



# TECHNISCHE DOKUMENTATION

Projektarbeit im Rahmen der Veranstaltung Labor für Softwareprojekte und  
Project Skills

von Lea Häußermann, Marvin Rekovsky, Benjamin Steigerwald und Vanessa Neumann

## Inhaltsverzeichnis

Einstieg .....	2
Anmerkung .....	2
Analyse- und Designmodelle .....	3
Klassendiagramm .....	3
Aktivitätsdiagramm .....	4
Data-Dictionary .....	5
Eingesetzte Tools und Werkzeuge .....	8
Entwicklungsumgebung .....	8
Versionsverwaltung .....	8
Projektmanagement .....	8
Eingesetzte Software .....	8
Eingesetzte Hardware .....	9
Definition of Done .....	9
Daily-Scrum .....	10
Sprint Reviews .....	23
Sprint 1 – Review am 13.04.2017 .....	23
Sprint 2 - Review 2 am 27.04.2017 .....	24
Sprint 3 - Review am 12.05.2017 .....	25
Sprint 4 - Review am 29.05.2017 .....	26
Sprint 5 - Review am 12.06.2017 .....	27
Sprint 6 - Review am 26.06.2017 .....	28

## Einstieg

In diesem Projekt bestand die Aufgabe darin, für den Kunden CeGaT GmbH, die gesamte Primerdokumentation in Form einer Android-Applikation zu digitalisieren. Die Problemstellung des Kunden bestand darin, dass keine Möglichkeit vorhanden war Primer in Echtzeit zu tracken.

Im Folgenden sind alle technischen Daten, Informationen, Softwarepakete und Diagramme aufgeführt, die zum Verständnis und zum Einarbeiten in unsere Applikation nötig bzw. hilfreich sind.

Wie zum Beispiel Diagramme, das Data Dictionary, unsere Definition of Done, sowie die Dokumentationen unserer Daily Scrums und die der Sprint Reviews.

Das EntnahmeApp-Entwicklungsteam

## Anmerkung

Dateien und Diagramme, die aufgrund ihrer Größe und Umfangreiche hier nicht eingebaut werden können, werden nur kurz als Bilddatei eingefügt, im Folgenden jedoch noch als separate Dateien hinzugefügt.

Das Klassendiagramm ist dem VisualParadigm Project *App2* unter den Klassendiagrammen mit dem Namen „Klassendiagramm APP“ zu finden. Das Aktivitätsdiagramm befindet sich bei den Aktivitätsdiagrammen unter dem Namen „EntnahmeApp Neu“. In diesem VisualParadigm Project können auch die Fortschritte und die Versionen in der Diagrammerstellung betrachtet werden.





## Data-Dictionary

Das Data-Dictionary liefert eine Übersicht, aller erstellten und programmierten Klassen, deren Funktion, sowie Kommunikation untereinander.

Name der Klasse	Package	Funktion	Kommunikation zu anderen Klassen
ResponseCode	enumResponseCode	Listet ResponseCode auf der angefragt / genutzt werden kann	
CustomObserver	serverAPI	Interface zur Definition eines individuellen Betrachters	ResponseCode
ListAPI	serverAPI	Interface zur Definition der ListAPI	PickList, PrimerTube
ListImpl	serverAPI	Implementiert die REST-Anfragen bezüglich aller List-Typen	ResponseCode, CustomObserver, PickList, PrimerTube, ListAPI
LoginAPI	serverAPI	Interface zur Definition der LogiAPI	
LoginControllerImpl	serverAPI	Implementiert die REST-Anfragen bezüglich der Nutzer-Authentifizierung	ResponseCode, CustomOberserver, LoginAPI
PrimerAPI	serverAPI	Interface zur Definition der ListAPI	PrimerStatus, PrimerTube,
PrimerImpl	serverAPI	Implementiert die REST-Anfragen bezüglich der Primer	ResponseCode, CustomOberserver, PrimerStatus, PrimerTube, PrimerAPI
Location	model_List	Beinhaltet alle möglichen Standorte eines Primers	
PickList	model_List	Enthält alle bearbeitete Listen eines Tages	
PrimerStatus	model_List	Repräsentiert den Status eines Primers	
PrimerTube	model_List	Enthält die Primer, die als Ausgangspunkt für die Replikation dienen	
Procedure	model_List	Repräsentiert die Verfahren, die in einem Labor durchgeführt werden können	
StorageLocation	model_List	Beinhaltet die Koordinaten der Lagerplätze der Primer	
NewLocation	parsedate	Definiert einen neuen Standort eines Primers. Zweck dieser Klasse ist, dass der Standort sofort aktualisiert wird, bevor die Liste neu geladen wird	

User	parsedate	Definiert einen Benutzer mit allen nötigen Informationen bezüglich der Nutzer-Authentifizierung	
ListAdapterGathered-Primer	adapter	Konvertiert eine Liste von PrimerTube - Objekten in der View, die in den Container geladen wurden.	ListImpl, PrimerImpl, PrimerTube, PopReturn
ListAdapterLabor	adapter	Konvertiert eine Liste von PrimerTube - Objekten in der View, die in den Container geladen wurden.	ListImpl, NewLocation, User, PrimerTube
ListAdapterLastSanger	adapter	Konvertiert eine Liste von PrimerTube - Objekten in der View, die in den Container geladen wurden.	ListImpl, PrimerImpl, NewLocation, User, PickList, PrimerTube
ListAdapterPicklist	adapter	Konvertiert eine Liste von PrimerTube - Objekten in der View, die in den Container geladen wurden.	ListImpl, PrimerImpl, User, PickList, PrimerTube
LaborGui	guis	Zeigt die Primer-Tracking-Ansicht im Labor an. Diese Klasse unterstützt auch die Suche der Primer basierend auf Primernamen	CustomObserver, ListImpl, ResponseCode, NewLocation, User, PrimerTube, PopTracking, ListAdapterLabor
LagerRueckgabeGUI	guis	Zeigt die Primer-Rückgabe-Ansicht im Lager an	CustomObserver, ListImpl, PrimerImpl, ResponseCode, User, PrimerTube, PopReturn, ScanActivity, ListAdapterGatheredPrimer
LastProcessed-ListActivity	guis	Zeigt die Ansicht für die zuletzt erfolgreich bearbeitete Roboter-Liste im Labor an	CustomObserver, ListImpl, PrimerImpl, ResponseCode, NewLocation, User, PickList, PrimerTube, PopSanger, ListAdapterLastSanger
LoginActivity	guis	Zeigt den Login-Screen der Applikation an	CustomObserver, LoginControllerImpl, ResponseCode, User
PickListActivity	guis	Zeigt die Primer-Entnahme-Ansicht im Lager an. Es werden die drei verschiedenen Verfahren unterschieden: Sanger-Listen, Manuelle Listen und Sonderlisten	CustomObserver, ListImpl, PrimerImpl, ResponseCode, User, PickList, PrimerTube, PopPicklist, ScanActivity, ListAdapterPicklist

