Kinobuchungstool

Autoren: [Leonard Obernhuber](#LeonardObernhuber)

[Alexander Leimbacher](#AlexanderLeimbacher)

[Ryan Blasbalg](#RyanBlasbalg)

# Änderungsnachweis

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Was | Wer | Version |
| 07.05.2015 | - Projektauftrag erhalten  - Projektauftrag durchgelesen  - Dokumentation erstellt  - Informationsbeschaffung (Team Konversation) | **Alexander Leimbacher**  **Leonard Obernhuber**  **Ryan Blasbalg** | 1.0 |
| 21.05.2015 | - |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Legende:   * Braun steht für Arbeiten welche Ryan Blasbalg erledigt hat. * Kaiser-blau steht für Arbeiten welche Leonard Obernhuber erledigt hat. * Grün steht für Arbeiten welche Alexander Leimbacher erledigt hat. * Orange steht für Arbeiten welche Leonard und Alexander erledigt haben.   Schwarz steht für Arbeiten welche die Anwesenden an dem jeweiligen Tag zusammen erledigt haben. | | | |
| [Tabelle #1](#Tnr1) | | | |

# Einführung

Dieses Programm dient den Kinobetreibern zur Verwaltung von Sitzplätzen ihrer Kinosälen. Mit diesem Programm kann auf eine einfache Art ein Platz einer Person zugeordnet werde. Die bereits reservierten Plätze sollen in einer anderen Farbe angezeigt werden.

Inhalt

Ctrl+Klick

[Änderungsnachweis 2](#_Toc419292409)

[Einführung 2](#_Toc419292410)

[Analyse 4](#_Toc419292411)

[Aufgabenstellung 4](#_Toc419292412)

[Ausgangslage 4](#_Toc419292413)

[Soll Zustand 4](#_Toc419292414)

[Detaillierte Aufgabenerstellung 4](#_Toc419292415)

[Meilensteine 6](#_Toc419292416)

[Projektmanagement 7](#_Toc419292417)

[Projektorganisation 7](#_Toc419292418)

[Projektrollen 7](#_Toc419292419)

[Kontakte 8](#_Toc419292420)

[Backup Szenario 8](#_Toc419292421)

[BSCW](#_BSCW) 8

[GitHub 8](#_Toc419292423)

[Lokal 8](#_Toc419292424)

[Versionierung 8](#_Toc419292425)

[Projektmethode 8](#_Toc419292426)

[Vorteile der Projektmethode 9](#_Toc419292427)

[Nachteile der Projektmethode 9](#_Toc419292428)

[Planung 10](#_Toc419292429)

[Arbeitspakete Planung 10](#_Toc419292430)

[Arbeitspakete Beschreibung 11](#_Toc419292431)

[Arbeitspakete Realisierung 12](#_Toc419292432)

[Zeitplan 12](#_Toc419292433)

[System 12](#_Toc419292434)

[Systembeschreibung 12](#_Toc419292435)

[Komponentenübersicht 12](#_Toc419292436)

[UML 13](#_Toc419292437)

[Use Case Diagramm 13](#_Toc419292438)

[Use Case Beschreibung 14](#_Toc419292439)

[Sequenzdiagramm 17](#_Toc419292440)

[Aktivitätsdiagramm 17](#_Toc419292441)

[Klassendiagramm 17](#_Toc419292442)

[Lösungsansatz 17](#_Toc419292443)

[Mockup 17](#_Toc419292444)

[Abschluss 17](#_Toc419292445)

[Blackbox Tests 17](#_Toc419292446)

[Testprotokoll 18](#_Toc419292447)

[Fazit/Abschlussbericht 18](#_Toc419292448)

[Leonard 18](#_Toc419292449)

[Ryan 18](#_Toc419292450)

[Alexander 18](#_Toc419292451)

[Zeitunterschiede 18](#_Zeitunterschiede)

[Leonard 18](#_Leonard)

[Ryan 18](#_Ryan)

[Alexander 18](#_Toc419292455)

[Tabellen-Verzeichnis 18](#_Toc419292456)

[Bilder-Verzeichnis 18](#_Toc419292457)

[Glossar 18](#_Toc419292458)

# Analyse

## Aufgabenstellung

### Ausgangslage

Wir erstellen dieses Projekt neu. Ohne Beispiel oder sonstige Hilfe wird ein Kinobuchungstool von Beginn auf entwickelt.

