

# Pflichtenheft

## Urban Garden

Entwicklung eines smarten Schlosses für einen Schlüsselkasten, mit dem der Inhalt und der Zugang zum Kasten über ein Buchungssystem und ein 2 Faktor Authentifizierung System verwaltet werden kann.

ein wirklich sehr gut gelungenes und detailliertes Pflichtenheft. Verweise auf fremde Dokumente sollte man bei Dokumenten an Kunden vermeiden, weil ein online Zugang notwendig ist. Aber hier geht's in Ordnung. Projektplan ist auch gut gemacht, außer der Ressource "Alle". Diese gibt's nicht wirklich.  
weitere kleine Anmerkungen siehe unten.

### Wertung:

PH : 20 Punkte

Projektplan : 5 Punkte

Autor: Ahmed Kutbi, Firas Ben Yedder, Nader Gongi, Heltonn Harold  
Letzte Änderung: 25. Mai 2023  
Dateiname: Pflichtenheft Smartschloss.docx

Version: 1.0

### Copyright

© Ahmed Kutbi, Firas Ben Yedder, Nader Gongi, Heltonn Harold

Die Weitergabe, Vervielfältigung oder anderweitige Nutzung dieses Dokumentes oder Teile davon ist unabhängig vom Zweck oder in welcher Form untersagt, es sei denn, die Rechteinhaber/In hat ihre ausdrückliche schriftliche Genehmigung erteilt.

---

**Version Historie**

Version	Datum	Verantwortlich	Änderung
0.1	07.05.2023	Alle	Initiale Dokumenterstellung
0.2	10.05.2023	Nader Gongi	Erweiterungen
0.3	13.05.2023	Ahmed	Erweiterungen
0.4	14.05.2023	Heltonn	Erweiterungen
0.5	19.05.2023	Firas	Erweiterungen
0.6	20.05.2023	Ahmed, Heltonn	Erweiterungen
0.7	21.05.2023	Firas	Erweiterungen
0.8	22.05.2023	Alle	Bearbeitung
0.9	24.05.2023	Alle	Format und Bearbeitung
<b>1.0</b>	<b>26.05.2023</b>	<b>Alle</b>	<b>Abgabe</b>

## Inhalt

1	Überblick .....	1
2	Hauptziele .....	1
3	Annahmen und Abgrenzungen .....	1
4	Workflow .....	2
5	Funktionalität .....	5
5.1	Überblick - Use Case Diagramm .....	5
5.2	U1: Login.....	6
5.3	U2: User Registrieren .....	6
5.4	U3: Schlüssel buchen .....	7
5.5	U4: Buchung stornieren .....	8
5.6	U5: Verfügbarkeit der Schlüssel überprüfen .....	9
5.7	U6: Bestätigungsmail senden .....	9
5.8	U7: Erinnerungsmail senden .....	10
5.9	U8: Schlüssel verwalten .....	10
5.10	U9: Users verwalten .....	11
5.11	U10: Schlüssel überwachen .....	12
5.12	U11: Zugang zum Schlüsselkasten verwalten .....	13
5.13	U12: Admin verwalten .....	14
5.14	U13: RFID-Karte/ Barcode scannen .....	15
5.15	U14: Kasten öffnen .....	16
5.16	U15: Alarm auslösen/ Warnung-E-Mail senden.....	17
5.17	U16: Insert Userdaten .....	18
6	Offene Fragen.....	21
7	Modulabhängigkeiten.....	22
8	Wer hat was gemacht .....	23

## **Tabellenverzeichnis**

Tabelle 1: Hauptziele .....	1
Tabelle 2: Annahmen .....	1
Tabelle 3: Abgrenzungen .....	1
Tabelle 4: U1: Login .....	6
Tabelle 5: U2: User Registration .....	7
Tabelle 6: U3: Schlüssel buchen .....	8
Tabelle 7: U4: Buchung stornieren .....	8
Tabelle 8: U5: Verfügbarkeit der Schlüssel überprüfen .....	9
Tabelle 9: U6: Bestätigungsmail senden .....	10
Tabelle 10: U7: Erinnerungsmail senden .....	10
Tabelle 11: U8: Schlüssel verwalten .....	11
Tabelle 12: U9: Users verwalten .....	11
Tabelle 13: U10: Schlüssel überwachen .....	13
Tabelle 14: U11: Zugang zum Schlüsselkasten verwalten .....	14
Tabelle 15: U12: Admin verwalten .....	15
Tabelle 16: U13: RFID-Karte/ Barcode scannen .....	16
Tabelle 17: U14: Kasten öffnen .....	17
Tabelle 18: U15: Alarm auslösen/ Warnung Mail senden .....	18
Tabelle 19: U16: Insert Userdaten .....	19
Tabelle 20: Offene Fragen .....	21
Tabelle 21: Modulabhängigkeiten .....	23
Tabelle 22: Wer hat was gemacht .....	23

## **Abbildungsverzeichnis**

Abbildung 1: Buchungsprozess .....	2
Abbildung 2: Schlüssel Abholung/ Abgabe .....	3
Abbildung 3: Admin Funktionen .....	4
Abbildung 4: :Use Case Diagramm .....	5
Abbildung 5: Figma Prototype (GUI-Design) .....	19
Abbildung 6: Kasten/ Fach Design .....	20

## 1 Überblick

Das zu erstellende Modul beinhaltet die Entwicklung eines intelligenten Schlosses für einen Schlüsselkasten. Das Schloss soll über ein Buchungssystem gesteuert werden und der Zugang zur Box soll über den HTW-Studierendenausweis verwaltet werden. Zu den wichtigsten Funktionalitäten gehören die Identifikation des Benutzers durch den HTW-Studentenausweis, die Freigabe des Zugangs zum Fach, das automatische Verschließen des Schlosses nach der Benutzung und die Verwaltung des Buchungssystems. Ziel des Moduls ist es, die Sicherheit und Effizienz der Schlüsselverwaltung an der Hochschule zu verbessern und zukünftige studentische Initiativen zu unterstützen.

