werden, wie z. B. Kundenterminplanung, Erstellung von Arbeitsplänen, Verwaltung von Lagerbeständen etc. Zum anderen soll Mosti die Verkaufsabwicklung vereinfachen, indem die einzelnen Kundenaufträge, welche manuell während des Betriebes erfasst werden (z. B. Literzahl, Verbrauchsmaterial etc.), direkt im System eingetragen und verarbeitet werden. Im Gegensatz zu bestehenden Lösungen soll Mosti all diese Funktionen zur Unterstützung des Mosterei-Betriebes vereinen.

## 3.2 Produkt Funktion (Product Functions)

Kernfunktion der Anwendung ist die Unterstützung der Verkaufsabwicklung. Abhängig von den spezifischen Einkaufsparametern (gepresste Literzahl, benötigte Materialien), die vom Mitarbeiter ins System eingegeben werden, berechnet das System automatisch den Preis für den getätigten Einkauf. Zusätzlich wird dem Kunden eine Übersicht über seinen Einkauf zum Ausdruck bereitgestellt.

Für die interne Verwaltung der Mosterei stehen in der Mosti-Software Kalenderfunktionen für die Terminplanung der Kunden und die Verwaltung des Mitarbeiterschichtplans bereit. Der Mosterei-Inhaber hat die Möglichkeit, neue Termine und Schichten zu erstellen sowie bestehende zu bearbeiten beziehungsweise zu löschen.

Eine weitere Funktionalität ist die Speicherung von Kunden-und Mitarbeiterdaten. Um Kunden bei eventuellen Terminänderungen erreichen zu können, werden bei der ersten Terminvereinbarung Kundendaten gespeichert und mit dem entsprechenden Termin verknüpft. Später können diese Daten wieder eingeholt und bearbeitet werden.

Ebenso werden die Daten des Mitarbeiters in Verbindung mit seinem eigenen Account im System abgespeichert und stehen zur Bearbeitung zur Verfügung.

Des Weiteren kann der Mosterei-Inhaber mit der Lagerfunktionalität den aktuellen Lagerbestand einsehen, sowie Dienstleistungen und Produkte zum Sortiment hinzufügen oder wieder entfernen bzw. bearbeiten. Die im Kundeneinkauf verbrauchten Materialien werden automatisch dem Lager abgezogen und so der Bestand laufend aktualisiert.

Dazu kommt die Funktion der Erstellung von Übersichten. Dies gilt sowohl für die Verkäufe an Kunden über die Kassenabrechnung, als auch für die Tresterverkäufe. Dabei kann der Anwender auswählen, ob er die Verkäufe für einen bestimmten Kunden oder für alle Kunden anzeigen lassen will und auch für welchen Tag bzw. Zeitraum er diese Übersicht erstellen lassen möchte.

Zuletzt kann der Tresterpreis festgelegt und Tresterverkäufe erfasst werden.

All diese Funktionen sind nur für angemeldete Benutzer des Systems verwendbar. Hier wird nach dem Account des Mitarbeiters und dem des Mosterei-Inhabers unterschieden, dem ein Administratoraccount zugeteilt wird. Allgemein kann der Administrator für seinen Mosterei-Betrieb spezifische Konfigurationswerte (z. B. Öffnungszeiten) festlegen.

## 3.3 Benutzer Charakteristik (User Characteristics)

Die Zielgruppe für die Mosti-Software sind vor allem Mosterei-Betriebe ohne eigene Verwaltungssoftware beziehungsweise mehreren verschiedenen Programmen, die durch entsprechende Schnittstellen zusammenarbeiten. Da Mosti die wichtigsten Funktionen einer Verwaltungssoftware in sich vereint, dient es als gute Alternative zu mehreren Systemen oder zur händischen und papiergestützten Verwaltung. Durch die einfache Handhabung ist Mosti auch in Betrieben einsetzbar, in welchem die Mitarbeiter nicht viel Zeit auf die Bedienungserlernung des Systems verwenden können und wenig mit der computertechnischen Umsetzung der Dokumentation und Abrechnung vertraut sind.

## 3.4 Einschränkungen (Constraints)

Die geplante Vision von Mosti sieht vor, das Programm auf einem einzigen Computer laufen zu lassen. Dies impliziert, dass die einzelnen Funktionen nicht parallel ausgeführt werden können. So können beispielsweise während den Öffnungszeiten (also wenn die Kassenfunktion verwendet wird) keine neuen Termine erstellt werden.

Des Weiteren muss der Mosterei-Inhaber zur Belegung der einzelnen Schichten die Qualifikation seiner Mitarbeiter kennen, da nicht automatisch vom System überprüft wird, welcher Mitarbeiter für welche Aufgaben geeignet ist.

## 3.5 Annahmen (Assumptions)

Da auch der Mitarbeiterschichtplan mithilfe der Mosti-Software erstellt und eingesehen werden kann, kam die Frage auf, die Mitarbeiterschichtpläne gefiltert nach einzelnen Mitarbeitern anzeigen zu lassen.

Eine weitere Frage ist die Wichtigkeit der Accounts. Das Team hat es als sinnvoll empfunden, einige Funktionen nur dem Mosterei-Inhaber zuzuteilen

**3.6 Abhängigkeiten (Dependencies)**

Da Mosti ein eigenständiges Verwaltungsprogramm ist, ist es auf keine weiteren Programme oder Dienstleistungen angewiesen. Lediglich für die Übersicht über den Einkauf ist eine Verbindung zu einem Drucker notwendig, um dem Kunden eine Zusammenfassung seines Einkaufs und eine Übersicht der Preise und benötigten Materialien mitgeben zu können.

## 3.7 Use Case Überblick (Use Case Survey)

**UC 1: Verkauf abwickeln**

Der Kunde kommt nach dem Press- und Abfüllvorgang an die Kasse. Dort erfasst der Kassierer alle vom Kunden verbrauchten Materialien. Das System berechnet die totale Summe und erstellt eine Übersicht des Verkaufs für den Kunden. Nach der Zahlung durch den Kunden speichert das System die Verkaufsinformationen.

**UC 2: Kunde bearbeiten**

Das System präsentiert dem Mitarbeiter bzw. dem Mosterei-Inhaber eine Übersicht aller Kundendaten. Diese kann der Benutzer nun bearbeiten, löschen bzw. neue Kundendaten zum System hinzufügen. Das System speichert die getätigten Änderungen.

