

Pflichtenheft



Lesepreis-Application-System LesePreisUI

Auftraggeber:

Dorothea Prenn

Dorothea.Prenn@schule.suedtirol.it

Auftragnehmer:

Manuel Ploner, Mattia Galiani, Maximilian
Miterrutzner

E-Mail: stploman@bx.fallmerayer.it

Ansprechpartner: Manuel Ploner

Datum: 12.04.2020

Dokumenthistorie

Version	Datum	Bearbeiter	Kommentar	Status
0.1	27.01.2020	Alle	Erste Version	WIP
0.2	16.02.2020	Alle	<Muss Anforderungen	WIP
0.3	11.04.2020	Manuel Ploner	<Use Cases	STABLE
1.0	12.04.2020	Manuel Ploner	Release	STABLE

Inhaltsverzeichnis

1	Anforderungen und Ziele	5
1.1	Überblick	5
1.2	Detail Muss-Anforderungen	5
1.2.1	Teilnehmer hinzufügen	5
1.2.2	Teilnehmer verwalten	5
1.2.3	Buch zu Teilnehmer hinzufügen	5
1.2.4	Verwendung für den Gruppenlesepreis	6
1.2.5	Visualisierung aller Teilnehmer mit allen gelesenen Büchern	6
1.2.6	Generierung einer Liste mit allen Teilnehmern nach Anzahl an Losen sortiert.	6
1.2.7	Verlosung von Preisen	6
1.3	Detail Kann-Anforderungen	6
1.3.1	Speicherung der Daten auf einer Cloud	6
1.3.2	Erweiterte Einstellungen	6
2	Produkteinsatz	7
2.1	Geschäftsprozessanalyse	7
2.2	Problembereich	8
2.3	Zielgruppe und Voraussetzungen	8
3	Use Cases	9
3.1	Akteure	9
3.1.1	Bibliothekarin	9
3.1.2	System	9
3.2	Use Case Diagramm	9
3.3	Einzelne Use Cases	10
3.3.1	Use Case: „Leser hinzufügen“	10
3.3.2	Use Case: „Verlosung starten“	11
3.3.3	Use Case: „Gewinner aussuchen“	12
3.3.4	Use Case: „Buch bei einem Leser als gelesen markieren“	13
3.3.5	Use Case: „Prüfen ob bereits gelesen“	14
3.3.6	Use Case: „Data Backup erstellen“	15
3.3.7	Use Case: „Buch/Buchliste importieren/exportieren“	16
3.3.8	Use Case: „Aktuellen Punktestand als Liste exportieren“	17
4	Reengineering (Ist-Zustand)	18
4.1	Sequenzdiagramm	18
4.2	Aktivitätsdiagramm	19

5	Grobentwurf (Soll-Zustand)	20
5.1	Klassendiagramm.....	20
6	Qualitätsmerkmale	21
6.1	Benutzbarkeitsanforderungen.....	21
6.2	Effizienzanforderungen.....	21
6.3	Wart- und Portierbarkeitsanforderungen	21
6.4	Sicherheitsanforderungen	21
6.5	Gesetzliche Anforderungen	21
7	Umsetzungsdetails	22
7.1	Zielplattform	22
7.2	Datenhaltung	22
7.3	Oberfläche.....	22
7.4	Systemarchitektur.....	22
8	Projektplan	23
8.1	Zeitplan	23
8.2	Kostenschätzung.....	23

Abbildungsverzeichnis

Abbildung 1: Geschäftsprozessanalyse	7
Abbildung 2: Use Case Diagramm	9
Abbildung 3: Sequenzdiagramm.....	18
Abbildung 4: Aktivitätsdiagramm Programm	19

1 Anforderungen und Ziele

1.1 Überblick

Es soll ein System zum Ersetzen der derzeitigen Methode zur Erfassung von Lesepreis-Losen sowie deren Verlosung entwickelt werden, da der Prozess derzeit allgemein zu arbeitsaufwändig und fehlerbehaftet ist. Die Datenspeicherung soll zentral über eine Datenbank erfolgen und lokal bearbeitbar sein.

Folgende Funktionen sollen vom System abgedeckt werden:

- Teilnehmer hinzufügen
- Teilnehmer verwalten
- Buch zu Teilnehmer hinzufügen
- Verwendung für den Gruppen-Lesepreis
- Visualisierung aller Teilnehmer mit allen gelesenen Büchern (für Bibliothekar)
- Generierung einer Liste mit allen Teilnehmern nach Anzahl an Losen sortiert (für Teilnehmer)
- Verlosung von Preisen (inkl. Animation)

1.2 Detail Muss-Anforderungen

1.2.1 Teilnehmer hinzufügen

Das System soll eine Möglichkeit bieten, Teilnehmer zum Wettbewerb hinzuzufügen. Dabei sollen Vorname und Nachname gespeichert werden und der Teilnehmer in eine Datenbank abgelegt werden.

