**User Interfaces 1, HS 2010**

05.12.2010

**BookMasterPro**

**Entwicklungs- und Benutzerdokumentation**

**Christina Heidt & Lukas Elmer**

# Inhalt

[1. Inhalt 1](#_Toc278219190)

[2. Todos 2](#_Toc278219191)

# Szenarien

## «Muss» Funktionalität

### Szenario 1.1 (Buchverfügbarkeit prüfen & Buchstandort finden)

### Szenario 1.2 (Buchtitel hinzufügen)

### Szenario 2.1 (Buch ausleihen)

### Szenario 2.2 (Bücher ausleihen mit überfälligem Buch)

### Szenario 2.3 (Buch ausleihen welches schon ausgeliehen ist)

### Szenario 2.4 (Mehrere Bücher ausleihen)

### Szenario 3.1 (Buch zurückgeben)

### Szenario 3.2 (Buch zurückgeben welches überfällig ist)

## «Kann» Funktionalität

### Szenario 3.3 (Mehrere Bücher zurückgeben)

### Szenario 3.4 (Buch zurückgeben welches nicht der Ausleihe entspricht)

## «Optionale» Funktionalität

### Commands & Undo/Redo

### Help

Um zu demonstrieren, wie die Hilfe aufgerufen werden kann, haben wir eine Action implementier (F1) und ein Menu Eintrag hinzugefügt. Es wurde aber keine Benutzerhilfe geschrieben, deshalb wird, statt dass die Hilfe angezeigt wird, ein Popup angezeigt.

### CellRenderer

Für den “Book Master” hatten wir zuerst gedacht, dass wir einen CellRenderer brauchen, um bestimmte Zeilen einzufärben. Allerdings haben wir dafür einen anderen Weg gefunden, ohne einen CellRenderer zu implementieren:



Bei dieser Lösung hat man den Vorteil, dass nicht für jede einzelne Spalte ein CellRenderer hinzugefügt werden muss. Da in unseren Tabelle sich eine ganze Zeile immer auf ein bestimmtes Objekt bezieht (z.B. auf ein Buch), haben wir diese Lösung bevorzugt.

### CellEditor

#### Cell Editor im „BookMaster“

Um zu zeigen, dass wir den CellRenderer und den CellEditor beherrschen, haben wir als Demo einen CellEditor im BookMaster implementiert:



Wir empfehlen aber explizit, dass dieser CellEditor wieder entfernt wird. Denn es soll vermieden werden, dass ein Benutzer aus Versehen einen Fehler macht und ein Buch bearbeitet (Missgriff), obwohl er es nicht will.

#### Cell Editor im „Buch Detail“, um Exemplar zu bearbeiten

Da das Exemplar nur ein Feld hat, das bearbeitet werden kann, haben wir uns entschieden, dafür einen CellEditor zu verwenden, damit das Exemplar direkt in der Tabelle bearbeitet werden kann:



### Master-Detail View für Kundenverwaltung

### Printing

### Internationalisierung / Mehrsprachigkeit

### Breadcrumbs

### Trashcan für gelöschte Objekte

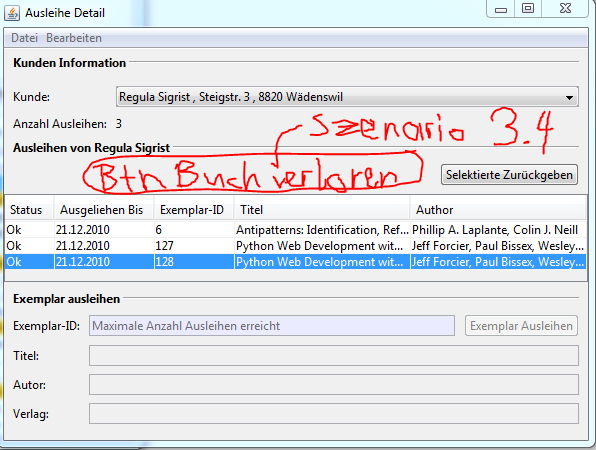
### Java 2D eingesetzt

# Todos

## Prio 1

## Dones

* Ausleihe entfernen 🡪 Tabelle updaten
* Accelerators / Hotkeys im dokumentieren
* Default Actions, Bsp. <Enter> zum Speichern
* F1 + Help Menu
* label.setLabelFor(field) überall implementieren
* Default Selection nach dem Öffnen eines Fensters
* Wenn LoanDetail offen ist und ein Buchtitel im anderen Fenster bearbeitet wird (bsp. Author), dann soll das Buch auch im LoanDetail aktualisiert werden 🡪 Observer Pattern
* Wenn LoanDetail offen ist und ein neuer Kunde hinzugefügt wird im BookMaster, dann soll der neue Kunde auch in der Kundenliste von LoanDetail angezeigt werden 🡪 Observer Pattern
* Wenn LoanDetail offen ist und Kunde im anderen Fenster bearbeitet wird (bsp. Name), dann soll der Name des Kunden auch im LoanDetail aktualisiert werden 🡪 Observer Pattern
* Wenn BookDetail offen ist und eine Copy im anderen Fenster zurückgegeben oder ausgeliehen wird, dann soll sich die Tabelle mit den Loans auch im BookDetail aktualisieren 🡪 Observer Pattern
* Wenn Copy im BookDetail entfernt wird, müssen auch der Loan entfernt werden, sofern Loan vorhanden ist
* Ausleihe Detail 🡪 falls Buch ausgeliehen 🡪 neue Ausleihe nicht möglich (Button deaktiviert)
* Menus implementieren
* Neue Ausleihe erfassen…
* Doppelklick Table
* UIManager
  + Überall implementieren
  + Subframes!
* Kunden 🡪 Ausleihen anzeigen
* BookMaster Loans 🡪 Ausgewählte zurückgeben
* Wizard? Mehrere Bücher zurückgeben 🡪 Nein, besser im AusleiheMaster
* Buch zurückgeben welches nicht der Ausleihe entspricht?
  + Buch verloren 🡪 Liste „lostCopies“ in der Library
  + Idee: Button „Buch verloren“ im Ausleihe Detail entfernt Kopie



## Fragen, ob entwickeln

* Business Logik auch in der Library prüfen und bei Gui-Fehlverhalten Exceptions werfen
  + Muss nicht implementiert werden
* Loan / Buchtitel / Kunde Löschen?
  + Muss nicht implementiert werden

# Accelerators und Mnemonics

# CellEditor und CellRenderer