**UC 3: Lagerbestand bearbeiten**

Das System präsentiert dem Mosterei-Inhaber eine Übersicht über die im Lager befindlichen Produkte. Nun kann der Mosterei-Inhaber Produktdaten ändern sowie neue Produkte hinzufügen und löschen. Das System speichert alle getätigten Änderungen.

**UC 4: Mitarbeiter bearbeiten**

Das System präsentiert dem Mosterei-Inhaber eine Übersicht über alle in seinem Betrieb angestellten Mitarbeiter. Die angezeigten Daten kann der Mosterei-Inhaber nun bearbeiten und außerdem Mitarbeiter löschen sowie neue Mitarbeiter anlegen, für die das System einen neuen Account erstellt. Alle getätigten Änderungen werden vom System gespeichert

**UC 5: Dienstleistungen bearbeiten**

Das System präsentiert dem Mosterei-Inhaber eine Übersicht über angebotene Dienstleistung (z. B. Apfelsaft – heiß in Beutel). Nun kann der Mosterei-Inhaber Daten dieser Dienstleistungen ändern sowie neue Dienstleistungen hinzufügen und löschen. Das System speichert alle Änderungen.

**UC 6: Schichtplan bearbeiten**

Das System präsentiert dem Mosterei-Inhaber eine Tagesvorlage des Schichtplans. Der Mosterei-Inhaber trägt hier nun die einzelnen Mitarbeiter in die gekennzeichneten Schichten ein. Das System speichert alle Eintragungen ab.

**UC 7: Kundentermin bearbeiten**

Das System präsentiert dem Mosterei-Inhaber eine Übersicht über die erstellten Termine. Der Mosterei-Inhaber kann nun einzelnen Termin auswählen und bearbeiten sowie Termine löschen und neue Termine erstellen. Das System speichert alle getätigten Änderungen.

**UC 8: Admin-Account anlegen**

Beim ersten Start des Systems an einem Computer ist bereits ein Admin-Account vorhanden. Der Mosterei-Inhaber kann nun persönliche Daten hinzufügen und das Passwort ändern. Das System speichert alle Daten und vergibt den Administrationsstatus an den Account des Mosterei-Inhabers.

**UC 9: Passwort ändern**

Nachdem sich der Mitarbeiter beziehungsweise der Mosterei-Inhaber ins System eingeloggt hat, fordert er die Änderung seines Passworts. Dazu gibt er sein altes Passwort und zweimal das neue Passwort an. Das System prüft die Konsistenz der Eingabe und speichert die Änderungen ab.

**UC 10: Systemanmeldung durchführen**

Das System fordert vom Mosterei-Inhaber beziehungsweise vom Mitarbeiter den Benutzernamen und ein Passwort. Nach der Eingabe prüft das System die Daten auf ihre Richtigkeit und authentifiziert den Mosterei-Inhaber beziehungsweise den Mitarbeiter.

**UC 11: Übersicht über Verkäufe erstellen**

Der Mosterei-Inhaber fordert eine Übersicht von Verkäufen in einem bestimmten Zeitraum an und gibt die gewünschten Parameter ins System ein. Das System erstellt eine Übersicht abhängig von den eingegebenen Parametern.

**UC 12: Trester abrechnen**

Kassierer fordert Berechnung der Trestermenge an. System berechnet aus den tagesaktuellen Verkäufen die Trestermenge und dessen Preis. Kassierer bestätigt Zahlung durch den Kunden und System speichert den Verkauf des Testers ab.

# 4. Spezifische Anforderungen (Specific Requirements)

Ein Mitglied unseres Teams arbeitet selbst in einem Mosterei-Betrieb und ist mit den Problemen eines fehlenden beziehungsweise mehreren, nicht korrespondierenden Verwaltungsprogrammen vertraut. Ebenso hat sie als Mitarbeiterin eine genaue Vorstellung der zu entwickelnden Software. Bei Fragen bezüglich der funktionalen Anforderungen stehen uns verschiedene Mitarbeiter der Mosterei Hemau zur Verfügung.

## 4.1 Funktionale Anforderungen (Functionality)

Die folgenden Anforderungen spezifizieren die Funktionalität der Mosti-Software.

### 4.1.1 Funktionale Anforderung F-1: Kassenfunktion

Die Mosti-Software soll die einfache Berechnung der Endsumme des Kunden unterstützen. Dazu soll im Frame der Kassenfunktion eine Übersicht aller Produkte und Dienstleistungen angezeigt werden, die bei Bedarf markiert werden können und zu der eine dazugehörige Mengenanzahl eingegeben werden kann. Die Mosti-Software soll automatisch die zu entrichtende Summe berechnen. Zusätzlich soll das System alle verkaufsspezifischen Daten in einer Übersicht zum Ausdrucken bereitstellen und die einzelnen Verkäufe kundenspezifisch in der Datenbank speichern.

### 4.1.2 Funktionale Anforderung F-2: Kalenderfunktionen

Für die Verwaltung der Kundentermin und zur Erstellung des Mitarbeiterschichtplans soll in der Mosti-Software eine Kalenderfunktion zur Verfügung stehen. In einer Tagesübersicht soll der Mosterei-Inhaber die Kundentermine, deren Länge vom System berechnet werden, abhängig von der voraussichtlich zu pressenden Litermenge, eingeben oder bearbeiten und wieder löschen können. Analog soll die Mosti-Software Masken bereitstellen, damit der Mosterei-Inhaber für jede Tagesschicht genau drei Mitarbeiter eintragen kann beziehungsweise diese wieder bearbeiten kann.

### 4.1.3 Funktionale Anforderung F-3: Kunden- und Mitarbeiterverwaltung

Die Mosti-Software soll Möglichleiten bereitstellen, um Kundendaten und Mitarbeiterdaten speichern und verwalten zu können. Dabei müssen jeweils Name, Adresse und Telefonnummer gespeichert werden. Mosti soll all diese Daten in einer Datenbank speichern und sie zur Bearbeitung durch den Mosterei-Inhaber oder einen Mitarbeiter freigeben.

### 4.1.4 Funktionale Anforderung F-4: Lagerbestand

Die Mosti-Software soll den Bestand des Lagers verwalten und selbstständig aktualisieren. Der Mosterei-Inhaber soll neue Produkte zum Lager hinzufügen können sowie bestehende Produkte ändern oder löschen. Das System soll den Mosterei-Inhaber informieren, wenn ein Produkt aus dem Lager eine gewisse Mindestmenge unterschreitet.