1.2.2 Teilnehmer verwalten

Es soll möglich sein, die Teilnehmer zu verwalten, zu entfernen und Sperren für eventuelle Betrugsversuche zu erstellen.

1.2.3 Buch zu Teilnehmer hinzufügen

Es soll eine Möglichkeit geben, bei bestandem Test zu einem Buch dies im System zu vermerken. Dabei soll der Name des Buches, Name des/r Autors/in, die Sprache des Buches, die Anzahl an dafür erhaltenen Losen, das Datum sowie Zeitpunkt des Testes, eine Auflistung der gegebenen Antworten zu den Fragen und der Name des/r den Test durchführenden Bibliothekars/in vermerkt werden. Auch nicht bestandene Tests sollen erfasst werden, um Betrugsversuche leichter aufdecken zu können. Dabei soll es nur nötig sein, das Buch aus einer Liste mit allen für den Lesepreis bereitstehenden Büchern mithilfe einer Suchfunktion welche wahlweise den Namen des Buches oder des/r Autors/in entgegennimmt auszuwählen und die Leitung des Lesers einzutragen. Die Anzahl an dafür erhaltenen Losen und die Sprache des Buches sollen vollautomatisch aus einer für diesen Zweck bereitstehenden Liste und das Datum und der Zeitpunkt mithilfe des Betriebssystems ausgelesen werden. Der Name des/r den Test durchführenden Bibliothekars/in soll durch den derzeitigen Benutzer des Systems erfasst werden.

Sollte der Leser den Test bestanden haben (mindestens vier richtige Antworten von maximal sechs Fragen) werden ihm dann die Lose gutgeschrieben. Außerdem soll das System mögliche Fehler seitens des Benutzers erkennen indem es unter anderem überprüft, ob der Teilnehmer das Buch bereits gelesen, den Test nicht bestanden oder bereits mehr als die maximalen 50 Bücher gelesen hat und das aktuelle Datum mit der Frist vergleicht um unerlaubte verspätete Einträge zu erkennen und zu melden.

1.2.4 Verwendung für den Gruppenlesepreis

Das System soll einen Modus für den Gruppenlesepreis besitzen bei dem gleich wie beim Einzel-Lesepreis Leser sowie gelesene Bücher hinzugefügt und verwaltet werden können. Zusätzlich soll es die Möglichkeit geben, Gruppen zu erstellen und zu verwalten.

1.2.5 Visualisierung aller Teilnehmer mit allen gelesenen Büchern

Das System soll eine Möglichkeit bieten, alle Teilnehmer mit den gelesenen Büchern als kurze Übersicht für den/die Bibliothekar/in anzuzeigen.

1.2.6 Generierung einer Liste mit allen Teilnehmern nach Anzahl an Losen sortiert.

Das System soll eine Möglichkeit bieten, eine Liste mit allen Teilnehmern inklusive der Anzahl der jeweiligen Lose sortiert nach ebenjenen auszugeben, welche in ausgedruckter Form in der Bibliothek aufgehängt werden kann und als Hauptinformationsquelle für die Teilnehmer und andere Personen für den aktuellen Status des Lesepreises dient.

1.2.7 Verlosung von Preisen

Das System soll die Aufgabe der Verlosung von Preisen bei der Abschlussfeier übernehmen. Dabei soll das Programm dafür sorgen, dass kein Teilnehmer mehr als drei Preise erhält. Auch die grafische Präsentation der Verlosung soll durch eine Animation in einer Art ansprechend gestaltet werden, dass man sie auf eine Leinwand gebeamt vor den versammelten Teilnehmern präsentieren kann und nicht nur der Name des Gewinners erscheint.

1.3 Detail Kann-Anforderungen

1.3.1 Speicherung der Daten auf einer Cloud

Daher, dass das Programm auf mehrere Rechner laufen sollte und deswegen eine Synchronisation nötig sein würde, müsste geschaut werden, dass das Programm über eine mögliche Synchronisationsfunktion fügt, um die Arbeit auf mehreren Rechnern zu erlauben.

1.3.2 Erweiterte Einstellungen

Generelle Einstellungen, wie der Namenvergabe beim Gruppenlesepreis, Festlegung des Datums für die Gültigkeit des Lesepreises, Verwaltung der Sprachen

2 Produkteinsatz

Das LesepreisUI wird zur Verwaltung des Lesepreises am Oberschulzentrum Fallmerayer von den zuständigen Bibliothekarinnen eingesetzt.