### Soll Zustand

Der Soll Zustand dieses Programmes sieht

### Detaillierte Aufgabenerstellung

#### Rahmenbedingungen

Die Rahmenbedingungen stehen im Auftrag, diese sind:

* Ein Platzreservationssystem für ein Kino, für Kinoinhaber/Kinoarbeiter
* Die Verwaltung soll für mehrere Kinosäle möglich sein
* In jedem Kinosaal hat es mehrere Plätze, welche in Reihen gegliedert sind
* Jeder Kinosaal und dessen Sitzplatz gehört zu einer gewissen Film Vorstellung
* Es können Reservationen im vorhinaus betätigt werden
* Der Kinobesucher muss eine Telefonnummer hinterlegen

#### Dokumentation

Der Dokumenten-Aufbau soll einer IPA ähneln. Somit ist dieses Dokument auch eine gute Vorbereitung. Da das Dokument in dieser Weise aufgebaut ist, sind auch alle Themen für das Projekt vertreten.

#### Kinobuchungstool

Das Tool muss sauber aufgebaut sein, dies im MVC Pattern.

#### Mittel und Methoden

* Netbeans
* Eclipse
* Word
* GitHub

#### Vorkenntnisse Ryan

|  |  |  |
| --- | --- | --- |
| Bereich | Erfahrungsgrad | Kenntnisgrad |
| Java | ½ Jahr | Mittel |
| NetBeans | ½ Jahr | Mittel |
| Word | 6 Jahr | sehr gute Kenntnisse |
| Tabelle # | | |

#### Vorkenntnisse Leonard

|  |  |  |
| --- | --- | --- |
| Bereich | Erfahrungsgrad | Kenntnisgrad |
| Java | ½ Jahr | Mittel |
| NetBeans | Erstes Mal | Schlecht |
| Word | 7 Jahr | sehr gute Kenntnisse |
| Tabelle # | | |

#### Vorkenntnisse Alexander

|  |  |  |
| --- | --- | --- |
| Bereich | Erfahrungsgrad | Kenntnisgrad |
| Java | ½ Jahr | Mittel |
| NetBeans | Erstes Mal | Schlecht |
| Word | 7 Jahr | sehr gute Kenntnisse |
| Tabelle # | | |

#### Dokumente

Für die Dokumentation wird Microsoft Word verwendet. Die Schriftart ist „Calibri (Textkörper)“. Standartfarbe ist Schwarz. Die Schriftgrösse ist nicht immer gleich, Titel u.a. sind grösser als die Standartschrift. Die Standartschrift (wie hier) ist 11. In der Kopfzeile hat es das Datum und die Autoren. In der Fusszeile wird die Version und die Seitenzahl aufgeführt. Dazu ist dieses Dokument so aufgebaut, dass alles detailliert beschrieben ist. Im Inhaltsverzeichnis ist es möglich mit Ctrl + Klick auf das jeweilige Kapitel zu wechseln. Bei Schriften, welche Blau und unterstrichen sind, kann auch mit Ctrl + Klick zu der jeweiligen Beschreibung gewechselt werden.

#### Produkt

Das Programm soll für ein Kino einsetzbar sein. Es soll ergonomisch designt werden. Weiter soll es leicht verständlich sein. In dem Programm soll es möglich sein, zwischen den verschiedenen Kinosälen zu entscheiden. Wenn ein Platz ausgewählt wird hat diese eine Nummer, auch sollte angezeigt werden, in welcher Reihe sich dieser Platz befindet.