**4.1.5 Funktionale Anforderung F-5: Account**

Alle aufgeführten Funktionalitäten sollen in der Mosti-Software nur mit gültigem Account durchgeführt werden können. Das System soll durch die Anmeldung zu einem Account sicherstellen, dass nur authentifizierte Personen das System nutzen. Dabei soll Mosti dem Mosterei-Inhaber einen Administratoraccount zuteilen, der Zugriff auf alle Funktionalitäten hat, und den Mitarbeitern nur einen Account mit beschränktem Zugriff.

## 4.2 Bedienbarkeit (Usability)

Die folgenden Anforderungen spezifizieren die Bedienbarkeit der Mosti-Software.

### 4.2.1 Bedienbarkeitsanforderung U-1: Übersichtlich

Die Mosti-Software soll so übersichtlich gestaltet sein, dass alle Funktionen nach einer 10-minütigen Demonstration von allen Mitarbeitern bedient werden können. In einem User Acceptance Test sollen mindestens 95% der Tester die Software als „übersichtlich“ oder „sehr übersichtlich“ bewerten.

### 4.2.2 Bedienbarkeitsanforderung U-2: Selbsterklärend

Jede einzelne Funktion (Kundenverwaltung, Kassenfunktion, Lagerverwaltung etc.) soll in einem eigenen Frame dargestellt werden. Zur Navigation zwischen den einzelnen Funktionen soll dem Mitarbeiter ein Hauptmenü zur Verfügung stehen. Die einzelnen Unterfunktionen des Frames sollen in Buttons oder einem weiterem Menü dargestellt werden und keine Möglichkeit bieten, in andere Hauptfunktionalitäten zu wechseln. In einem User Acceptance Test  sollen mindestens 95% der Tester die Software als „selbsterklärend“ beurteilen.

**4.2.3 Bedienbarkeitsanforderung U-3: Lesbar**

Die Aufmachung der Software soll so gestaltet sein, dass Texte von 1m Entfernung gelesen werden können. Die Farbgestaltung soll auf ein Minimum reduziert werden, das heißt bis auf die farbigen Buttons und dem Hintergrundbild sollen keine zusätzlichen Farben verwendet werden.

## 4.3 Zuverlässigkeit (Reliability)

Die folgenden Anforderungen spezifizieren die Zuverlässigkeit der Mosti-Software*.*

### 4.3.1 Zuverlässigkeitsanforderung R-1: Exakte Berechnung

In 99,9% der Fälle soll die Mosti-Software den Preis für den Kundeneinkauf richtig berechnen. Bei fehlenden Eingaben, die für die adäquate Berechnung nötig sind, soll das System durch eine Plausibilitätsprüfung auf fehlende Eingabe hinweisen und erst bei vollständiger Informationseingabe den Einkaufspreis berechnen.

### 4.3.2 Zuverlässigkeitsanforderung R-2: Plausibilität

In 99,9% der Fälle soll die Mosti-Software schon während der Eingabe eine Plausibilitätsprüfung durchführen und den Benutzer auf eventuelle Fehler hinweisen. So soll beispielsweise eine Meldung erscheinen, wenn das Geburtsdatum eines Kunden in der Zukunft liegt, ein Mitarbeiter in mehrere Schichten gleichzeitig eingeteilt ist oder der Preis für ein Produkt einen negativen Wert besitzt.

### 4.3.3 Zuverlässigkeitsanforderung R-3: Durchschnittliche Ausfallzeit

Die durchschnittliche Zeit zwischen zwei Systemausfällen soll mindestens 30 Tage betragen.

### 4.3.4 Zuverlässigkeitsanforderung R-4: Fehlersuche

Bei einem Ausfall der Mosti-Software soll das Programm beim Neustart immer den zuletzt gespeicherten Zustand übergeben.

**4.3.5 Zuverlässigkeitsanforderung R-5: Backup**

Zur Sicherung der Daten soll das System den Benutzer in einem von ihm gewählten Zeitraum regelmäßig an ein Backup erinnern, welche er innerhalb der Software durchführen kann.

## 4.4 Leistung (Performance)

Die folgenden Anforderungen spezifizieren die Leistung der Mosti-Software

### 

### 4.4.1 Leistungsanforderung P-1: Geschwindigkeit

Die Anzahl aller gespeicherten Datensätze (bis zu 5000 Kundeneinträgen) soll die Performance des Systems für den Benutzer nicht merklich beeinflussen. In einem User Acceptance Test sollen mindestens 95% der User die Software als „schnell“ oder „sehr schnell“ beurteilen.

### 4.4.2 Leistungsanforderung P-2: Durchsatz

Die Mosti-Software soll alle der Funktionen innerhalb von 10s ausführen

## 4.5 Wartbarkeit (Supportability)

Die folgenden Anforderungen spezifizieren die Wartbarkeit der Mosti-Software.

### 4.5.1 Wartbarkeitsanforderung S-1: Geringe Kopplung

Durch eine geringe Kopplung der Softwareklassen soll sichergestellt werden, dass die Mosti-Software um weitere Funktionalitäten erweitert werden kann, ohne die bisherige Implementierung negativ zu beeinflussen. Dies kann bereits beim Zusammenfügen der einzelnen Codesegmente geprüft werden.

## 4.6 Installation

Die folgenden Anforderungen spezifizieren die Installation der Mosti-Software.

### 4.6.1 Installationsanforderung I-1: Download

Die Mosti-Software soll nach Fertigstellung des Programms auf einer Website zum Download zur Verfügung stehen.

### 4.6.2 Installationsanforderung I-2: Windowskompatibel

Die Mosti-Software soll mit allen Windows-Systemen ab Version 7 kompatibel sein.

## 4.7 Schnittstellen (Interfaces)

### 4.7.1 Benutzerschnittstelle (User Interface)

Die einzige Schnittstelle der Mosti-Software zum Benutzer ist der Computer, auf welchem die Software installiert wird.

### 4.7.2 Hardwareschnittstelle (Hardware Interface)

Um die Zusammenfassung der Kundeneinkäufe ausdrucken zu können, benötigt die Mosti-Software eine Schnittstelle zu einem Schwarz-Weiß-Drucker. Die Einkaufsübersicht soll im DIN A4 Format ausgedruckt werden.

### 4.7.3 Softwareschnittstellen (Software Interface)

Da Mosti ein eigenständiges Verwaltungsprogramm ist, sind keine Schnittstellen zu anderen Systemen nötig.

### 4.7.4 Datenbankschnittstelle (Database Interface)

Die Mosti-Software braucht zur Speicherung seiner Datensätze eine Datenbank und somit eine Schnittstelle zu einem ODBC-Treiber. Die Datenbanksoftware mit Treiber muss vor dem Download auf dem entsprechenden Computers zur Verfügung stehen.