PopPicklist	popup	Zeigt ein PopUp-Fenster an, sobald ein Primer in der Liste angeklickt wurde. Unterstützt die Anforderung von neuen Primern	CustomObserver, PrimerImpl, ResponseCode, User, PrimerStatus, PrimerTube
PopReturn	popup	Zeigt ein PopUp-Fenster an, sobald ein Primer in der Liste angeklickt wurde. Unterstützt das Entfernen von Primern	CustomObserver, ListImpl, PrimerImpl, ResponseCode, User, PrimerStatus, PrimerTube
PopSanger	popup	Zeigt ein PopUp-Fenster an, sobald ein Primer in der Liste angeklickt wurde. Unterstützt die Anforderung von neuen Primern, sowie das ändern des derzeitigen Standorts	CustomObserver, PrimerImpl, ResponseCode, NewLocation, User, PrimerStatus, PrimerTube
PopTracking	popup	Zeigt ein PopUp-Fenster an, sobald ein Primer in der Liste angeklickt wurde. Unterstützt die Anforderung von neuen Primern, sowie das ändern des derzeitigen Standorts	CustomObserver, PrimerImpl, ResponseCode, NewLocation, User, PrimerStatus, PrimerTube
ScanActivity	scan	Verantwortlich für die Erkennung von Barcodes, während der Nutzung eines spezifischen Geräts	



## Eingesetzte Tools und Werkzeuge

### Entwicklungsumgebung

Als Basis für die Entwicklung und Implementierung der Software diente Java Entwicklungsumgebung IntelliJ. Grund für die Entscheidung von IntelliJ waren die mit eingebundenen Built-in tools und unterstützte Frameworks. Essentiell waren zum Beispiel die freie Software git, die zur Versionsverwaltung von Dateien genutzt wird, sowie das Buildtool Gradle. Enthalten waren zudem Maven, Datenbanken Tools und weitere Features, die von anderen Teilkomplexen des Projekts verwendet wurden und somit auch von uns zur erfolgreichen Kommunikation mit den Schnittstellen installiert sein mussten. Ausschlaggebend war die Möglichkeit mit IntelliJ Android Projekte zu entwickeln.

*IntelliJ IDEA 2017.1.4*

### Versionsverwaltung

Zur Protokollierung von Änderungen, Archivierung und Zusammenführung von verschiedenen Ständen, sowie zur Koordinierung des Zugriffs auf das Projekt mit fünf Personen wurde Bitbucket verwendet: *Atlassian Bitbucket v4.3.2*.

### Projektmanagement

Das gesamte Projekt, inklusive Terminfristen, Sprints und anderweitigen Aufgaben wurde mit Jira dokumentiert und gepflegt. Hier wurden die jeweiligen Sprints angelegt, User Stories und Tasks erfasst, sowie Aufgaben zugewiesen.

Des Weiteren wurde Jira genutzt, um in Burndown-Charts das Voranschreiten der einzelnen Sprints und des Projekts im Gesamten betrachten und analysieren zu können.

*Atlassian JIRA Project Management Software (v7.1.9#71013-sha1:1aa0586)*

### Eingesetzte Software

Als Betriebssystem für unsere Applikation wurde Android gewählt, da auch nach Information des Kunden die Applikation in der späteren Nutzung auf Androidgeräten ausgeführt werden sollte.

Verwendet wurde hier, die zurzeit aktuellste verfügbare Androidversion:

*Android 7.1.1 Nougat*

Um die entwickelten Programme erfolgreich ausführen zu können, war eine aktuelle Java Runtime Environment nötig.

Sonstige:

- Android SDK 1.8
- Android API 25
- Android API Level 7

Zur Kommunikation unserer Software mit der entsprechenden Datenbank wurde zudem folgende Tools genutzt:

- Springboot
- Postgres: *pgAdmin III – PostgreSQL Tools*

## Eingesetzte Hardware

Um den Code bzw. die Software zu testen, wurde ein uns zur Verfügung gestelltes Android-Tablet von HTC genutzt.

HTC Google Nexus 9

In Fällen in denen nicht mit dem Tablet getestet wurde, wurde auf einen Emulator zurückgegriffen, der ebenfalls auf das Nexus Tablet mit der Android API 25 ausgelegt war.

## Definition of Done

- Dokumentation vom Code
- Unit-Tests auf grün
- Der Code ist fertiggestellt und im Versionierungssystem eingespielt
- Es wurde ein Code Review durchgeführt
- Coding Guidelines und Standards wurden eingehalten
- Es sind keine kritischen Bugs offen

Definition of Done Overview:

In software, this may a Definition of Done may be: "Done means coded to standards, reviewed, implemented with unit Test-Driven Development (TDD), tested with 100 percent test automation, integrated and documented."

In a services context, it might look something like this: "Done means every task under the User Story has been completed and any work created is attached to the User Story so the Product Owner can review it and make sure it meets his or her expectations."

## Daily-Scrum

Die Daily-Scrums, geben einen Überblick, was bis zu diesem einen dokumentierten Treffen geleistet wurde und welche Punkte bis zum nächsten Treffen noch bearbeitet werden sollten. Die Dokumentation der Team-Treffen gibt allgemein einen Überblick, über die Überlegungen und Ansätze zum Vorgehen der Entwicklung unserer Entnahme-App.

### Treffen am 16.03.2017

- Aufgabenstellung erhalten
  - o Labor benötigt Primer, um eine Untersuchung durchzuführen. Diese werden momentan von einem Mitarbeiter zu Beginn des Arbeitstages aus dem Lager geholt. Dazu bekommt er eine Liste, welche Primer benötigt werden und diese holt er dann. Auf einer Papierliste abhaken.
    - Risiko, dass ein falscher Primer entnommen wird, soll durch die Entnahme-App minimiert werden
  - o Dazu wird von der Gruppe Entnahmeliste die Papierliste digitalisiert
  - o Unsere Gruppe soll eine mobile Anwendung schaffen, um die Sicherheit zu verbessern, dass der richtige Primer geholt wird.