# Meilensteine

|  |  |
| --- | --- |
| Meilenstein # | Meilenstein Beschreibung |
| #1 | Analyse |
| #2 | Realisierung |
| #3 | Planung |
| Tabelle # | |

# Projektmanagement

## Projektorganisation

Hilfestellung

André Frey

Google

Schulkollegen

Kino-Buchungs-Tool

Erstellung

Projektleitung

Gruppenleitung

Projektabnahme

André Frey

Christoph Kölbener

Testing

Gruppe

Ryan Blasbalg

Dokumentation

Gruppe

Ryan Blasbalg

Planung

Gruppe

Ryan Blasbalg

Realisierung

Gruppe

Ryan Blasbalg

Mitarbeiter

Ryan Blasbalg

Leonard Obernhuber

Alexander Leimbacher

Abbildung

## Projektrollen

|  |  |  |
| --- | --- | --- |
| Name | Rolle | Beschreibung |
| Gruppe | Projektleitung | Die Verantwortung liegt bei allen Team-Mitgliedern damit das Projekt bei dem Abgabetermin vollständig ist. |
| Ryan Blasbalg  Leonard Obernhuber  Alexander Leimbacher | Mitarbeiter | Bekommt Aufgaben von anderen Teammitglieder und stellt selbst Aufgaben, um das Projekt voranzutragen. |
| Google | Hilfestellung | In erster Instanz wird Google als Nachschlagewerk für Entwicklungfragen genutzt. |
| André Frey | Hilfestellung | André Frey ist zuständig um Fragen zu beantworten, welche beim Projekt entstehen und mit Google nicht beantwortet werden können. |
| André Frey | Projektabnahme  Fachvorgesetzte | André Frey ist für die Projektabnahme Ende dieses Modules verantwortlich. Bewertet wird das Dokument sowie das Tool. |
| Tabelle # | | |

## Kontakte

|  |  |  |
| --- | --- | --- |
| Gruppenmittglied | Ryan Blasbalg | |
| Adresse | Sonnenstrasse 23  8331 Auslikon | |
| Telefon | Privat | 079 533 33 25 |
| E-Mail | rblasbalg@soreco.ch | |
| Tabelle # | | |

|  |  |  |
| --- | --- | --- |
| Gruppenmittglied | Leonard Obernhuber | |
| Adresse | Ringstrasse 7  8603 Schwerzenbach | |
| Telefon | Privat | 058 666 36 62 |
| E-Mail | [fhitz@soreco.ch](mailto:fhitz@soreco.ch) | |
| Tabelle # | | |

|  |  |  |
| --- | --- | --- |
| Gruppenmittglied | Alexander Leimbacher | |
| Adresse | Ringstrasse 7  8603 Schwerzenbach | |
| Telefon | Privat | 058 666 36 62 |
| E-Mail | fhitz@soreco.ch | |
| Tabelle # | | |

## Backup Szenario

### BSCW

Auf dem BSCW sind die jeweiligen Backups der neusten Version abgelegt. Die Dokumentation sowie das Programm werden ebenfalls versioniert auf dem BSCW aufgeführt.

### GitHub

### Lokal

Jede Person unseres Team hat eine Lokale Sicherung dieses Programmes. So sind immer drei der neusten Versionen gesichert.

## Versionierung

## Projektmethode

Die Aufteilung der Arbeit und die Methode, welche diese Aufgaben gliedert, ist sehr wichtig. Es gibt sehr viele Projektmanagementmethoden. Jede dieser Planungs-Strukturen zeichnen sich mit unterschiedlichen Eigenschaften aus. Wir haben uns für die Wasserfallmethode entschieden.

In diesem Dokument werden die Problemanalysen der Anforderungsdefinition und der Entwurf sowie das Design gezeigt. Danach kann codiert werden. Anschließend wird getestet. Eine Wartung ist nicht vorgesehen.

Abbildung

Entwurf

Realisierung

Analyse

Einführung

Test

Planung

### Vorteile der Projektmethode

* Der Entwicklungsprozess wird in Phasen eingeteilt. Dies hat zum Vorteil, dass die Ergebnisse jeder einzelnen Phase angeschaut werden können, jedoch muss der Meilenstein bis zu dieser Zeit auch schon erreicht sein.
* Nach jeder Phase kann ein Termin abgemacht werden, um die Fortschritte zu besprechen. Falls die Lösungen nicht entsprechend sind, muss die ganze Phase wiederholt werden. Dies ist eigentlich nicht möglich. Darum wird hier auch mit dem Wasserfallmodel mit Iteration gearbeitet.
* Da die Phasen genau und detailliert beschrieben werden, ist diese Methode für ein solches Projekt sehr geeignet.