2.1 Geschäftsprozessanalyse

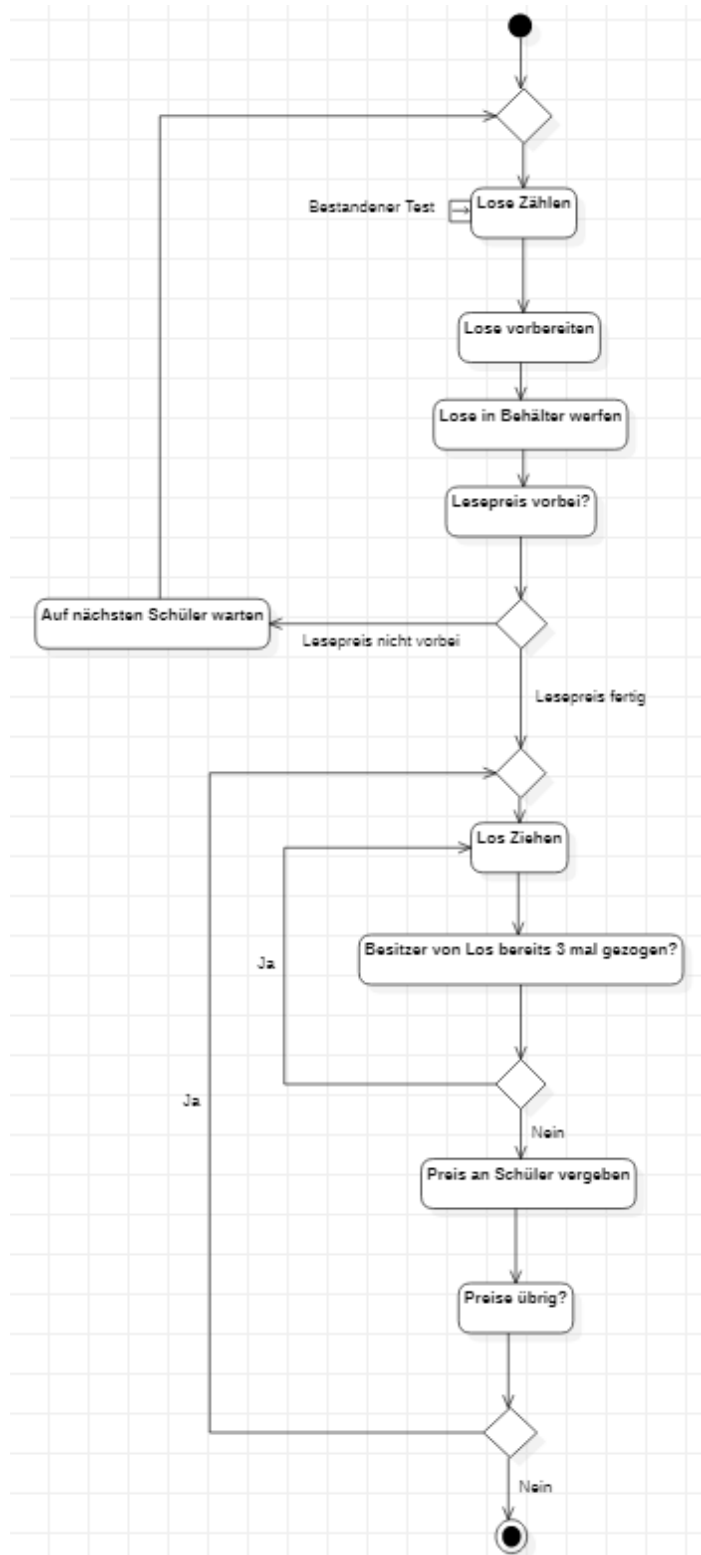


Abbildung 1: Geschäftsprozessanalyse

Der aktuelle Prozess des Lesepreises erfolgt wie folgt: Die Bedingung, dass man am Lesepreis teilnehmen kann, ist, dass man die Prüfung eines gelesenen Buches besteht. Hierdurch erhält man eine vordefinierte Anzahl an Losen, welche dann später bei der Verlosung des Lesepreises verwendet werden können. Diese Lose werden in den Behälter geworfen, in dem alle Lose aller Schüler vorzufinden sind. Dieser Prozess wird solange wiederholt, bis der Lesepreis vorbei ist. Am Ende findet die Verlosung statt, bei der jeder zugelassene Schüler teilnehmen kann. Dieser kann nur maximal 3 Preise gewinnen. Dieser Vorgang wird solange wiederholt, bis keine Preise mehr zur Verfügung stehen.

2.2 Problembereich

Das LesepreisUI wird gebraucht um den händisch aufwendigen und monotonen sowie ineffiziente Prozess der Verwaltung des Lesepreises zu vereinfachen. Hierbei kann der Nutzer alle Schüler sowie Bücher digital und effizient verwalten. Die Fehlerbehebung erfolgt durch die gute Struktur der App und aufgrund des digitalen Verfahrens effizienter. Die Schummelgefahr wird durch eingebaute Schutzmechanismen beseitigt.

2.3 Zielgruppe und Voraussetzungen

Der Benutzer dieser App wird eine zuständige Bibliothekarin der Schulbibliothek am Oberschulzentrum Fallmerayer sein, welche grundlegende informatische Kenntnisse haben muss. Zur Führung dieser App werden keine weiteren Fähigkeiten vorausgesetzt.