### Treffen am 17.03.2017

- Grundsätzlich besprochen, wie ein Mock-Up aussehen könnte, also, wie viele Activities benötigt werden
  - o Vorerst: 2 Activities: Log-In und Display Entnahmeliste
  - o Es soll eine log-Datei erstellt werden, welcher Mitarbeiter wann was entnommen hat (nötig??)
  - o Entnahmeliste kann erst geladen werden, wenn der Log-In erfolgreich war
  - o Wenn das richtige Produkt abgescannt wurde, wird dies auf der Liste vermerkt (Haken oder ausgrauen) und wird nach unten auf der Liste gesetzt
    - Wenn ein Artikel abgescannt wurde, soll dieser aus der DB gelöscht werden, dazu Funktion von Gruppe EntnahmeListe
  - o Wenn Liste abgearbeitet wurde, kann die Log-Datei versendet werden. Wer benötigt diese?
  - o Werden Statistiken benötigt zb bezüglich einer Nachbestellung etc?
  - o Wenn falscher Artikel gescannt wurde, rot markieren, Vibration?

### Treffen am 21.03.2017

- Fragen an Kunde:
  - o Ist es möglich, dass die Primer gescannt werden? Barcodes oder ähnliches drauf?
    - So könnten Fehler bei der Entnahme verhindert werden, da das System einen Abgleich zwischen dem entnommenen Produkt und dem Produkt auf der Liste vornimmt
  - o Welche Berechtigungen haben die Mitarbeiter? Werden diese überhaupt benötigt? Wieviel Mitarbeiter haben Berechtigung? Schnittstelle zur Verwaltung bzw. wer verwaltet die Berechtigungen? Und wer löst Probleme, wenn ein Log-In nicht funktioniert?
  - o Gibt es ein bevorzugtes Gerät für den Scanvorgang?
  - o Anwender bzgl. Usability etc. befragen?
  - o Wie läuft der Vorgang momentan ab?

- Besondere Anforderungen bezüglich Corporate Design gewünscht?
- Ist eine Anmeldung nötig, dass der Mitarbeiter die Liste abrufen kann? Wenn ja, wie soll der Mitarbeiter sich anmelden? Mitarbeiternummer, Name, Passwort?
- Log-Datei nötig? An wen soll das gehen? Hier werden auch die Anmerkungen eingetragen, falls zb ein Primer nicht vorrätig/ auffindbar war.
  
- Variable Sortierung der Informationen erwünscht. GUI
- Welche Informationen sollen in der App angezeigt werden? GUI
- Notizen der TA bei Primer Entnahme erwünscht?
- Gibt es ein Verfallsdatum der Primer?
- Was Passiert aktuell wenn ein Primer Beschädigt wird oder er nicht vorhanden ist.
  
- Fragen an die anderen Gruppen
  - Android ok?
  - Klassendiagramm vor Anforderungsanalyse nötig?

### Treffen am 23.03.2017

- Verantwortlichkeiten:
  - Mock-Up: Vanessa, Lea
  - Jira: Marvin
  - Doku: Vanessa, Lea
  - Android, Scanfunktion: Benni
  - Bitbucket, Git: Alle
- To-do
  - Wer macht UI und wer Schnittstellen
  - Welche Arten von Scannen sind möglich? Barcode, QR-Code, Texterkennung
    - Dem Kunden vorstellen und fragen, was möglich ist
    - Gibt es Alternativen zum Scannen??
  - Aktivitätsdiagramm für Kunde vereinfachen und offener halten. Z.B. Log-in nötig und wie soll scannen gehen?
  - Was passiert, wenn ein Artikel nicht vorhanden ist? Weil DB und Wirklichkeit nicht übereinstimmen
  - Qualitätssicherung, wer braucht welche Info, also wer muss zb die Datenbank updaten, wenn der berechnete Bestand nicht mit Wirklichkeit übereinstimmt → Wer behebt das dann?
  - Mail Hr. Hastenteufel, ob ein Sprint 0 möglich ist in Jira, um Aufgaben vor der eigentlichen Programmierung zu dokumentieren?
- Mock-Up:
  - Visual Paradigm, BalsamiQ, Pencil
  - Verschiedene Schriftgrößen
  - Corporate Design einbinden (Farben, Logo etc.) → Homepage anschauen
- Projektplan grob erstellt (mit Datum)
  - Woche 1: 16., 17.3 und 23., 24.3. → Kick-off und Vorarbeit

## Technische Dokumentation

- Woche 2: bis 31.3. → PO bei Kunde, Anforderungsanalyse
- Woche 3: 3.4. – 7.4. → Sprint 1
- Woche 4: 10.4. – 14.4. → Sprint 1
- Woche 5: 17.4. – 21.4. → Sprint 2
- Woche 6: 24.4. – 28.4. → Sprint 2
- Woche 7: 1.5. – 5.5. → Sprint 3
- Woche 8: 8.5. – 12.5. → Sprint 3
- Woche 9: 15.5. – 19.5. → Integration, Integrationstests, Präsentation beim Kunden (PO) → Sprint 4
- Woche 10: 22.5. – 26.5. → beta-Release, PO bei Kunde, Kundenpräsentation, Usability-Tests → Sprint 4
- Woche 11: 29.5. – 2.6. → Sprint 5
- Woche 12: 5.6. – 9.6. → Sprint 5
- Woche 13: 12.6. – 16.6. → Sprint 6
- Woche 14: 19.6. – 23.6. → Sprint 6
- Woche 15: 26.6. – 30.6. → Final Release, Endpräsentation beim Kunden mit Kunden (alle)

### Treffen am 27.03.2017

- Verschlüsselungen
- Restschnittstelle
- Tomcat laden, Pfad in eclipse eingeben unter preferences → Maven als build struktur für java
- Wlan, verschlüsselung (namen zu schützen?)
- Camunda
- Hygiene
- Logging
- Sicherheitsstandards
- Usability

### Treffen am 06.04.2017

Gespräch mit Product Owner zwecks Treffen bei CeGat

- Da kein WLAN bei CeGat-> Dockingstation für Tablet, um Daten synchronisieren zu können, Scannen mit Pistole
- Verständnis der Primerposition abgeglichen, sowie Ablauf mit der Picklist
- Abgleichen des gleichen Verständnisses der User Stories
- Festlegen der Definition of Done
- Priorisieren der Userstories (Product Backlog)
- Möglicherweise Glossar anlegen?
- Projekt in IntelliJ angelegt

### Organisation

- Tablet wird zu Verfügung gestellt mit Android Version 5.1 (höher nicht möglich)
- Scrum Master festgelegt: Marvin

## Technische Dokumentation

- Scannpistole? (mit Bluetooth)
- Erkenntnis: Touch kann mit handelsüblichen Einweghandschuhen bedient werden!