## 4.8 Lizenzanforderungen (Licensing Requirement)

Da die Mosti-Software auf keine anderen Komponenten zugreift, werden auch keine Lizenzen benötigt.

## 4.9 Verwendete Standards (Applicable Standards)

Verwendung der Coding Conventions (vgl. separates Dokument, Dateiname: Coding Conventions Team 11.odt), orientiert an der Google Java Convention

# 5. Use Cases

## 5.1. Use Case Diagramm

*Anmerkung: zur besseren Übersicht wurde der Primärakteur Mitarbeiter auf der rechten Seite abgebildet, obwohl es sich hierbei nicht um einen <<supporting actor>> handelt.*

## 5.2. Aktoren & Stakeholders

* Mosterei-Inhaber: Möchte eine einfache Verwaltung der Kunden, seiner Mitarbeiter und der Terminplanung, möchte effektive Unterstützung des Arbeitsablaufes
* Mitarbeiter: Möchte einfach und schnelle Bedienung der Software, möchte als einziger Zugriff auf seinen Account haben, möchte Einsicht in seinen persönlichen Schichtplan erhalten
* Kassierer: Möchte eine einfache Bedienung der Kassenfunktion, möchte flexible Handlungsmöglichkeiten bei individuellen Kundeneinkäufen
* Kunde: Möchte, dass seine Daten vertrauenswürdig behandelt werden, möchte den aktuellen Preis für die gekauften Produkte zahlen und eine Übersicht über seinen Einkauf erhalten, möchte, dass die Terminvergabe und der Einkauf schnell und einfach abgewickelt werden können
* Mosterei-Anlagen-Firma: Möchte Daten über die Auslastung seiner Maschinen sammeln und dokumentieren
* Tresterkäufer: Möchte, dass die Litermengen richtig berechnet und dokumentiert werden, um die exakte Menge an Pressrückständen zu kaufen und zu bezahlen

# 5.3. Use Case UC1: Verkauf abwickeln

**Primary Actor:** Kassierer

**Stakeholders and Interests:**

* Kassierer: Will übersichtliche Darstellung der angebotenen Produkte. Will, dass Abwicklung eines Verkaufs kundenspezifisch erfolgt und nach Abschluss automatisch gespeichert wird. Will eine Kundenauflistung nach aktuellem Terminplan. Will einfache und schnelle Eingabe der einzelnen Kundeneinkäufe und korrekte Ermittlung der Rechnungssumme für die Kunden. Will automatische Berücksichtigung von Kunden-Besonderheiten (z.B. Rabatt) und Aktualisierung der Lagerbestände nach jedem Verkauf von Materialien.
* Kunde: Will schnelle und korrekte Ermittlung der Rechnungssumme. Will persönliche Einkaufsübersicht in Form eines Belegs direkt nach dem Einkauf (insbesondere mit Angaben über die erzeugte Literzahl, Berücksichtigung kundenspezifischer Besonderheiten wie z.B. Rabatt). Will vertrauliche Behandlung der erfassten Daten.
* Mosterei-Inhaber: Will korrekte, schnelle Verkaufsabwicklung durch schnelle und übersichtliche Eingabe. Will Speicherung der abgeschlossenen Einkäufe (mind. der verkauften Literzahl) für spätere Auswertungen oder auch Kundenrückfragen. Will Aktualisierung der Lagerbestände nach jedem Verkauf von Materialien.
* Jäger: Will korrekte Erfassung und Speicherung der an einem Tag verkauften Literzahlen (Gesamttages-Literzahl zur Berechnung des Rechnungsbetrags für den Trester).
* Mosterei-Anlagen-Firma: Will korrekte Erfassung und Speicherung der erzeugten Literzahlen in einem bestimmten Zeitraum, um Rückschlüsse auf die Funktionalität der Maschinen ziehen zu können.

**Preconditions:** Kassierer ist eingeloggt/authentifiziert. Aktueller Tages-Terminplan liegt vor.

**Postconditions:** Verkauf ist korrekt abgewickelt, abgeschlossen und gespeichert. Ausdruckbare Auflistung des jeweiligen Kundeneinkaufs wurde erstellt. Lagerbestände wurden aktualisiert

**Main Success Scenario:**

1. Kassierer wählt einen Kunden aus der tagesaktuellen Terminliste aus.
2. System zeigt die Einkäufe-Eingabe-Maske für diesen Kunden an.
3. Kassierer trägt die Anzahl der vom Kunden verbrauchten Materialien ein (incl. Literzahl und Abfüllweise).
4. System bildet die Summe der Kosten für die aktuell eingetragenen Produkte.

*Schritte 3-4 wiederholen sich solange, bis der Kassierer alle der vom Kunden zu erwerbenden Produkte in der Eingabemaske erfasst hat.*

1. Kassierer schließt die Gesamt-Kostenermittlung endgültig ab.
2. System erstellt eine Auflistung des Kundeneinkaufs und stellt sie zum Ausdruck bereit.
3. Kassierer bestätigt die Zahlung durch den Kunden.
4. System markiert den Kunden als abgeschlossen, speichert die Verkaufsinformationen und aktualisiert die Lagerbestände.

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

* 1. Kassierer startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
  2. System rekonstruiert zuletzt gespeicherten Zustand.

1a. Aktueller Kunde ist trotz Termin nicht in der Terminliste vorhanden.

* 1. Kassierer fordert Zusatzmaske an.
  2. System zeigt erweiterbare Zusatzkunden-Liste an
  3. Kassierer prüft, ob Kunde in der Kundendatenbank registriert ist.

3a. Kunde ist in der Kundendatenbank registriert

3b. Kunde ist nicht in der Kundendatenbank registriert

1. UC2: Kunde bearbeiten ausführen.

* 1. Kassierer fügt Kunden zur Zusatzkunden-Liste hinzu und wählt den Kunden aus.

3a. Gesuchtes Produkt ist nicht in der Eingabemaske verfügbar.

1. UC3: Lagerbestand bearbeiten ausführen.
2. System zeigt aktualisierte Eingabemaske an.

**Special Requirements:**

* Schriftgröße mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.

**Technology/Data Variations List:**

* Ausdruck der Kostenzusammenstellung/des Belegs mittels Drucker.
* Speicherung der Verkäufe in Datenbank.