3 Use Cases

Kapitel 3.1 beschreibt die Akteure, Kapitel 3.2 zeigt das vollständige Use-Case-Diagramm, das Kapitel 3.3 beschreibt die einzelnen Use-Cases auf Basis der Anforderung aus Kapitel 1.

3.1 Akteure

3.1.1 Bibliothekarin

Dieser Benutzer ist menschlich, primär und aktiv. Dieser Nutzer ist der Administrator des gesamten Programmes, da nur dieser Benutzer Aktionen im Programm ausführen kann. Dieser Benutzer kann Bücher, Teilnehmer und Prüfungen verwalten. Er kann auch Listen von Büchern und Schülern importieren sowie exportieren. Generelle Programmeinstellungen können nur von diesem Nutzer durchgeführt werden.

3.1.2 System

Dieser Akteur ist nicht menschlich, primär und passiv. Dieser Nutzer ist zur Durchführung des Programmes notwendig und ist an jedem Prozess, der von der Bibliothekarin ausgeführt wird, beteiligt. Auch kann dieser Akteur auf Reaktion einer Aktion der Bibliothekarin Prozesse ausführen, die nicht das Einverständnis der Bibliothekarin erfordern.

3.2 Use Case Diagramm

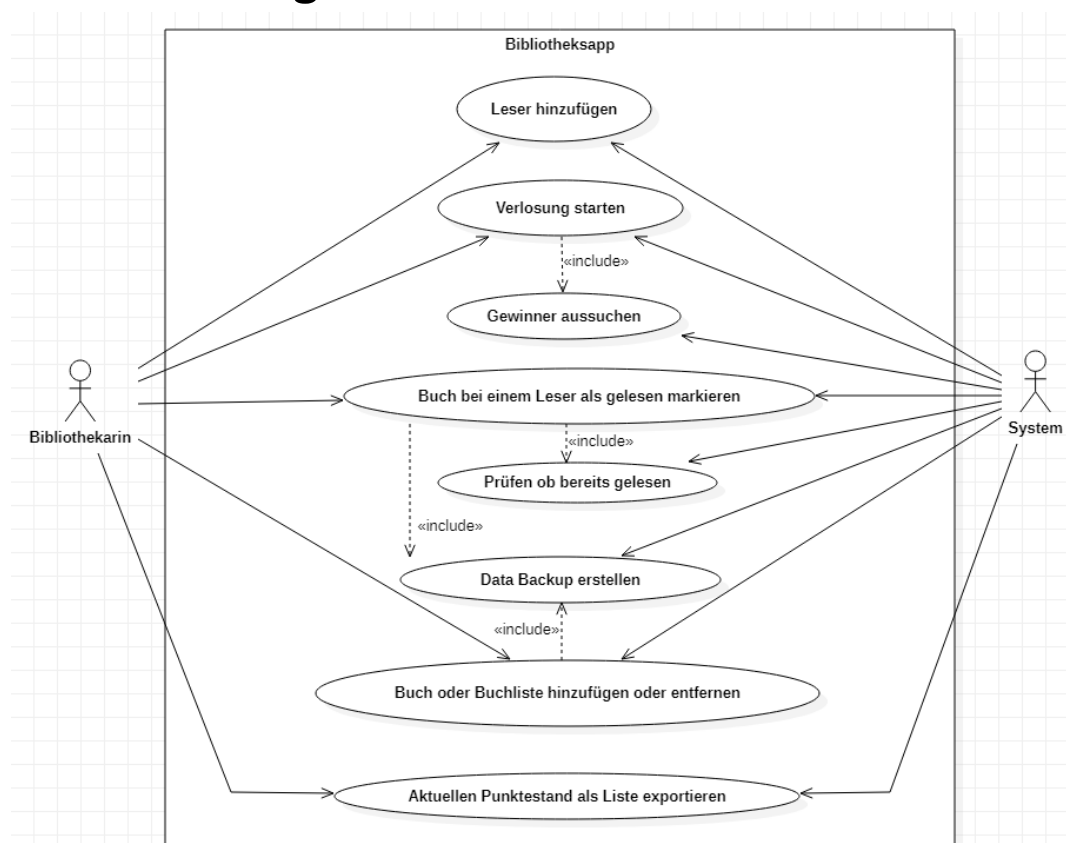


Abbildung 2: Use Case Diagramm

3.3 Einzelne Use Cases

3.3.1 Use Case: „Leser hinzufügen“

Kurze Beschreibung:

Diese Funktionen erstellt einen neuen Teilnehmer und fügt ihn nach der Bearbeitung in die Liste der Teilnehmer hinzu.