## Offene Punkte

- Chris fragen, wie es mit Login aussieht
- Wie sind Primer hinterlegt. Barcode in Datenbank hinterlegt?
- Problem mit Gradle in IntelliJ

## Definition of Done

- DoD:
  - Dokumentation vom Code
  - Unit-Tests auf grün
  - Der Code ist fertiggestellt und im Versionierungssystem eingespielt
  - Es wurde ein Code Review durchgeführt
  - Coding Guidelines und Standards wurden eingehalten
  - Es sind keine kritischen Bugs offen

### Definition of Done Overview:

In software, this may a Definition of Done may be: "Done means coded to standards, reviewed, implemented with unit Test-Driven Development (TDD), tested with 100 percent test automation, integrated and documented."

In a services context, it might look something like this: "Done means every task under the User Story has been completed and any work created is attached to the User Story so the Product Owner can review it and make sure it meets his or her expectations."

## Treffen am 07.04.2017

### Besprochene Punkte:

- Nach Aussage Product Owner: Tablet wird organisiert
- Es gibt drei verschiedene Primergrößen (aber nicht relevant für unsere App)
- Jeder Primer hat eine eigene ID, fortlaufend. Gleiche Primer haben trotzdem eine unterschiedliche ID
- ID wird zurück zum Labor geschickt
- Lager hat die Primerdatenbank ( ID, Datum, Lot, Lagerort)
- Entnahmeliste ist nach Lagerort sortiert und Position in Carrier wird festgelegt
- Login wird benötigt
- Schnittstellen festgelegt:
  - Entnahmeliste (pro Liste sollen 128 Primer gespeichert werden. Lagerort und Auftragsnummer wird mitgegeben, da Kisten mit 32 Primerplätzen). Liste muss von uns angefragt werden.

## Technische Dokumentation

- Lager: Direkt nach Scannen bekommt das Lager PrimerID und Auftragsnummer von uns.
- Controlling: Bekommen PrimerID und Datum der entnommenen Primer. Controlling fragt bei uns an. -> Implementierung einer get-Methode

### Done:

- Gruppenordner in Ilias angelegt
- Gruppenname festgelegt: Gargamels

### Offene Punkte:

- Evtl anzeigen der Lot-Nummer in der App (kommt von der Produktion, vergleichbar mit Chargennummer)
- Klassendiagramm anlegen

## Treffen am 10.04.2017

### Besprochene Punkte

- Dokument anlegen mit allen verwendeten Versionen, Treibern etc.
- Wegen Berechtigungssystem nachfragen
- Ist die PrimerID im Barcode gespeichert?
- Es werden zwei UIs benötigt, je nach dem wer sich wo einloggt können Funktionen verfügbar gemacht oder gesperrt werden
- REST-Schnittstelle anschauen (Vorlesung kommenden Donnerstag)
- Abklären wegen Bluetooth für Scannen
- Möglicherweise WLAN mit Accesspoint?
- Postgresversionen mit anderen Gruppen abgleichen
- **Klassendiagramm** nach Fertigstellung der Gruppe Entnahmeliste geben
- **Aktivitätsdiagramm überarbeiten**
- JDBC Treiber notieren

### Done

- Dokumentation der verwendeten Versionen und Treiber

## Treffen am 13.04.2017 Sprint 1 – Review

- Es wurden im vergangenen Sprint Vorbereitungen getroffen
  - Projekt angelegt und auf jedem PC zum Laufen gebracht
  - Benötigte Hardware festgelegt und bestellt
  - Recherche zu verschiedener Software und Schnittstellen
  - Definition of Done
  - Tomcat angeschaut

## Technische Dokumentation

- Noch keine Funktionalitäten implementiert
- Was fehlt noch?
  - Aktivitätsdiagramm überarbeiten und fertigstellen
  - Klassendiagramm erstellen
  - Mock-Up überarbeiten
  - Mit REST arbeiten/ beschäftigen
  - Barcode-Scanner API anschauen

Treffen am 18.04.2017

Gruppenintern

ToDo / Methoden:

- Dropdown Menü mit möglichen Standardbemerkungen für einen Primer
- Abteilung muss eintragbar sein (Rückverfolgung Primer)
- Zu jedem Primer kann Bemerkung eingegeben werden und kann dann mit einem OK-Button bestätigt werden
- Drei verschiedene AppAnsichten abhängig von:
  1. Planmäßige Entnahme für das Labor
  2. außerplanmäßige Entnahme (ein Primer wird in anderer Abteilung benötigt)
  3. Abfrage wo sich Primer befindet (aus dem Labor)

Done:

- Ausweichtermine für Treffen mit Sebastian festgelegt
- Aktivitätsdiagramm angepasst und überarbeitet

Fragen an Chris / ProductOwner:

- Wie ist das Passwort für unseren Gruppenordner in Ilias?
- Geht die Entnahmeliste nach dem Scanvorgang wieder zur der Datenbank zurück?
- Methoden-Interface -> Anzahl Entnahme hochzählen? (an Listengruppe)
- Was sind mögliche Standardbemerkungen bei Primern? (kaputt, nicht vorhanden, ...)
- Wie soll angegeben werden, in welcher Abteilung der Primer landet? (einer kann ja nicht bei mehreren Primern immer angeben müssen, welcher Abteilung er angehört)
- Gibt es eine Mitarbeiterdatenbank? Zwecks Berechtigungen
- Kann man Methode verfassen, bei dir wir Bemerkungen zu einem Primer direkt in die Datenbank schreiben können?

Fragen an Sebastian:

- Sollen zwei verschiedene Apps je nach Standort implementiert werden (zwei Apps mit gleicher Funktionalität) oder eine App mit zwei wählbaren Ansichten
- Wie soll hinterlegt werden, wie oft der Primer schon verwendet wurde
- Gibt es in IntelliJ die Möglichkeit, dass sich Angaben "gemerkt" werden? Z.b Abteilung

Treffen am 20.04.2017 Besprechung mit Gruppe „Entnahmeliste“

Besprochene Punkte:



- Wir kommunizieren nur mit Entnahmeliste
- Entnahmeliste gibt uns Daten der Mitarbeiterdatenbank (MA-DB wird von Labor gemacht)
- Primer werden in Karton ins Labor gebracht und dort in Carrier umgepackt
- Primer müssen ja wieder zurückgebracht werden!
- Primer werden nach Namen gesucht
- Default-Wert für Standortabfrage eines Primers ist: Labor.
- Carrier bleiben im Labor
- Primer können nach Typen angefordert werden
- Zwei Arten von Anfragen möglich:
  - o Normale Primersuche: Frage nach Primertyp, Position des Primers wird uns dann zurückgegeben
  - o Primertyp