### Nachteile der Projektmethode

* Ergebnisse sind erst nach dem Ende einer Phase einsehbar. Falls die Ergebnisse nicht zufrieden stellend sind, muss eventuell die komplette Phase wiederholt werden
* Keine Unterstützung von parallelen Aufgaben
* Im Wasserfallmodell werden keine Verantwortlichen definiert

# Planung

## Arbeitspakete Planung

|  |  |  |  |
| --- | --- | --- | --- |
| Nr. | Bezeichnung | Beschreibung | Soll-Aufwand  In Stunden |
| #01 | Änderungsnachweis  (Arbeitsprotokoll) | Das Arbeitsprotokoll wird nach jedem Tag, dieser Dokumentation angefügt (Allgemeine Team Reflektion) | x \* 0.5 |
| #02 | Zeitplan | Zeitplan aufgrund der Arbeitspakete erstellen | 1 |
| #03 | Anwendungsfälle (Use Case Diagramm) | Anwendungsfälle werden aufgrund der Anforderungen definiert und beschrieben und der Dokumentation hinzugefügt | 1.5 |
| #04 | Meilensteine | Die Meilensteine stehen für Aufgaben welche in einer bestimmten Zeit erledigt sein müssen | 0.25 |
| #05 | Einführung (Analyse, Aufgabenstellung, usw.) | Um dem Leser den Einstieg in die Dokumentation zu vereinfachen, wird eine Einführung erstellt und der Dokumentation hinzugefügt | 2.5 |
| #06 | Sequenzdiagramm | Verhaltensdiagramm welches die Kommunikation zwischen Objekten darstellt | 1 |
| #07 | Mockup | Mockup designen, zeichnen und beschreiben | 1.5 |
| #08 | Use Case Beschreibung | Die Use Case beschreiben was möglich ist - dies ist für Entwickler wichtig | 1 |
| #09 | Projektmanagementmethode & Vor-/Nachteile | Die Projektmanagement Methode steht für die Organisation. Hier haben wir das Wasserfallmodel genommen - weitere Erklärungen im Verlaufe dieses Dokumentes | 2 |
| #10 | Testfälle definieren | Die Testergebnisse werden aufgrund der erstellten Testfälle dokumentiert | 1.5 |
| #11 | Aktivitätsdiagramm | Dieses Diagramm beschreibt das Verhalten von System | 1.5 |
| #12 | Backup Szenario | Im Backup Szenario ist aufgeführt , wo die neuste Version gesichert ist | 0.5 |
| #13 | Komponentenübersicht | Die Komponentenübersicht zeigt die verschiedenen Komponenten auf, welche in diesem Programm eingesetzt werden, auch die Arbeitsaufteilung ist dabei ersichtlich | 1.5 |
| #14 | Systembeschreibung / Systemlandschaft | Die Systemlandschaft zeigt wie das Programm Kommuniziert. | 1 |
| #15 | Arbeitspakete | Die Arbeitspakete teilen die Arbeit so auf, dass die einzelnen Komponenten gut getestet, Realisiert und geplant werden können. | 1.5 |
| #16 | Soll Zustand | Der Soll Zustand zeigt auf, wie das Programm am Ende sein soll | 0.5 |
| #17 | Realisierung | In diesem Arbeitsschritt wird das Tool effektiv erstellt / codiert | 8 |
| #18 | Versionierung | In der Versionierung werden die Versionen angezeigt | 0.5 |
| #19 | Dokumentation (Fertigstellung, Korrektur, Abgabe) | Fertigstellung:   * Fazit * Reflexion   Korrektur   * Dokumentation korrigiert   Abschluss:  Dokumentation hochladen | 3 |
| #20 | Pufferzeit | Pufferzeit | 2 |
| Total | | Voraussichtlicher Arbeitsaufwand | 37.75 |
| Tabellen # | | | |