**Frequency of Occurrence:** 50-80-mal am Tag, bis zu 10-mal pro Stunde.

**Open Issues:**

* Einbezug von kundenspezifischer Begünstigungen (Verknüpfung mit Kundenverwaltung)
* Extra Abrechnung für Tresterkäufer – mit nur Literzahl und entsprechender Preisumrechnung?

## 5.4. Use Case UC2: Kunde bearbeiten

**Primary Actor:** Mosterei-Inhaber oder Mitarbeiter

**Stakeholders and Interests:**

* Mosterei-Inhaber: Will einen Kunden schnell und einfach hinzufügen, bearbeiten oder löschen. Will Übersicht über Kunden, welche seine Leistungen in Anspruch nehmen.
* Mitarbeiter: Will bestimmte Kundendaten schnell und einfach einsehen (z. B. die Telefonnummer im Falle eines Nichterscheinens trotz Termins). Will einen Kunden schnell und einfach hinzufügen, bearbeiten oder löschen.
* Kunde: Will, dass seine Daten korrekt abgespeichert sind und vertrauenswürdig behandelt werden. Will, dass auf seinen Wunsch hin die Daten geändert werden.

**Preconditions:** Mosterei-Inhaber oder Mitarbeiter ist eingeloggt/authentifiziert.

**Postconditions:** Ein neuer Kunde wurde erstellt.

**Main Success Scenario:**

1. Mosterei-Inhaber oder Mitarbeiter fordert Erstellen eines neuen Kunden.
2. System fordert Name, Vorname, Adresse, Telefonnummer des Kunden.
3. Mosterei-Inhaber oder Mitarbeiter gibt geforderte Daten ein.
4. System legt neuen Kunden an.
5. System präsentiert aktualisierte Übersicht über die ersten dreißig, nach Nachnamen sortierten Kunden.

*Diese Ansicht präsentiert das System jeweils beim Eintritt in die Kundenverwaltung.*

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mosterei-Inhaber oder Mitarbeiter startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
2. System rekonstruiert zuletzt gespeicherten Zustand.

1a. Mosterei-Inhaber oder Mitarbeiter ändert Name, Vorname, Adresse und/oder Telefonnummer

eines Kunden:

1. System speichert die neuen Änderungen.

1b. Mosterei-Inhaber oder Mitarbeiter löscht die Daten eines Kunden:

1. System erfragt Bestätigung des Löschvorgangs.

2. Mosterei-Inhaber oder Mitarbeiter bestätigt Löschvorgang.

3. System speichert die neuen Änderungen.

**Special Requirements:**

* Schriftgröße mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90 % der Fälle soll die Übersicht in weniger als 10 Sekunden geladen sein.

**Technology and Data Variations List:**

* Speicherung der Kundendaten in Datenbank.
* Verbindung zur Datenbank, um geänderte Kundendaten abzuspeichern.

**Frequency of Occurence:** Mehrmals pro Stunde.

**Open Issues:**

* Sollen Kundendaten nach bestimmter Zeit gelöscht werden?
* Müssen Kunden immer angelegt sein, um einen Einkauf zu tätigen?

## 5.5. Use Case UC3: Lagerbestand bearbeiten

**Primary Actor:** Mosterei-Inhaber

**Stakeholders and Interests:**

* Mosterei-Inhaber: Will klare Übersicht über aktuelle Lagerbestände. Will Produkte schnell und einfach in der Lagerübersicht hinzufügen, bearbeiten oder löschen. Will Meldung/Warnung, falls bestimmte Untergrenze eines Produkts unterschritten wird.
* Kunde: Will stets gewünschte Produkte zur Verfügung haben.

**Preconditions:** Mosterei-Inhaber ist eingeloggt/authentifiziert.

**Postconditions:** Ein neues Produkt wurde dem Lager hinzugefügt.

**Main Success Scenario**

1. Mosterei-Inhaber fordert Erstellen eines neuen Produktes.
2. System fordert Produktname, Einzelverkaufspreis für Produkt, aktuell zu Verfügung stehende Produktmenge und Untergrenze für Warnhinweis an.
3. Mosterei-Inhaber gibt geforderte Daten ein.
4. System legt neues Produkt an und speichert die Produktdaten.
5. System präsentiert eine Übersicht über die Produkte.

*Diese Ansicht präsentiert das System jeweils beim Eintritt in die Lagerverwaltung.*

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mosterei-Inhaber startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
2. System rekonstruiert zuletzt gespeicherten Zustand.

1a. Mosterei-Inhaber ändert Produktname, Einzelverkaufspreis, Produktmenge und/oder Unter-  
 grenze des Produkts:

1. System speichert die neuen Änderungen.

1b. Mosterei-Inhaber löscht ein Produkt:

1. System erfragt Bestätigung des Löschvorgangs.
2. Mosterei-Inhaber bestätigt Löschvorgang.
3. System löscht Produkt und speichert neue Änderungen.

**Special Requirements:**

* Schriftgröße mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90 % der Fälle soll die Übersicht in weniger als 10 Sekunden geladen sein.

**Technology/Data Variation List:**

* Speicherung der Lagerbestände in einer Datenbank.

**Frequency of Occurrence:** Ein- bis zweimal im Monat.

**Open Issues:**

* Aktualisierung in Zusammenhang mit Kassenfunktion
* Meldungen?

## 5.6. Use Case UC4: Mitarbeiter bearbeiten

**Primary Actor:** Mosterei-Inhaber

**Stakeholders and Interests:**

* Mosterei-Inhaber: Will neue Mitarbeiter schnell und einfach erstellen oder bearbeiten. Will eine Übersicht über seine Angestellten samt seiner Kontaktdaten, um ihn bei Bedarf kontaktieren zu können.
* Mitarbeiter: Will, dass seine Daten korrekt abgespeichert werden. Will Zugriff auf seine Daten, um sie bei Bedarf ändern zu können.

**Preconditions:**Mosterei-Inhaber ist im System eingeloggt/authentifiziert.

**Postconditions:**Ein neuer Mitarbeiter wurde erstellt.

**Main Success Scenario:**

1. Mosterei-Inhaber fordert Erstellen eines neuen Mitarbeiters.
2. System fordert Vor- und Nachname, Geburtsdatum, Adresse und Telefon- bzw. Handynummer des Mitarbeiters.
3. Mosterei-Inhaber gibt geforderte Daten ein.
4. System legt neuen Mitarbeiter an.
5. System erstellt einen Account mit Benutzernamen und Passwort für den neuen Mitarbeiter.
6. Mosterei-Inhaber leitet Benutzernamen und Passwort an Mitarbeiter weiter.
7. System präsentiert aktualisierte, nach Nachnamen geordnete Ansicht der Mitarbeiter.