Wir bekommen von Entnahmeliste:

- Eine Liste mit allen Primern die entnommen werden sollen, inkrementieren der Primer-Entnahmen – getAll()
- Eine Liste mit Primern mit einer Anzahl, die von dem Nutzer der App vorgegeben wird (Bsp. „erste 30...“) – getDefinedAmountOfPrimer (int amount)
- Bei Login wird uns ein Token zugesendet
- Liste mit allen Primern die der gesuchten Primerart entsprechen
- Logdateien mit Informationen über Primerentnahmen (Wer, wann, was)?
- Im Fall eines leeren, kaputten Primers etc. werden uns die Daten eines neuen zugeschickt
- Status ändern, wenn erfolgreich gescannt – updatePrimerStatus() (ID von Primer an Liste übergeben)

Methoden, die wir implementieren:

- Wir übergeben bei jeder Anfrage Benutzernamen + Passwort (statt Token) und ID
- Primer so und so wurde entnommen und ist auf dem Weg ins Labor -> evtl Methode updatePrimerStatus, wir übergeben der Entnahmeliste eine ID
- Primer werden zurückgelegt -> ID an Liste damit Status zurückgesetzt werden kann ☑Token muss immer mitgesendet werden, wenn wir eine Anfrage stellen
- Es kann nach Primerart gesucht werden, mit ID
- String mit Namen der Abteilung in der sich Primer befindet (außerplanmäßige Entnahme)
- Labor-Ansicht: Primer kann als leer vermerkt werden -> dann aus DB gelöscht
- Neuer Primer soll angefordert werden können: Lager und Labor -> Primerart wird angefragt
- Error messages, wenn Primer:
  - o nicht vorhanden (Übergabe von Token und ID)
  - o kaputt oder zerdeppert (Übergabe von Token, ID und zusätzlicher Message mit genaueren Details)

Done:

- Zuweisung aller Tasks für Sprint 2 in Jira

Offene Punkte:

- Von wo kommen die Barcodes? Hersteller? Werden bei Warenannahme erstellt (also von CeGaT)

- Wenn einzelne Primer entnommen werden, gibt es dann auch Aufträge? Oder werden die wahllos entnommen – Ja, es gibt zu allem einen Auftrag, daher erhalten wir auch hier eine Liste
- Werden Erfahrungswerte bei uns oder im Lager eingetragen? Werden im Lager eingetragen, bei uns wird nur eingetragen, wenn ein Versuch nicht funktioniert hat (wird bei Primer eingetragen) so kann das Controlling sich zu einem fehlgeschlagenen Versuch die Erfahrungswerte einer Primers holen und somit eine Auswertung machen
- Holen Mitarbeiter aus anderen Abteilungen einzelne Primer aus dem Lager? -> woher kommt diese Information? Ja, holen sie, dafür ist eine Methode nötig, aus welcher Abteilung der MA war, der den Primer entnommen hat
- Was hat es mit dem Tracking auf sich? Soll der Primer in verschiedenen Abteilungen gesucht werden oder soll der Fortschritt der Nutzung im Labor beobachtet werden????

### Treffen am 21.04.2017 Besprechung Gesamtarchitektur

- Es gibt eine manuelle Entnahme im Lager, dafür gibt es auch einen Auftrag (Labor)
  - o Es wird dann eine neue Liste erstellt, die wir abrufen können (kommt nicht oft vor, daher ist eine Suche zu einer AuftragsID nicht nötig – es werden dann alle Primer auf der Liste angezeigt, die außerplanmäßig entnommen werden)
  - o Ist eine Liste möglich, in der alle Abteilungen sind, die einen Primer holen können (ausser das Labor)? Wenn ja, könnten wir ein DropDown-Menü einfügen, aus der der MA wählen kann → Abklären, wie viele das sind!
  - o Hier muss bei der Entnahme die Abteilung des entnehmenden Mas eingetragen werden, sonst ist Default Labor
- Suche im Labor immer nach Primernamen → Es gibt mehrere mit dem gleichen Namen, daher den als erstes anzeigen, der als letztes bearbeitet wurde (Abteilung bei Entnahme eingetragen) → Sehr wahrscheinlich, dass der dann auch gesucht wird
- Erfahrungswerte werden von Lager eingetragen
  - o Wir tragen in Labor nur ein, ob ein Versuch mit einem bestimmten Primer fehlgeschlagen ist → Auswertung wird von Controlling gemacht (Schauen, welche Erfahrungswerte hinterlegt sind, wenn ein Versuch nicht funktioniert hat)
- Berechtigungen
  - o Es wird eine csv-Datei erstellt, in der nur der Log-In (Benutzername und Passwort) und eine Gruppe eingetragen ist
  - o Daher müssen wir bei jeder Anfrage (wenn wir der Nutzer erfolgreich eingeloggt ist – also auf der Liste steht) den Benutzernamen und das Passwort mitsenden

### Treffen am 25.04.2017

- log datei als txt-Datei erstellen und speichern → Wir müssen auch eine machen!
- Ersatzplatz für Primer bei Rückgabe nicht nötig, wenn einer entnommen wird, kommt der auch wieder an genau den Platz zurück
  - o Wenn doch schon einer an dem Platz steht, muss der gescannt werden und richtig eingeräumt werden
- Tablet- BS → was ist am weitesten verbreitet (SDK an Tablet anpassen)
- Labor: Primer suchen: Status ändern, wenn es ein einem anderen Büro ist (Scanner auch noch in Labor??)

- Wie oft geht ein Primer in ein nächstes Labor? Sind das viele oder nur einzelne Fegeler/CeGaT
- Wenn es nur einzelne sind, dann manuelle Eingabe
- Aktueller Stand der Dinge bzgl. Android: Nach Guideline für Apps suchen, google demos --> Material Design?
- In Jira keine Unteraufgaben für die User Stories erstellen, sondern normale Tasks und dann mit „relates to“ an der Story zuweisen

### Treffen am 27.04.2017 Sprint 2 - Review

#### Search Activity

<https://developer.android.com/guide/topics/search/search-dialog.html>

#### PopUp Window

<https://developer.android.com/reference/android/widget/PopupWindow.html>

#### ToDo

- Preference Activity für Schriftgröße, Farbe und andere Änderungen
- Im Labor muss nach ID eine Suche möglich sein. Da mehrere Primer den gleichen Namen haben und nach bestimmten Primern einzeln gesucht werden will
- Schnittstellenspezifikation!
- Klassendiagramm überarbeiten

#### Done:

- Tasks für nächsten Sprint festgelegt

#### Besprochene Punkte:

- Bei Login wird Benutzername + Passwort an Entnahmeliste gesendet
- Bei weiteren Anfragen nur der Benutzername

#### Offene Punkte:

- Wie zeigt man eine Activity als PopupWindow an?