Auslöser:

Bibliothekarin klickt auf den „Teilnehmer hinzufügen“-Button

Akteur(e):

- Bibliothekarin
- System

Schritte:

1. Der Nutzer kann die nötigen Einstellungen treffen
 - a. Name
 - b. Nachname
 - c. Klasse
 - d. Gruppe
 - e. Prüfungen

Alternative Schritte:

- /

Nachbedingung(en):

- Das System speichert den neuen Nutzer

Inkludierte Use Cases:

- /

3.3.2 Use Case: „Verlosung starten“

Kurze Beschreibung:

Diese Funktionen evaluiert alle Punkte der Gruppen, sowie die der Teilnehmer und packt alle zugelassenen Teilnehmer/Gruppen in eine Liste. Nachdem startet eine Animation mit der Verlosung der Gewinner.

Auslöser:

Bibliothekarin klickt auf den „Verlosung starten“-Reiter

Akteur(e):

- Bibliothekarin
- System

Schritte:

1. Der Nutzer kann die Anzahl der Preise festlegen
2. Bestätigen
3. Mit Enter weiterschalten

Alternative Schritte:

- /

Nachbedingung(en):

- Durch das Drücken der ENTER-Taste kann immer wieder ein neuer Gewinner verlost werden

Inkludierte Use Cases:

- „Gewinner aussuchen“

3.3.3 Use Case: „Gewinner aussuchen“

Kurze Beschreibung:

Diese Funktion sucht durch das Zufallsprinzip einen Gewinner aus, welcher noch nicht drei Mal vorgekommen ist.

Auslöser:

Verlosung läuft

Akteur(e):

- System

Schritte:

1. Ist der Teilnehmer zugelassen worden?
2. Wurde dieser bereits drei Mal prämiert?

Alternative Schritte:

- /

Nachbedingung(en):

- Gewinner wird an die Verlosung weitergegeben

Inkludierte Use Cases:

- /

3.3.4 Use Case: „Buch bei einem Leser als gelesen markieren“

Kurze Beschreibung:

Diese Funktion fügt dem Leser eine Prüfung hinzu, welche der Nutzer nicht unbedingt bestanden haben muss.

Auslöser:

Beim Teilnehmer „Prüfung hinzufügen“-Button klicken

Akteur(e):

- System
- Bibliothekarin

Schritte:

1. Buchliste wird aufgerufen
2. Bibliothekarin sucht Buch aus
3. Es müssen die Antworten des Lesers eingetragen werden

Alternative Schritte:

- /

Nachbedingung(en):

- Teilnehmer wird gespeichert

Inkludierte Use Cases:

- Prüfen ob bereits gelesen

3.3.5 Use Case: „Prüfen ob bereits gelesen“

Kurze Beschreibung:

Diese Funktion schaut, ob das Buch bereits gelesen ist. Und leitet das System zurück auf den Teilnehmer bzw. gibt eine Alert aus.

Auslöser:

Beim Teilnehmer „Prüfung hinzufügen“-Button klicken

Akteur(e):

- System

Schritte:

1. Schauen ob das Buch gelesen ist

Alternative Schritte:

- /

Nachbedingung(en):

- Weiterleitung zu Prüfung hinzufügen bzw. Alert ausgeben

Inkludierte Use Cases:

- /

3.3.6 Use Case: „Data Backup erstellen“

Kurze Beschreibung:

Speichert die Daten in das Aktuelle File und erstellt eine ZIP Datei mit allen Vorgängerversionen.

Auslöser:

Änderung von Teilnehmer-, Prüfungs- oder Buchdaten

Akteur(e):

- System

Schritte:

1. Daten sammeln
2. In XML File exportieren
3. ZIP erstellen

Alternative Schritte:

- /

Nachbedingung(en):

- /

Inkludierte Use Cases:

- /

3.3.7 Use Case: „Buch/Buchliste importieren/exportieren“

Kurze Beschreibung:

Daher, dass nach fünf Jahren die Buchlisten erneut verwendet werden, gibt es eine Importfunktion alter Buchlisten, sowie individueller Bücher.

Auslöser:

Bibliothekarin klickt auf den „Buchliste importieren/exportieren“-Button

Akteur(e):

- Bibliothekarin
- System

Schritte:

1. Datei auswählen
2. Exportieren/Importieren
3. Daten anzeigen lassen

Alternative Schritte:

- /

Nachbedingung(en):

- /

Inkludierte Use Cases:

- „Data Backup erstellen“

3.3.8 Use Case: „Aktuellen Punktestand als Liste exportieren“

Kurze Beschreibung:

Daher, dass es immer eine aktuelle Rangliste der Teilnehmer gibt, ermöglicht diese Funktion das Exportieren einer Rangliste.