*Diese Ansicht präsentiert das System jeweils beim Eintritt in die Mitarbeiterverwaltung.*

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mosterei-Inhaber startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
2. System rekonstruiert zuletzt gespeicherten Zustand.

1a. Mosterei-Inhaber ändert Vorname, Nachname, Geburtsdatum, Adresse und/oder Telefonnum- mer des Mitarbeiters:

1. System speichert die neuen Änderungen.

1b. Mosterei-Inhaber wählt Mitarbeiter und fordert Löschung an:

1. System fordert Bestätigung des Löschvorgangs.
2. Mosterei-Inhaber bestätigt Löschvorgang.
3. System löscht alle Daten und den Account des Mitarbeiters.

**Special Requirements:**

* Schriftgröße mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90% der Fälle soll die Übersicht in weniger als 10s geladen sein.

**Technology and Data Variations List:**

* Speicherung der Mitarbeiter in einer Datenbank.
* Verbindung zur Datenbank, um geänderte Mitarbeiterdaten abzuspeichern.

**Frequency of Occurence:** Ein- bis zweimal im Monat.

## 5.7. Use Case UC5: Dienstleistung bearbeiten

**Primary Actor:** Mosterei-Inhaber

**Stakeholder and Interests:**

* Mosterei-Inhaber: Will, dass maschineller Aufwand sowie unterschiedlich aufwändige Abfüllweisen berücksichtigt und ebenso berechnet werden können. Will die aktuellen Preise schnell und einfach bearbeiten können.
* Mitarbeiter: Will mit den richtigen/aktuellen Preisen abrechnen.
* Kunde: Will, dass sein Einkauf nach den korrekten/aktuellen Preisen abgerechnet wird.

**Preconditions:**Der Mosterei-Inhaber ist im System eingeloggt/authentifiziert.

**Postconditions:**Eine neue Dienstleistung wurde hinzugefügt.

**Main Success Scenario:**

1. Mosterei-Inhaber fordert Erstellen einer neuen Dienstleistung.
2. System fordert Name der Dienstleistung und den Preis pro Liter für diese Dienstleistung an.
3. Mosterei-Inhaber gibt geforderte Daten ein.
4. System legt neue Dienstleistung an und speichert die Dienstleistungsdaten.
5. System präsentiert eine Übersicht über die ersten zehn, alphabetisch sortierten Dienstleistungen.

*Diese Ansicht präsentiert das System jeweils beim Eintritt in die Dienstleistungsverwaltung.*

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mosterei-Inhaber startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
2. System rekonstruiert zuletzt gespeicherten Zustand.

1a. Mosterei-Inhaber ändert Name der Dienstleistung und/oder den Preis pro Liter für eine Dienstleistung und bestätigt Änderung.

1. System speichert die neuen Änderungen.

1b. Mosterei-Inhaber löscht eine Dienstleistung:

1. System erfragt Bestätigung des Löschvorgangs.
2. Mosterei-Inhaber bestätigt Löschvorgang.
3. System löscht Produkt und speichert neue Änderungen.

**Special Requirements:**

* Schriftgröße mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90% der Fälle soll die Übersicht in weniger als 10s geladen sein.

**Technology/Data Variations List:**

* Speicherung der Dienstleistungen in einer Datenbank.
* Verbindung bzw. Zugriff auf Datenbank.

**Frequency of Occurrence:** Einmal jährlich.

**Open Issues:**

* Aktualisierung in Zusammenhang mit Kassenfunktion

## 5.8. Use Case UC6: Schichtplan bearbeiten

**Primary Actor:** Mosterei-Inhaber

**Stakeholder and Interests:**

* Mosterei-Inhaber: Will den Schichtplan schnell und einfach erstellen bzw. bearbeiten können.
* Mitarbeiter: Will auf seinen individuellen Schichtplan schnell und einfach zugreifen können. Will eine übersichtliche Darstellung seiner persönlichen Schichten.

**Preconditions:** Mosterei-Inhaber ist eingeloggt/authentifiziert.

**Postconditions:** Ein aktueller Schichtplan liegt vor.

**Main Success Scenario:**

1. Mosterei-Inhaber fordert Erstellen eines neuen Schichtplans.

2. System zeigt vorgefertigte Schichtplanmaske.

3. Mosterei-Inhaber trägt Mitarbeiter in entsprechende Schichten ein.

4. Mosterei-Inhaber bestätigt Eingabe.

5. System speichert Änderungen.

6. System präsentiert eine Wochenübersicht des Schichtplans.

*Diese Ansicht präsentiert das System jeweils beim Eintritt in die Schichtplanverwaltung.*

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mosterei-Inhaber startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
2. System rekonstruiert zuletzt gespeicherten Zustand.

 1a. Mosterei-Inhaber fordert Bearbeiten eines aktuellen Schichtplanes an:

1. System zeigt aktuellen Schichtplan an.

2. Mosterei-Inhaber wählt Tag und Schichtart aus.

3. Mosterei-Inhaber ändert die Mistarbeiter der gewählten Schicht und bestätigt.

4. System speichert Änderungen.

1-2a. Mitarbeiter fordert Anzeige des Schichtplans:

1. System präsentiert den Kalender

2a. Mitarbeiter wählt ein Datum aus.

1. System präsentiert den Schichtplan des ausgewählten Tages.

**Special Requirements:**

* Sprache: deutsch.
* Schriftgröße von mindestens 12.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90% der Fälle soll die Übersicht in weniger als 10s geladen sein.

**Technology/Data Variations List:**

* Eine Schnittstelle zu einem Drucker sollte vorhanden sein.

**Frequency of Occurrence:**1-2-mal pro Woche.

**Open Issues:**

* Sollen Schichtpläne vergangener Arbeitstage dauerhaft gespeichert werden?

## 5.9. Use Case UC7: Kundentermin bearbeiten

**Primary Actor:** Mosterei-Inhaber

**Stakeholders and Interests:**

* Mosterei-Inhaber: Will Termine leicht in einer Kalenderübersicht verwalten können (erstellen, ändern, löschen). Will schnelle Übersicht über freie bzw. mögliche Termine.
* Mosterei-Mitarbeiter: Will übersichtlichen Terminplan, mit Kundeninformation (Name, Menge des Obstes).
* Kunde: Will schnellen Service.