### Treffen am 28.04.2017

- Primer sind schon nach Lagerpositionen auf der Liste sortiert (Truhe 1,...), Plätze im Carrier werden auch dementsprechend vergeben → Bei Entnahmeliste nachfragen
  - Carrierposition wird bei der Entnahmeliste gespeichert
  - Im Labor oben sollen die Reihenfolge für die Carrierpositionen angegeben werden, muss bestätigt werden
- Zeitstempel, dass Primer im Lager geholt wurden (Logout des MA) und jetzt im Carrier sind und in die Maschine geschoben werden (gespeichert bei Entnahmelist -> Information an BI Gruppe)
- Kommunikation zwischen beiden Tablets (Lager und Labor) läuft über die Entnahmeliste
- Zwischenspeicher für technische Defekte wird benötigt bspw. Akku leer, Systemabsturz
- Bei Login muss Passwort mit TLS verschlüsselt werden → anderen Gruppen mitteilen

- Certificate Pinning (nice to have) → geht es an richtigen Empfänger?
- Alle Mitarbeiter, die eine Berechtigung für die App haben, dürfen alle Ansichten öffnen
- Einloggen sollte über Username laufen, für Anfrage wird Username und Passwort eingegeben, Rückgabewerte sind true/false sowie Vorname und Nachname und Gruppe zur Dokumentation in der Log-Datei, wir entscheiden dann je nach Gruppe, ob Berechtigung für verschiedene Ansichten besteht
  - o Oder gibt es jetzt doch Token?
- Nice to have: in Laboransicht -> evtl Abfrage nach ProzessID: welche Primer werden in diesem Auftrag verwendet? → Sogar wichtig, um zu sagen, welcher Primer gerade in Carrier ist? Und um zu sagen, bei welchem Primer ein Versuch fehlgeschlagen ist
- **Es gibt keine AuftragsID sondern nur ProzessIDs**
- Ein Auftrag kann mehrere Prozesse beinhalten
- **GUI von planmäßiger und außerplanmäßiger Entnahme zusammenlegen**
- Stattdessen eine GUI mit allen offenen Picklisten -> es kann eine Pickliste ausgewählt werden -> Übergang zur Lager GUI
- Nach Login zusätzliche GUI: Prozessliste wählen (Filter: alle, nur die manuellen etc)
  - o Hier unterscheiden, ob der Primer schon ein festes Ziel hat (dann ist es das Labor), wenn nicht, muss es dann beim Scan eingetragen werden
- Auf einer Pickliste sind mehrere Carrier erfasst (1-32), im Labor oben können in einem Durchgang 4 Carrier genutzt werden
- Schon gescannte Primer werden grau dargestellt aber trotzdem noch anklickbar gelassen, um im Notfall noch eine Bemerkung hinzufügen zu können (Scrollbar einbauen, dass die abgescannten Primer für den Moment nicht mehr sichtbar sind aber wieder angeschaut werden können, wenn man nach oben scrollt)
- **Zum Scannen werden die Kameras der Tablets genutzt**
- **Wir erstellen einen eigenen Barcode auf dem die PrimerID gespeichert wird**
- im Labor müssen noch zusätzlich Bemerkungen zu den Primern eingetragen werden können (Versuch fehlgeschlagen, Primer verloren gegangen, ...)
- Log-Datei nur für uns zum Testen erstellen, ansonsten soll die Entnahmeliste alle unsere Zeitstempel und Informationen speichern
- Hotspot einrichten zum Testen der Apps
- Laboransicht: Ort aktualisieren

### Treffen am 05.05.2017

- Erste Testversuche mit GUI und Primerscanfunktion
- Diskussion und Besprechung der einzelnen Funktionen pro View der APP

### Treffen am 11.05.2017

- Klassendiagramm bearbeitet
- Mockups zusammengefügt und Namensgebung angepasst
  - o Mockups auf Deutsch anlegen, Stichpunkt Internationalisierung ist wenn dann erst später zu beachten
- Schnittstellendefinition und Treffen mit Gruppe „Entnahmeliste“ um Schnittstellen zu besprechen

## Treffen am 12.05.2017 Sprint 3 - Review

Vorstellung des aktuellen Stands vor dem Product Owner:

- Von den vorgenommenen Tasks konnten in diesem Sprint nicht alle umgesetzt werden. Grund dafür waren häufige Änderungen der Gesamtarchitektur nach Treffen im Plenum mit dem Kurs und den Dozenten
  - Erledigt:
    - Schnittstellendefinition
    - GUI für die Primeranzeige
    - Implementierung der Scanfunktion
    - Überarbeitung der Diagramme
    - Mockups
    - Dokumentation aller Treffen in Sprint3
  - Nicht geschafft:
    - Login Berechtigungsabfrage
    - GUI für die Prozessauswahl
    - Manuelle Primersuche
    - Login GUI
    - Menü erstellen
- Im Plenum wurden Begrifflichkeiten und Interpretationen von Namensgebungen im DataDictionary festgehalten
- Der Fokus wurde in diesem Sprint auf die Fertigstellung von Mockups gelegt, da diese beim Beta-Release dem Kunden vorgelegt werden müssen
  - Mockups bzw. Mockup-Präsentation wurde vorgestellt und erklärt
  - Programmierung ist somit in diesem Sprint liegen geblieben
- Vorstellung des Klassendiagramms und der Schnittstellendefinition
- Punkte die in Sprint 4 zusätzlich noch erledigt werden müssen:
  - Schnittstellendefinition muss noch angepasst werden
  - Implementierung der GUI basierend auf den Mockups
- Positiven Aspekte während Sprint3:
  - Das Gruppenklima ist noch gut
  - Gruppeninterne treffen konnten wahrgenommen werden
  - Organisation und Kommunikation in der Gruppe funktionieren gut
- Negative Aspekte während Sprint3:
  - Die Planung und die Architektur ändern sich täglich
  - Uneinigkeiten zwischen den Gruppen, sowie Änderungen bei einzelnen Gruppen von denen man erste Tage später zufällig etwas erfährt
  - Es kann nicht richtig mit der Programmierung und Implementierung begonnen werden
  - Während der Zeiten, die für Programmierung genutzt werden könnte, werden wieder Änderungen besprochen