## Arbeitspakete Beschreibung

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Wer |
| 1 | Der Änderungsnachweis zeigt auf, wer welche Arbeiten an welchem Tag erledigt hat. Mit diesem Arbeitsprotokoll gibt es eine bessere Übersicht der Arbeit. | Ryan Blasbalg  Leonard Obernhuber  Alexander Leimbacher |
| 2 | Der Zeitplan ist das Mittel, mit welchem wir das Projekt überwachen können. Der Zeitplan ist mit der Wasserfall Methode kompatibel. | Alexander Leimbacher |
| 3 | Die Anwendungsfälle beschreibt, welche Möglichkeiten bzw. was der User alles machen kann auf dem Interface. | Ryan Blasbalg |
| 4 | Die Meilensteine geben ein Datum vor, an welchem eine bestimmte Arbeit erledigt sein muss. Somit gibt es eine bessere Zeitkontrolle. | Leonard Obernhuber |
| 5 | Die Einführung bietet dem Leser dieses Dokumentes einen ersten Einblick, was u.a. in diesem Projekt erwartet wurde uvm. | Leonard Obernhuber  Ryan Blasbalg |
| 6 |  |  |
| 7 |  |  |
| 8 | Die Use Case Beschreibung erklärt die einzelnen Anwendungsmöglichkeiten welche der Benutzer hat. | Ryan Blasbalg |
| 9 | Die Projektmanagement Methode ist eine wichtige organisatorische Aufgabe welche in jedes Projekt integriert werden muss. | Alexander Leimbacher  Leonard Obernhuber  Ryan Blasbalg |
| 10 | Die Testfälle zeigen auf wie sicher und lauffähig das Programm ist. Alle Testfälle müssen den Anforderungen gerecht werden. Das Programm muss auf die vorgegebenen Qualifikationen getestet werden. | Alexander Leimbacher |
| 11 |  | Leonard Obernhuber |
| 12 | Das Backup Szenario hilft dem Entwickler zu sehen wieviel er geändert hat und dies zu welcher Zeit/bzw. Datum. | Leonard Obernhuber |
| 13 | Die Komponentenübersicht ist eine Aufteilung der Programm Inhalte. Die einzelnen Komponenten werden den Teammittgliedern zugewiesen. | Ryan Blasbalg |
| 14 | Die Systemlandschaft gibt Informationen wie das Programm kommuniziert. | Alexander Leimbacher |
| 15 | Die Arbeitspakete sind Unterteilungen der Arbeiten. So werden den Teammitgliedern Aufgaben gegeben, dass alle wissen wer was bearbeitet. | Ryan Blasbalg |
| 16 | Der Soll Zustand sagt aus wie das Programm schlussendlich sein soll. | Alexander Leimbacher |
| 17 | Das Arbeitspacket der Realisierung wird in der Tabelle weiter unten noch detaillierter dargestellt. Dies ist die Hauptphase nachdem alles geplant und analysiert wurde. | Alexander Leimbacher  Leonard Obernhuber  Ryan Blasbalg |
| 18 | Die Versionierung ist für eine bessere Überwachung. Dies Hilft dem Entwickler. | Leonard Obernhuber |
| 19 | Das Verbessern dieses Dokument wird zugleich ein aufbessern sein. Dieses Dokument soll einem IPA Dokumentation ähneln. | Alexander Leimbacher  Ryan Blasbalg  Leonard Obernhuber |
| 20 | Die Pufferzeit wird gebraucht, falls in dem Projekt etwas falsch läuft und wir länger an einer Aufgabe haben als erwartet. |  |
| Tabelle # | | |

## Arbeitspakete Realisierung

|  |  |  |
| --- | --- | --- |
| # | Kurzbeschreibung | Wer |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Tabelle # | | |

## Zeitplan

# System

## Systembeschreibung

## Komponentenübersicht

MVC Pattern

Main Panel :BorderLayout

Kinosaal :Tabbedpane

Info Panel :Fieldset

Ticket :Tabbedpane

Ryan Blasbalg

Leonard Obernbuber

Alexander Leimbacher

Zusammen

Abbildung

## 

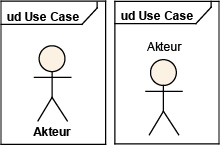
## UML

### Use Case Diagramm

Anzahl plätze definieren

Kinosaal 1 auswählen

Film angeben



Datum angeben

Ticket drucken

Ticket buchen

Plätze anwählen

Kinosaal 2 auswählen

Kinosaal 2 auswählen

Abbildung

Die Stories Kinosaal 1/2/3 auswählen sind alle genau gleich. Desswegen hat nur eine der 3 eine Beschreibung