Auslöser:

Bibliothekarin klickt auf den „Rangliste exportieren“-Button

Akteur(e):

- Bibliothekarin
- System

Schritte:

1. Datei auswählen
2. Exportieren

Alternative Schritte:

- /

Nachbedingung(en):

- /

Inkludierte Use Cases:

- /

4 Reengineering (Ist-Zustand)

4.1 Sequenzdiagramm

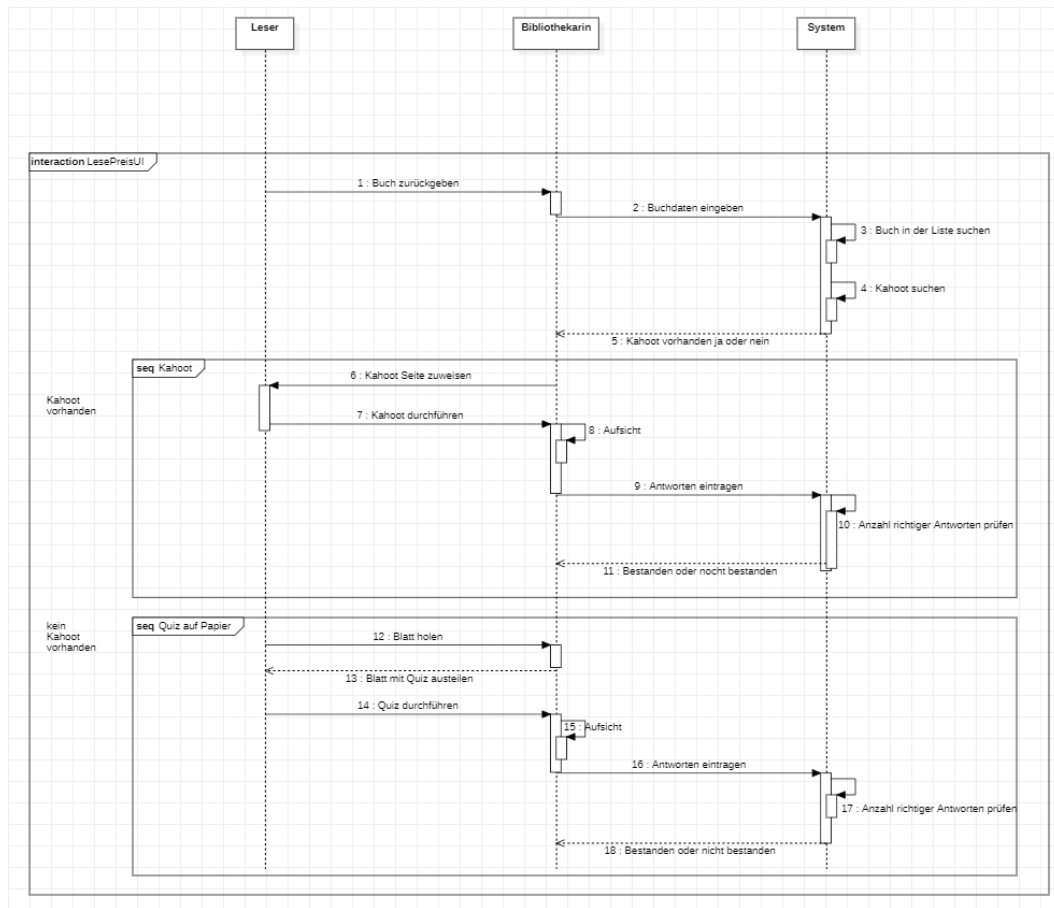


Abbildung 3: Sequenzdiagramm

4.2 Aktivitätsdiagramm

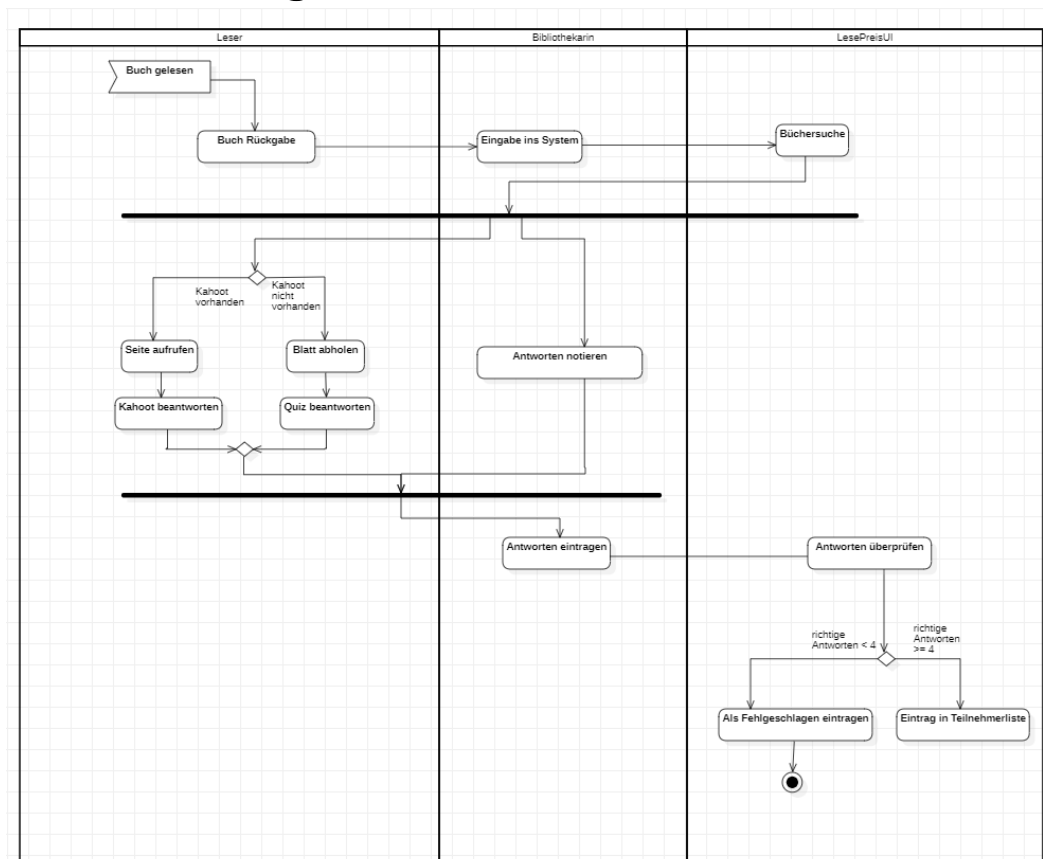


Abbildung 4: Aktivitätsdiagramm Programm

5 Grobentwurf (Soll-Zustand)

5.1 Klassendiagramm

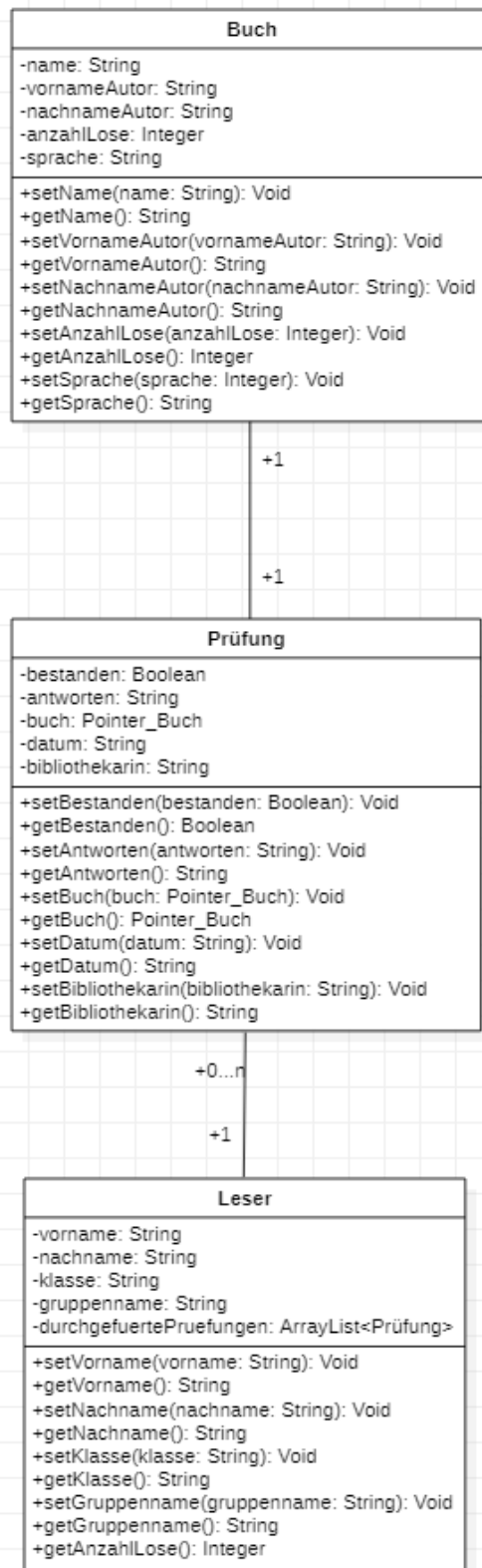


Abbildung 5: Klassendiagramm

6 Qualitätsmerkmale

6.1 Benutzbarkeitsanforderungen

Das Programm soll Benutzerfreundlich, sowie auf mehreren Rechnern gleichzeitig ausführbar sein. Desweiteren soll es möglich sein, dass Bücherlisten importierbar sein sollen sowie die Verwaltung der Teilnehmer und der Bücher in detaillierter Form als auch in einer allgemeinen Liste anzeigbar sein sollte. Das Programm soll ebenfalls einen Einstellungstab haben, um die Sprachen zu verwalten sowie eine Einstellmöglichkeit haben, das Gültigkeitsdatum des Lesepreises festlegen zu können. Auch soll das Programm über einen Verlosungstab verfügen, um eine animierte Verlosung der Preise zu starten. Bei den Bücher- bzw. Teilnehmerlisten soll es zudem eine Suchfunktion geben.

Das Design ist sekundär und spielt im Gegensatz zur Funktionalität keine große Rolle. Spezifiziert werden muss, dass es sich bei der Animation um eine ähnliche Vorgangsweise wie bis hierhin handelt, d.h. eine Hand, die in einem Topf greift und Lose zieht.