### Treffen am 18.05.2017

- Schnittstellendefinition abgeschlossen
- Dokumentation aktualisiert (Stand Ilias)
- Besprechung der Schnittstelle mit Gruppe Entnahmeliste
- Überarbeiten des Klassendiagramms
  - o Im Klassendiagramm „Services“ als Controller
- UI bis zum Sprintende fertigstellen

### Treffen am 19.05.2017

- Präsentation der Mockups vor Frau Finkeisen
- Besprechung Vorgehen zur Implementierung
- Absprache mit Gruppe Entnahmeliste bezüglich Schnittstellen
- Aufsetzen eines Dummy Server zum Simulieren (?)
  - o Task im nächsten Sprint
  - o Login, Kommunikation mit Entnahmeliste

### Treffen am 29.05.2017 Sprint 4 – Review

Besprechung des aktuellen Stands im Team:

- Von den vorgenommenen Tasks konnten in diesem Sprint nicht alle umgesetzt werden. Fokus auf Usability-Test-Präsentation bei CeGaT vor Ort:
  - Erledigt:
    - Erstellung der Mockups (zur Visualisierung bei CeGaT)
    - Schnittstelle überarbeitet
    - Klassendiagramm überarbeitet
    - Dokumentation aller Treffen in Sprint4
  - Nicht geschafft:
    - Login GUI
    - Login Berechtigungsabfrage
    - GUI Prozessauswahl
    - Manuelle Primersuche
    - Menü erstellen
- Es wurde nicht allzu viel an den Tasks weitergearbeitet, da auch auf die Rückmeldung des Kunden bezüglich der Mockups gewartet wurde
  - Folgend muss die Implementierung vorangetrieben werden
- Punkte die in Sprint 5 zusätzlich noch erledigt werden müssen:
  - Offene Tasks aus Sprint 4
  - Dummy Server zum Testen anlegen
- Positiven Aspekte während Sprint 4:
  - Das Gruppenklima ist noch gut
  - Gruppeninterne treffen konnten wahrgenommen werden
  - Organisation und Kommunikation in der Gruppe funktionieren gut

- Negative Aspekte während Sprint 4:
  - Verschiedene Ansichten der Implementierung in den einzelnen Gruppen
  - Wenige Fortschritte in der Implementierung
  - Fehlende Zeit, um genügend Gruppentreffen zu organisieren

### Treffen am 01.06.2017

- Besprechen des Protokolls der Präsentation bei CeGaT in Tübingen
  - Was ändert sich für unsere Implementierung?
  - Haben wir die Anforderungen richtig verstanden
- Tracking soll personalisiert sein
- Vorstellung des bisherigen Fortschritts vor den wissenschaftlichen Mitarbeitern

### Treffen am 02.06.2017

- Besprechung der Änderung der Schnittstellendefinition mit Gruppe Entnahmeliste
  - Änderung der einzelnen Methoden
  - Statuscodes für Primer besprochen

### Treffen am 12.06.2017 Sprint 5 – Review

Besprechung des aktuellen Stands im Team:

- In diesem Sprint konnten alle Tasks erfolgreich umgesetzt werden
  - Erledigt:
    - Login GUI
    - Login Berechtigungsabfrage
    - GUI Prozessauswahl
    - Manuelle Primersuche
    - Dokumentation aller Treffen in Sprint5
- Punkte die in Sprint 6 zusätzlich noch erledigt werden müssen:
  - Offene Tasks aus Sprint 5
  - Login Ansicht überarbeiten
  - Strings überarbeiten
  - Code kommentieren
  - Extra GUI für zuletzt bearbeitete Prozessliste
  - JUnit Tests schreiben
  - Scanfunktion in der Tabelle anpassen
  - Activities überarbeiten

## Sprint Reviews

### Sprint 1 – Review am 13.04.2017

- Es wurden im vergangenen Sprint Vorbereitungen getroffen
  - Projekt angelegt und auf jedem PC zum Laufen gebracht
  - Benötigte Hardware festgelegt und bestellt
  - Recherche zu verschiedener Software und Schnittstellen
  - Definition of Done
  - Tomcat angeschaut
  - Noch keine Funktionalitäten implementiert
- Was fehlt noch?
  - Aktivitätsdiagramm überarbeiten und fertigstellen
  - Klassendiagramm erstellen
  - Mock-Up überarbeiten
  - Mit REST arbeiten/ beschäftigen
  - Barcode-Scanner API anschauen



## Sprint 2 - Review 2 am 27.04.2017

Vorstellung des aktuellen Stands vor dem Product Owner:

- Alle Tasks die in Sprint 2 enthalten waren, konnten erfolgreich umgesetzt und Sprint2 ohne ausstehende Tasks abgeschlossen werden:
  - Überarbeitung der Mockups
  - Bearbeitung des Aktivitätsdiagramms
  - Erstellen eines Klassendiagramms
  - Recherche zu Android Barcode API
  - Recherche Google Guidelines User Interfaces
  - Dokumentation aller Treffen in Sprint 2
- Mockups wurden besprochen und erklärt
  - Wie kamen die verschiedenen GUIs zustande und welchen Zweck erfüllen sie
  - Wie sollen die einzelnen Funktionen umgesetzt werden
  - Abklärung von must-haves und nice-to-haves
- Vorstellung von Aktivitätsdiagramm und Klassendiagramm
  - Verständnis der verschiedenen Schnittstellen
  - Abhandlung der verschiedensten Fälle, die während eines Pickvorgangs auftreten können
- Sammeln von positiven Aspekten während der Gruppenarbeit in Sprint2:
  - Gute Kommunikation zwischen den Teammitgliedern
  - Erfolgreiche Organisation und Einhaltung der vereinbarten Gruppentreffen
  - Aufteilung der verschiedenen Aufgaben untereinander
  - Jedes Teammitglied hat sich bestmöglich in der Gruppe eingebracht und mitgearbeitet
  - Gute Ergänzung und Hilfestellung untereinander in verschiedenen Themenbereichen
- Sammeln von Verbesserungsvorschlägen für Sprint3:
  - Besseres Abklären von Definitionen und Verständnis von Begriffen, um auf einem Wissensstand und ohne Missverständnisse arbeiten zu können
  - Aufgaben noch spezifizierter unterteilen
  - Intensivere Absprache und Treffen mit anderen Gruppen, um immer auf dem gleichen Stand arbeiten zu können, was Änderungen der Implementierungen und Ideen in den jeweils anderen Gruppen angeht