Kinosaal Tab

Info Fieldset

Main Panel

Abbildung

### Use Case Beschreibung

|  |  |
| --- | --- |
| Use-Case ID | #1 |
| Story Name | Kinosaal 1 auswählen |
| Akteure | User |
| Kurzbeschreibung | Ein Platz kann im Kinosaal eingetragen werden. Nachdem ein Platz ausgewählt wurde, wird ein Popup-Fenster geöffnet, welches den Preis anzeigt. Mit der Bestätigung wird der Platz gebucht- ansonsten steht der Platz wieder zur freien Verfügung. |
| Vorbedingungen | - |
| Nachbedingungen | Die ausgewählten Ticketinformationen werden direkt in einem Tabbed-pane geschrieben und angezeigt |
| Normalablauf | 1. User öffnet das Programm 2. Wählt die Anzahl Plätze 3. User drückt auf den Stuhl im Kinosaal welchen er reservieren möchte 4. Wählt den Film aus 5. Die Sitzplatz Nummer wird automatisch in ein weiteres InputTextFeld geschrieben 6. Die Sitzreihe wird automatisch in ein weiteres InputTextFeld geschrieben 7. Im Ticket weiter unten wird auch der Sitzplatz wie die Sitzreihe als Label eingefügt 8. Der User kann die Tickets anschauen oder die Infos noch erweitern. |
| Ausnahmen | - |
| Tabelle # | |

|  |  |
| --- | --- |
| Use-Case ID | #2 |
| Story Name | Anzahl Plätze definieren |
| Akteure | User |
| Kurzbeschreibung | Die Anzahl Plätze muss hier eingetragen werden so können direkt 2 Tickets gebucht werden.(oder mehrere) |
| Vorbedingungen | - |
| Nachbedingungen | Der User trägt die Plätze im Kinosaal 1/2 oder 3 ein. |
| Normalablauf | 1. User öffnet das Programm 2. Wählt die Anzahl Plätze 3. User drückt auf den Stuhl im Kinosaal welchen er reservieren möchte 4. Wählt den Film aus. 5. Die Sitzplatz Nummer wird automatisch in ein weiteres InputTextFeld geschrieben 6. Die Sitzreihe wird automatisch in ein weiteres InputTextFeld geschrieben 7. Im Ticket weiter unten wird auch der Sitzplatz wie die Sitzreihe als Label eingefügt 8. Der User kann die Tickets anschauen oder die Infos noch erweitern. |
| Ausnahmen | Wenn Stühle bereits gebucht wurden können diese kein zweites Mal gebucht werden. |
| Tabelle # | |

|  |  |
| --- | --- |
| Use-Case ID | #3 |
| Story Name | Film angeben |
| Akteure | User |
| Kurzbeschreibung | Der User wählt aus welcher Film gesehen werden möchte. Zu jedem Kinosaal hat es verschiedene Filme. |
| Vorbedingungen | - |
| Nachbedingungen | - |
| Normalablauf | 1. User öffnet das Programm 2. Wählt die Anzahl Plätze 3. User drückt auf den Stuhl im Kinosaal welchen er reservieren möchte 4. Wählt den Film aus. 5. Die Sitzplatz Nummer wird automatisch in ein weiteres InputTextFeld geschrieben 6. Die Sitzreihe wird automatisch in ein weiteres InputTextFeld geschrieben 7. Im Ticket weiter unten wird auch der Sitzplatz wie die Sitzreihe als Label eingefügt 8. Der User kann die Tickets anschauen oder die Infos noch erweitern. |
| Ausnahmen | Wenn Stühle bereits gebucht wurden können diese kein zweites Mal gebucht werden. |
| Tabelle # | |