**Preconditions:**Mosterei-Inhaber ist im System eingeloggt/authentifiziert. Kalender mit Eingabemaske ist vorhanden.

**Postconditions**: Ein neuer Kundentermin wurde erstellt.

**Main Success Scenario:**

1. Kunde fordert Terminvereinbarung
2. Mosterei-Inhaber gibt geschätzte Obstmenge in das System ein.
3. System berechnet voraussichtliche Terminlänge.
4. Mosterei-Inhaber öffnet die Terminplanung mit Kalenderübersicht.
5. System zeigt momentanen Terminplan an.
6. Mosterei-Inhaber wählt mit Kunden einen freien Tag und passende Uhrzeit.
7. Mosterei-Inhaber sucht Kunden in der Datenbank und verknüpft ihn mit Termin.
8. Mosterei-Inhaber trägt Termin in das System ein und bestätigt.
9. System speichert die Änderungen.
10. System präsentiert eine Übersicht der Tagestermine.

*Diese Ansicht präsentiert das System immer beim Eintritt in die Terminverwaltung.*

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mosterei-Inhaber startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
2. System rekonstruiert zuletzt gespeicherten Zustand.

1a. Mosterei-Inhaber ändert Terminlänge, Terminbeginn und/oder Termindatum.

1. System speichert die neuen Änderungen.

1b. Kunde fordert Terminstornierung:

1. Mosterei-Inhaber wählt entsprechenden Termin aus und fordert Löschvorgang an.
2. System fordert Bestätigung des Löschvorgangs.
3. Mosterei-Inhaber bestätigt Löschvorgang.
4. System löscht gewählten Termin.

**Special Requirements:**

* Sprache: deutsch.
* Schriftgröße von mindestens 12.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90% der Fälle soll das Programm in weniger als 10s geladen sein.
* Belegte Tage sollen rot, teils belegte Tage grün, freie Tage ohne Farbe hinterlegt sein.

**Technology and Data Variations List:**

* Eine Schnittstelle zu einem Drucker sollte vorhanden sein.

**Frequency of Occurrence:**Mehrmals täglich.

**Open Issues:**

* Was für ein Kalender ist sinnvoll für unser Programm (Monatskalender, Tageskalender bzw. mit oder ohne Uhrzeiten)
* Sind nach Auswahl eines Kunden dessen vereinbarte bzw. vergangene Termine einsehbar?
* In welcher Form werden Terminlisten angezeigt – Tagesübersicht/Wochenübersicht?
* Wo wird Jäger eingetragen?

## 5.10. Use Case UC8: Admin-Account anlegen

**Primary Actor:** Mosterei-Inhaber

**Stakeholder and Interests:**

* Mosterei-Inhaber: Will als Einziger Administratorrechte für das System erhalten.

**Preconditions:**Software wurde noch nie auf diesem Computer ausgeführt.

**Postconditions:**Admin-Account wurde auf den Mosterei-Inhaber gemappt.

**Main Success Scenario:**

1. Mosterei-Inhaber startet zum ersten Mal das System mit vorgefertigtem Account und Passwort.
2. System präsentiert Startansicht.
3. Mosterei-Inhaber navigiert zur Mitarbeiterverwaltung und ändert seine persönlichen Daten.
4. System speichert geänderte Daten.
5. Mosterei-Inhaber fordert Passwortänderung an.
6. System öffnet Änderungsmaske.
7. Mosterei-Inhaber gibt altes Passwort und zweimal das neue Passwort ein und bestätigt.
8. System speichert Daten und legt neuen Admin-Account an.

**Extensions:**

7a. Passwörter stimmen nicht überein

1. System signalisiert Fehler und fordert neue Eingabe.

2. Mosterei-Inhaber gibt die Passwörter erneut ein.

**Special Requirements:**

* Schriftgröße von mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.

**Technology/Data Variations List**

* Verbindung zur Datenbank, um neuen Account zu speichern.

**Frequency of Occurence:**Nur einmal.

**Open Issues:**

* Wie wird sichergestellt, dass Admin-Account nur beim ersten Mal erstellt wird?
* Was ist, wenn der Admin sein Passwort vergessen hat?

## 5.11. Use Case UC9: Passwort ändern

**Primary Actor:** Mitarbeiter

**Stakeholder & Interests:**

* Mitarbeiter: Möchte auf als Einziger auf seine Daten zugreifen können, möchte sicheres Passwort.
* Mosterei-Inhaber: Möchte, dass sich seine Mitarbeiter jederzeit im System anmelden können.

**Preconditions:**Mitarbeiter besitzt gültigen Account.

**Postconditions:**Neues Passwort wurde abgespeichert und mit dem Mitarbeiterkonto verknüpft.

**Main Success Scenario:**

1. Mitarbeiter loggt sich im System ein.
2. Mitarbeiter fordert Änderung des Passworts an.
3. System fordert altes und neues Passwort mit Wiederholung an.
4. Mitarbeiter gibt altes einmal und neues Passwort zweimal ein und bestätigt Eingabe.
5. System prüft eingespeichertes und eingegebenes Passwort auf Übereinstimmung, speichert das neue Passwort ab und bestätigt den abgeschlossenen Vorgang an den Mitarbeiter.

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mosterei-Inhaber startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
2. System rekonstruiert zuletzt gespeicherten Zustand.

1.a Mitarbeiter kann sich wegen vergessenen Passwort nicht einloggen

1. Mitarbeiter fordert komplett neues Passwort an.

2. System erfragt Benutzernamen des Mitarbeiters.

3. Mitarbeiter gibt Benutzernamen ein.

4. System erstellt neues Passwort, verknüpft es mit diesem Account und stellt dem Mosterei- Inhaber das neue Passwort zur Verfügung.

5.a Passwörter stimmen nicht überein.

1. System signalisiert Fehler und fordert neue Eingabe.

2. Mitarbeiter gibt Passwörter erneut ein und bestätigt.

**Special Requirements:**

* Schriftgröße von mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90% der Fälle soll das Einloggen nicht länger als 10 Sekunden dauern.

**Technology/Data Variations List:**

* Zugriff auf Datenbank, um dort das neue Passwort zu speichern.

**Frequency of Occurence:**Eher selten.

**Open Issues:**

* Wie wird der Chef über das neue Passwort informiert?

## 5.12. Use Case UC10: Systemanmeldung durchführen

**Primary Actor:** Mitarbeiter oder Mosterei-Inhaber

**Stakeholder and Interests:**

* Mosterei-Inhaber: Will, dass nur berechtigte Personen sich im System anmelden können.
* Mitarbeiter: Will schnelles und unkompliziertes Einloggen.