## Sprint 3 - Review am 12.05.2017

Vorstellung des aktuellen Stands vor dem Product Owner:

- Von den vorgenommenen Tasks konnten in diesem Sprint nicht alle umgesetzt werden. Grund dafür waren häufige Änderungen der Gesamtarchitektur nach Treffen im Plenum mit dem Kurs und den Dozenten
  - Erledigt:
    - Schnittstellendefinition
    - GUI für die Primeranzeige
    - Implementierung der Scanfunktion
    - Überarbeitung der Diagramme
    - Mockups
    - Dokumentation aller Treffen in Sprint3
  - Nicht geschafft:
    - Login Berechtigungsabfrage
    - GUI für die Prozessauswahl
    - Manuelle Primersuche
    - Login GUI
    - Menü erstellen
- Im Plenum wurden Begrifflichkeiten und Interpretationen von Namensgebungen im DataDictionary festgehalten
- Der Fokus wurde in diesem Sprint auf die Fertigstellung von Mockups gelegt, da diese beim Beta-Release dem Kunden vorgelegt werden müssen
  - Mockups bzw. Mockup-Präsentation wurde vorgestellt und erklärt
  - Programmierung ist somit in diesem Sprint liegen geblieben
- Vorstellung des Klassendiagramms und der Schnittstellendefinition
- Punkte die in Sprint 4 zusätzlich noch erledigt werden müssen:
  - Schnittstellendefinition muss noch angepasst werden
  - Implementierung der GUI basierend auf den Mockups
- Positiven Aspekte während Sprint3:
  - Das Gruppenklima ist noch gut
  - Gruppeninterne treffen konnten wahrgenommen werden
  - Organisation und Kommunikation in der Gruppe funktionieren gut
- Negative Aspekte während Sprint3:
  - Die Planung und die Architektur ändern sich täglich
  - Uneinigkeiten zwischen den Gruppen, sowie Änderungen bei einzelnen Gruppen von denen man erste Tage später zufällig etwas erfährt
  - Es kann nicht richtig mit der Programmierung und Implementierung begonnen werden
  - Während der Zeiten, die für Programmierung genutzt werden könnte, werden wieder Änderungen besprochen

## Sprint 4 - Review am 29.05.2017

Besprechung des aktuellen Stands im Team:

- Von den vorgenommenen Tasks konnten in diesem Sprint nicht alle umgesetzt werden.  
Fokus auf Usability-Test-Präsentation bei CeGaT vor Ort:
  - Erledigt:
    - Erstellung der Mockups (zur Visualisierung bei CeGaT)
    - Schnittstelle überarbeitet
    - Klassendiagramm überarbeitet
    - Dokumentation aller Treffen in Sprint4
  - Nicht geschafft:
    - Login GUI
    - Login Berechtigungsabfrage
    - GUI Prozessauswahl
    - Manuelle Primersuche
    - Menü erstellen
- Es wurde nicht allzu viel an den Tasks weitergearbeitet, da auch auf die Rückmeldung des Kunden bezüglich der Mockups gewartet wurde
  - Folgend muss die Implementierung vorangetrieben werden
- Punkte die in Sprint 5 zusätzlich noch erledigt werden müssen:
  - Offene Tasks aus Sprint 4
  - Dummy Server zum Testen anlegen
- Positiven Aspekte während Sprint 4:
  - Das Gruppenklima ist noch gut
  - Gruppeninterne Treffen konnten wahrgenommen werden
  - Organisation und Kommunikation in der Gruppe funktionieren gut
- Negative Aspekte während Sprint 4:
  - Verschiedene Ansichten der Implementierung in den einzelnen Gruppen
  - Wenige Fortschritte in der Implementierung
  - Fehlende Zeit, um genügend Gruppentreffen zu organisieren

## Sprint 5 - Review am 12.06.2017

Besprechung des aktuellen Stands im Team:

- In diesem Sprint konnten alle Tasks erfolgreich umgesetzt werden
  - Erledigt:
    - Login GUI
    - Login Berechtigungsabfrage
    - GUI Prozessauswahl
    - Manuelle Primersuche
    - Dokumentation aller Treffen in Sprint5
- Punkte die in Sprint 6 zusätzlich noch erledigt werden müssen:
  - Offene Tasks aus Sprint 5
  - Login Ansicht überarbeiten
  - Strings überarbeiten
  - Code kommentieren
  - Extra GUI für zuletzt bearbeitete Prozessliste
  - JUnit Tests schreiben
  - Scanfunktion in der Tabelle anpassen
  - Activities überarbeiten

## Sprint 6 - Review am 26.06.2017

Besprechung des aktuellen Stands im Team:

- Von den vorgenommenen Tasks konnten in diesem Sprint nicht alle umgesetzt werden.  
Fokus auf Usability-Test-Präsentation bei CeGaT vor Ort:
  - Erledigt:
    - Primer aufrufen (Popup Fenster)
    - Bemerkungen und Abteilung hinzufügen
    - Liste aktualisieren (Primer ausgrauen o.ä)
    - Fehlermeldung bei falschem Primer
    - Laboransicht für zuletzt prozessierte Liste
    - Primer-Tracking: Bemerkung & Destination
    - LoginGUI überarbeiten
    - Activities überarbeiten
    - Junittests schreiben
    - Logout implementieren
    - Scanfunktion an Tabelle anpassen
    - PopUp-Fenster richtig anzeigen in Entnahme
    - Rückgabe GUI erstellen
    - Manuelle Entnahme in EntnahmeGUI implementieren
    - Liste anzeigen aus RestRequest
    - RestRequest schreiben
  - Nicht geschafft:
    - Strings überarbeiten
    - Kommentare schreiben
- In diesem Sprint wurde der Großteil der Software implementiert. Trotz großem Zeitdruck konnten fast alle Tasks fertiggestellt werden
- Es wurde noch ein kurzer zusätzlicher Sprint 7 begonnen, um offene Tasks und noch zu erledigende Aufgaben dokumentieren zu können
- Punkte die in Sprint 7 zusätzlich noch erledigt werden müssen:
  - DataDictionary anlegen
  - Technische Dokumentation anlegen
  - Benutzerdokumentation fertigstellen
- Positiven Aspekte während Sprint 6:
  - Großer Fortschritt in der Implementierung
  - Kommunikation mit den anderen Gruppen war inzwischen eingespielt
  - Jedem war bewusst, auf was für Aspekte er seinen Fokus legen muss
- Negative Aspekte während Sprint 6:
  - Zeitdruck
  - Ungleiche Arbeitsverteilung im Team
  - Fehlende Zeit, um genügend Gruppentreffen zu organisieren