|  |  |
| --- | --- |
| Use-Case ID | #4 |
| Story Name | Datum auswählen |
| Akteure | User |
| Kurzbeschreibung | Der User hat die Möglichkeit ein Datum festzulegen, so kann ein Film auch im Voraus gebucht werden. |
| Vorbedingungen | - |
| Nachbedingungen | - |
| Normalablauf | 1. User öffnet das Programm 2. Wählt die Anzahl Plätze 3. User drückt auf den Stuhl im Kinosaal welchen er reservieren möchte 4. Wählt den Film aus. 5. Wählt ein Datum aus um den Film zu sehen (Standard ist das aktuelle Datum) 6. Die Sitzplatz Nummer wird automatisch in ein weiteres InputTextFeld geschrieben 7. Die Sitzreihe wird automatisch in ein weiteres InputTextFeld geschrieben 8. Im Ticket weiter unten wird auch der Sitzplatz wie die Sitzreihe als Label eingefügt 9. Der User kann die Tickets anschauen oder die Infos noch erweitern. |
| Ausnahmen | Jeden Monat kommen neue Filme ins Kino. Somit kann auch nur einen Film gebucht werden im gleichen Monat. |
| Tabelle # | |

|  |  |
| --- | --- |
| Use-Case ID | #5 |
| Story Name | Plätze anwählen |
| Akteure | User |
| Kurzbeschreibung | Der User wählt die Plätze aus welcher der Kunde reserviert haben möchte. Dies kann er mit einem Mausklick auf die Plätze im Kinosaal Bild machen. |
| Vorbedingungen | Die Anzahl plätze müssen bereits eingetragen sein. Standard ist sonst 1. |
| Nachbedingungen | - |
| Normalablauf | 1. User öffnet das Programm 2. Wählt die Anzahl Plätze 3. User drückt auf den Stuhl im Kinosaal welchen er reservieren möchte 4. Wählt den Film aus. 5. Die Sitzplatz Nummer wird automatisch in ein weiteres InputTextFeld geschrieben 6. Die Sitzreihe wird automatisch in ein weiteres InputTextFeld geschrieben 7. Im Ticket weiter unten wird auch der Sitzplatz wie die Sitzreihe als Label eingefügt 8. Der User kann die Tickets anschauen oder die Infos noch erweitern. |
| Ausnahmen | Wenn Stühle bereits gebucht wurden können diese kein zweites Mal gebucht werden. |
| Tabelle # | |

|  |  |
| --- | --- |
| Use-Case ID | #6 |
| Story Name | Tickets buchen |
| Akteure | User |
| Kurzbeschreibung | Drückt den Knopf Tickets buchen, um diese im Programm zu reservieren. |
| Vorbedingungen | Die Anzahl plätze müssen bereits eingetragen sein, diese müssen auch angewählt sein, somit ist der Kinosaal auch angewählt. Weiter muss noch der Film angewählt sein. |
| Nachbedingungen | Zur Kontrolle müssen nochmals alle eingaben angeguckt werden. Wenn der Kunde zahlt, kann die Verkäuferin auf einem Pop-Up Panel noch bestätigen, dass der Kunde das Ticket bezahlt hat und somit ist dies jetzt reserviert. Storniert werden kann nicht. |
| Normalablauf | 1. User öffnet das Programm 2. Wählt die Anzahl Plätze 3. User drückt auf den Stuhl im Kinosaal welchen er reservieren möchte 4. Wählt den Film aus. 5. Die Sitzplatz Nummer wird automatisch in ein weiteres InputTextFeld geschrieben 6. Die Sitzreihe wird automatisch in ein weiteres InputTextFeld geschrieben 7. Im Ticket weiter unten wird auch der Sitzplatz wie die Sitzreihe als Label eingefügt 8. Der User kann die Tickets anschauen oder die Infos noch erweitern. 9. Kann diese Reservierung bestätigen. |
| Ausnahmen | Wenn der Kunde nicht bezahlt gibt es keinen Grund dieses Ticket zu buchen. Der Kunde muss zuerst zahlen bevor das Ticket reserviert wird. |
| Tabelle # | |

|  |  |
| --- | --- |
| Use-Case ID | #7 |
| Story Name | Tickets drucken |
| Akteure | User |
| Kurzbeschreibung | Um die Kaufbestätigung / Ticket zu drucken drückt der User auf diesen Knopf. |
| Vorbedingungen | Das Ticket muss reserviert sein. |
| Nachbedingungen | - |
| Normalablauf | 1. User öffnet das Programm 2. Wählt die Anzahl Plätze 3. User drückt auf den Stuhl im Kinosaal welchen er reservieren möchte 4. Wählt den Film aus. 5. Die Sitzplatz Nummer wird automatisch in ein weiteres InputTextFeld geschrieben 6. Die Sitzreihe wird automatisch in ein weiteres InputTextFeld geschrieben 7. Im Ticket weiter unten wird auch der Sitzplatz wie die Sitzreihe als Label eingefügt 8. Der User kann die Tickets anschauen oder die Infos noch erweitern. 9. Kann diese Reservierung bestätigen. |
| Ausnahmen | - |
| Tabelle # | |