**Preconditions:** Mitarbeiter oder Mosterei-Inhaber hat einen gepflegten Account.

**Postconditions:** Mitarbeiter oder Mosterei-Inhaber ist eingeloggt.

**Main Success Scenario:**

1. Mitarbeiter gibt seinen Benutzernamen und Passwort ein.
2. System prüft, ob Eingabe mit dem hinterlegten Benutzername bzw. Passwort übereinstimmt.
3. System stellt Übereinstimmung der Daten fest.
4. System authentifiziert Mitarbeiter.

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mitarbeiter oder Mosterei-Inhaber startet das System neu, meldet sich an und fordert

Wiederherstellung des zuletzt gespeicherten Zustandes an.

2. System rekonstruiert zuletzt gespeicherten Zustand.

3a. Benutzername oder Passwort sind falsch:

1. System signalisiert Fehler und fordert erneute Eingabe.

2. Mitarbeiter gibt Daten neu ein.

3. System stellt Richtigkeit der Daten fest.

**Special Requirements:**

* Sprache: deutsch.
* Schriftgröße von mindestens 12.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90% der Fälle soll das Einloggen nicht länger als 10 Sekunden dauern.

**Technology/Data Variations List:**

* Verbindung zu einer Datenbank, um Benutzernamen und Passwort abzufragen.

**Frequency of Occurerence:** Mehrmals pro Tag.

**Open Issues:**

* Soll ein Account gesperrt werden, wenn Passwort öfter falsch eingegeben wird?

## 5.13. Use Case UC11: Übersicht über Verkäufe erstellen

**Primary Actor:** Mosterei-Inhaber

**Stakeholder and Interests:**

* Mosterei-Inhaber: Will eine schnelle Erstellung einer Übersicht über die verkauften Gesamtzahlen bestimmter Verkaufspositionen innerhalb eines von ihm gewählten Zeitraum (bestimmter Tag bzw. von-/bis-Datum-Angabe). Will selber Parameter auswählen (bestimmter oder alle Kunden, Zeit).

**Preconditions:**Der Mosterei-Inhaber ist im System eingeloggt/authentifiziert. Es liegt eine Verbindung zur Datenbank vor, in welcher alle Verkäufe gespeichert sind.

**Postconditions:**Eine Übersicht wurde erstellt.

**Main Success Scenario:**

1. Mosterei-Inhaber fordert Erstellung einer Übersicht an.
2. System fordert die Auswahl bestimmter Filterparameter.
3. Mosterei-Inhaber wählt Filterparameter aus und gibt gewünschten Zeitraum an.
4. System zeigt Übersicht an, in welcher die gewünschten Verkäufe mit der jeweils ermittelten Kosten- und Litersumme über den eingegeben Zeitraum dargestellt werden.

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

1. Mosterei-Inhaber startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.

2. System rekonstruiert zuletzt gespeicherten Zustand.

3a. Eingegebenes Zeitraumende liegt vor dem eingegebenen Zeitraumbeginn:

1. System signalisiert Fehler und fordert erneute Eingabe.
2. Mosterei-Inhaber wiederholt Eingabe.

**Special Requirements:**

* Schriftgröße mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.
* In 90% der Fälle soll die Übersicht in weniger als 10s geladen sein.

**Technology/Data Variations List:**

* Verbindung bzw. Zugriff auf Datenbank, in welcher die Daten über abgeschlossene Verkäufe gespeichert sind.

**Frequency of Occurrence:** Ein- bis zweimal wöchentlich.

**Open Issues:**

* Sollen die gefilterten Daten sortiert werden?

## 5.14. Use Case UC12: Trester abrechnen

**Primary Actor:** Kassierer

**Stakeholders and Interests:**

* Kassierer: Will eine schnelle, korrekte Berechnung des Trester-Preises für die vorliegende Trester-Masse. Will einfache und schnelle Verkaufsabwicklung.
* Mosterei-Inhaber: Will korrekte, schnelle Verkaufsabwicklung durch schnelle Ermittlung des Preises für den Trester. Will, dass Kassierer – falls notwendig – die Liter-Zahl manuell anpassen kann.
* Trester-Kunde (i.d.R. Jäger): Will korrekte und schnelle Berechnung des Trester-Preises gemäß der entsprechenden Trester-Masse. Will schnelle Verkaufsabwicklung.

**Preconditions:** Kassierer ist eingeloggt/authentifiziert. Alle Verkaufsabwicklungen des entsprechenden Tages wurden abgeschlossen und gespeichert. In Terminplan ist ein Kunde aufgeführt, welcher den Trester für den aktuellen Betriebstag abholt.

**Postconditions:** Tresterpreis wurde anhand entsprechender Literzahl korrekt ermittelt. Tresterverkauf wurde erfolgreich abgeschlossen und gespeichert.

**Main Success Scenario:**

1. Kassierer fordert am Ende des Betriebstages die Abrechnung des Tresters an.
2. System berechnet anhand tagesaktueller Gesamt-Literzahl die Kostensumme und zeigt diese in einer Abrechnungsmaske an.
3. Kassierer bestätigt Zahlung durch den Kunden.
4. System markiert Trester-Kunden als abgeschlossen und speichert den Verkauf.

**Extensions:**

\*a. Jederzeit, wenn das System ausfällt:

* 1. Kassierer startet das System neu, meldet sich an und fordert Wiederherstellung des zuletzt gespeicherten Zustandes an.
  2. System rekonstruiert zuletzt gespeicherten Zustand.

2a. Trestermasse erschließt sich nicht korrekt aus der tagesaktuell verkauften Gesamt-Literzahl (z. B. weil Kunden ihre eigenen Trester-Anteile mitgenommen haben)

* + - 1. Kassierer fordert manuelle Bearbeitung der Liter-Zahl, anhand welcher der Tresterpreis berechnet werden soll.
      2. Kassierer korrigiert die Liter-Zahl manuell und bestätigt.
      3. System berechnet Trester-Preis anhand neuer Liter-Zahl und zeigt diese in einer Abrechnungsmaske an.

**Special Requirements:**

* Schriftgröße mindestens 12.
* Sprache: deutsch.
* Lesen von ca. 1 Meter Entfernung soll möglich sein.

**Technology/Data Variations List:**

* Speicherung der Verkäufe in Datenbank.

**Frequency of Occurrence:** 1x pro Betriebstag.

**Open Issues:**

* Soll auch hier eine druckbare Übersicht erstellt werden?