6.2 Effizienzanforderungen

Im Bereich der Effizienz gibt es keine Einschränkungen. Das Prozessieren der Daten sowie Verlosungen soll in einer angepassten Geschwindigkeit passieren, sodass keine unnötige Zeit verschwendet wird und das Warten auf den nächsten Preis angenehmer gestaltet werden kann. Die Suchzeit bei den Bücher- sowie Teilnehmerlisten soll keine Auswirkung auf die Effizienz des Programmes haben, jedoch ein Maximum von einer Sekunde nicht überschreiten.

6.3 Wart- und Portierbarkeitsanforderungen

Es soll zu späteren Zeitpunkten möglich sein alte Bücherlisten importieren zu können. Bei generelle Anwendungsfragen soll der Entwickler zur Verfügung stehen. Es soll auch möglich sein das Design bei Bedarf ändern zu können.

6.4 Sicherheitsanforderungen

Im Bereich der Sicherheit ist zu beachten, dass immer mitprotokolliert wird, wann und von welcher Bibliothekarin eine Prüfung geändert wurde. Die Daten der Teilnehmer sowie auch die der Bücher müssen immer aufrufbar sein müssen jedoch nicht extern abgespeichert werden, sondern es reicht eine lokale (am Besten XML) Datei zu haben. Das System soll auch erkennen ob eine Prüfung eines Buches bereits durchgeführt wurde, um Schummeln kein leichtes Spiel ermöglichen zu können.

6.5 Gesetzliche Anforderungen

Gesetzliche Anforderungen wurden Kundenseits aus keine vorgegeben. Das Programm wird Objektorientiert programmiert und beim Code richtet man an die Programmierstandards.

7 Umsetzungsdetails

7.1 Zielplattform

Die Bibliotheksapp-LesepreisUI wird als Java-Applikation umgesetzt. Zielplattform ist Java 8.0 SE und damit alle gebräuchlichen Betriebssysteme. Besondere Hardware-Anforderungen werden nicht gestellt, auf ältere Computermodelle wird jedoch nur bedingt Rücksicht genommen. Eine Bildschirmauflösung von maximal 1024*1280 Pixel wird – bedingt durch die Auflösung der zur Verfügung stehenden Computer– vorausgesetzt.

7.2 Datenhaltung

Die Daten werden in einer lokalen XML-Datei gespeichert. Zu einem späteren Zeitpunkt kann die Speicherung auch auf einem geteilten Speicher, wie z.B. OneDrive oder SQL erfolgen.

7.3 Oberfläche

Die Oberfläche wird mithilfe von JavaFX erstellt. Hierbei greift man nur auf Open-Source Bibliotheken zurück. Zur Speicherung kommt die w3c Bibliothek ins Spiel, welche ebenfalls keine zusätzlichen Kosten liefert.

Beim Design der Oberflächenkomponenten wird auf größtmögliche Übersichtlichkeit, klare Benutzerführung und intuitive Bedienbarkeit geachtet. Zusätzlich wird ein „Farbleit-System“ eingesetzt, das unterschiedliche Farben für bestimmte Anwendungsfälle vorsehen.

7.4 Systemarchitektur

Eine eigene Server-Komponente (als Teil der Anwendungsschicht) ist nicht vorgesehen. Aus heutiger Sicht wäre der Nutzen einer solchen Lösung verhältnismäßig gering. Dem gegenüber steht ein nicht unerheblicher Mehraufwand bei der Implementierung und in späterer Folge beim Betrieb des Systems. Daher wird die BibliotheksApp als Client-Applikation mit integrierter XML-Anbindung umgesetzt.

8 Projektplan

8.1 Zeitplan

Sprints	Zeitspannen
Sprint 1	13.04.2020 - 19.04.2020
Sprint 2	20.04.2020 – 26.04.2020

8.2 Kostenschätzung

- Zur Verfügung stehende Zeit: 2 Wochen
- Zur Verfügung stehende Entwickler: 3
- Iterationen mit Aufwendigkeit:

Sprints	Beschreibung	Aufwendigkeit [2-10]
Sprint 1:	Buchklasse	6
	Teilnehmerklasse	6
	Prüfungsklasse	8
	Speicherung	8
Sprint 2:	Visualisierung / Design	4
	Verlosung	10
	Gruppenlesepreis	6
	Einstellungen	6

Weitere Details können auf trello.com eingesehen werden.

- Kosten:

Beschreibung	Kosten [€]
Personal	3*1500
Entwicklungsumgebung	+ 0
Bibliotheken	+ 0
Hardware	+ 0
Externe Programme/Systeme	+ 0
Wartung	+ 100
Aufwendigkeit	() * 1
Zeit [Monate] (Planung + Umsetzung)	() * 1
Mehrwertsteuer IT	() * 1,22
Gesamt:	5.612

Die daraus resultierende Kostenschätzung beträgt €5.600 inkl. Mehrwertsteuer IT.