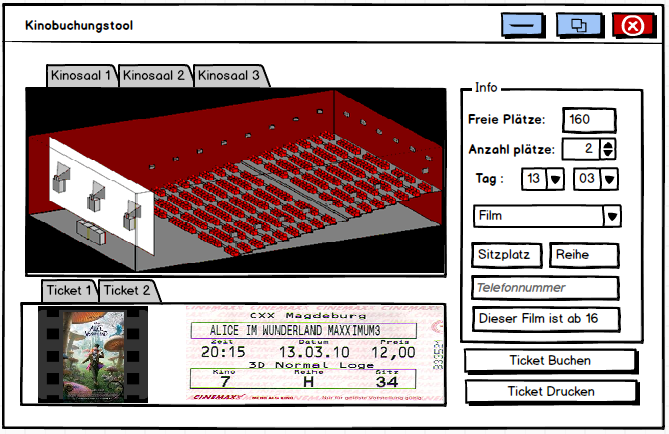
### Sequenzdiagramm

### Aktivitätsdiagramm

### Klassendiagramm

# Lösungsansatz

## Mockup



Abbildung

Das Bild oben zeigt wie das Programm am Ende aussieht. Im [Fieldset](#Fieldset) (Quadrad) Info stehen die freien Plätze. Momentan ist noch kein Platz gebucht. Im Tabbedpane (Kinosaal 1, Kinosaal 2, Kinosaal 3) kann der Kinosaal gewechselt werden. In jedem Kinosaal spielen verschiedene Filme. Die ausgewählte Platzanzahl (im Info Panel) macht gleichzeitig die Anzahl der Tabs im Tabbedpane(Ticket 1, Ticket 2) aus. Wenn der Sitzplatz im Kinosaal (Tabbedpane) ausgewählt wird, verändert der Platz die Farbe. Hinzu kommt, dass der Platz im Info Panel mit der korrekten Platznummer eingesetzt wird. Auch die Reihe wird automatisch in das Textfeld eingesetzt. Wenn der Film ausgewählt ist, validiert ein Prozess, um welche Zeit dieser stattfindet. Auch wird aufgeführt, ab welcher Altersgruppe der Film zugelassen ist. Falls dieses Ticket gebucht ist, werden die Stühle auf der Karte in einer anderen Farbe angezeigt und die Platzanzahl verkleinert sich. Das Ticket wird im underen Panel angezeigt. Mit dem Knopf „Ticket Buchen“ wird das Ticket gebucht. Mit dem Knopf Ticket drucken, wird das Ticket ausgedruckt.

# Abschluss

## Blackbox Tests

|  |  |  |
| --- | --- | --- |
| Test Number | What to Test | Erwartetes Ergebnis |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Tabelle # | | |

## Testprotokoll

## Fazit/Abschlussbericht

### Leonard

### Ryan

### Alexander

## Zeitabweichungen

### Leonard

### Ryan

### Alexander

## Tabellen-Verzeichnis

|  |  |  |
| --- | --- | --- |
| Tabellen # | Tabellen Name | Kurzbeschreibung |
| #1 | Änderungsnachweis | Der Änderungsnachweis zeigt welche Arbeiten von welcher Person erledigt wurden. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Tabelle # | | |

## Bilder-Verzeichnis

|  |  |  |
| --- | --- | --- |
| Bilder # | Bilder Name | Kurzbeschreibung |
| #1 |  |  |
| #2 |  |  |
| #3 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Tabelle # | | |

## Glossar

|  |  |
| --- | --- |
| Wort | Erklärung |
| Fieldset | Fieldset ist ein Java Panel in der Library… |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Tabelle # | |