## 2 Hauptziele

	Ziel	Beschreibung der Implementation
1	Identifikation des Benutzers durch den HTW-Studentenausweis und Barcode Karte	Lesegerät, Protokoll
2	Automatisches Öffnen und Schließen des Schlosses	Mikrocontroller ESP 32, elektrischer Aktuator
3	Kommunikation zwischen Schloss und Buchungssystem	Protokolle
4	Verwaltung des Buchungssystems	Drupal, Web-GUI
5	Verfügbarkeiten der Schlüssel	Sensor

Tabelle 1: Hauptziele

## 3 Annahmen und Abgrenzungen

Annahmen (fachliche und technische Annahmen)	
1	WLAN ist vorhanden.
2	Schlüsselkasten muss über einen Stromanschluss verfügen.
3	Der Schlüsselkasten muss für alle Benutzer zugänglich sein.
4	HTW-Karte und Barcode Karte müssen vorhanden sein.

Tabelle 2: Annahmen

Abgrenzungen (Was ist in dieser Lösung <b>nicht enthalten</b> bzw. abgedeckt)	
1	Kein Mobile-App.
2	

Tabelle 3: Abgrenzungen



#### 4 Workflow

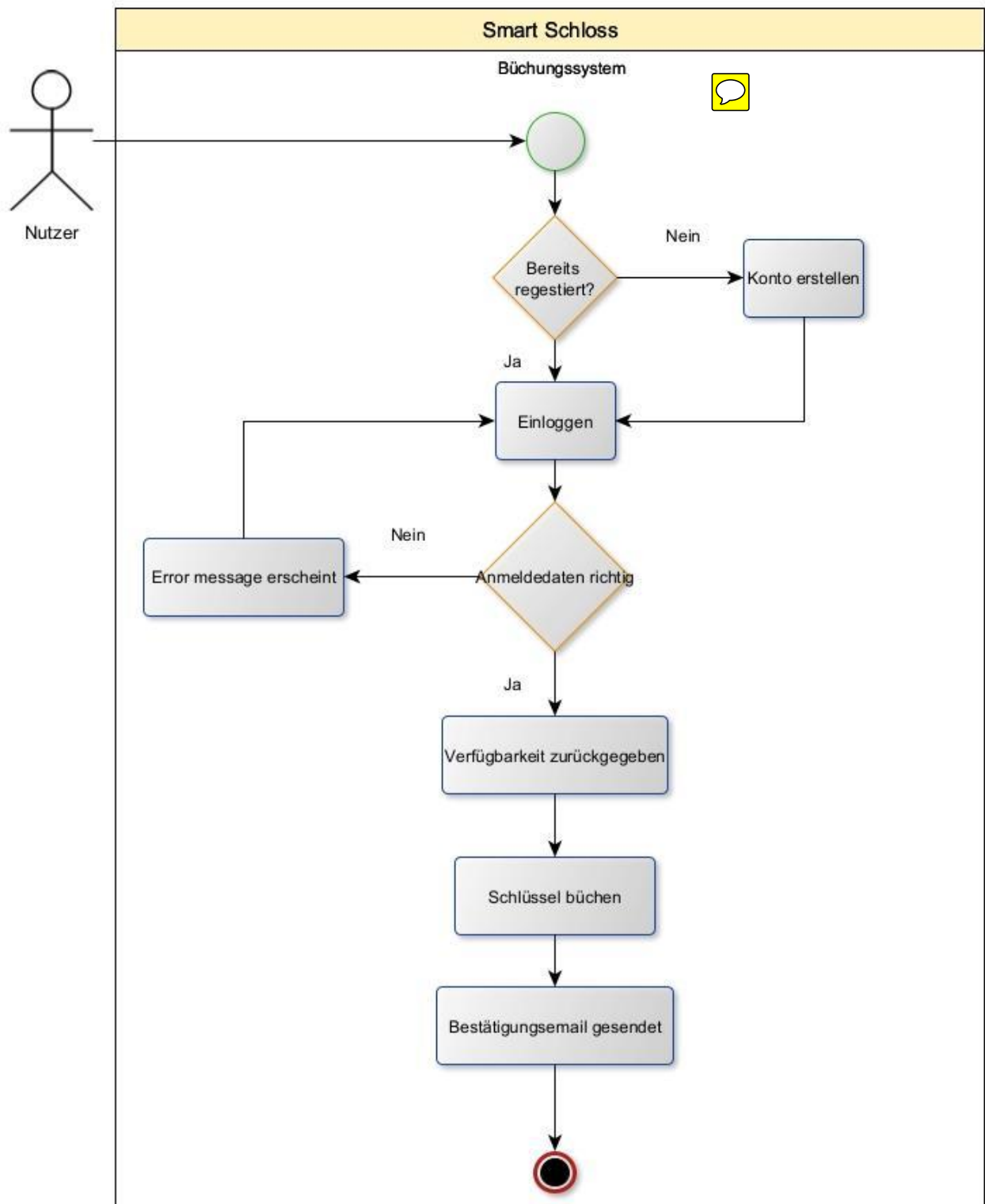


Abbildung 1: Buchungsprozess

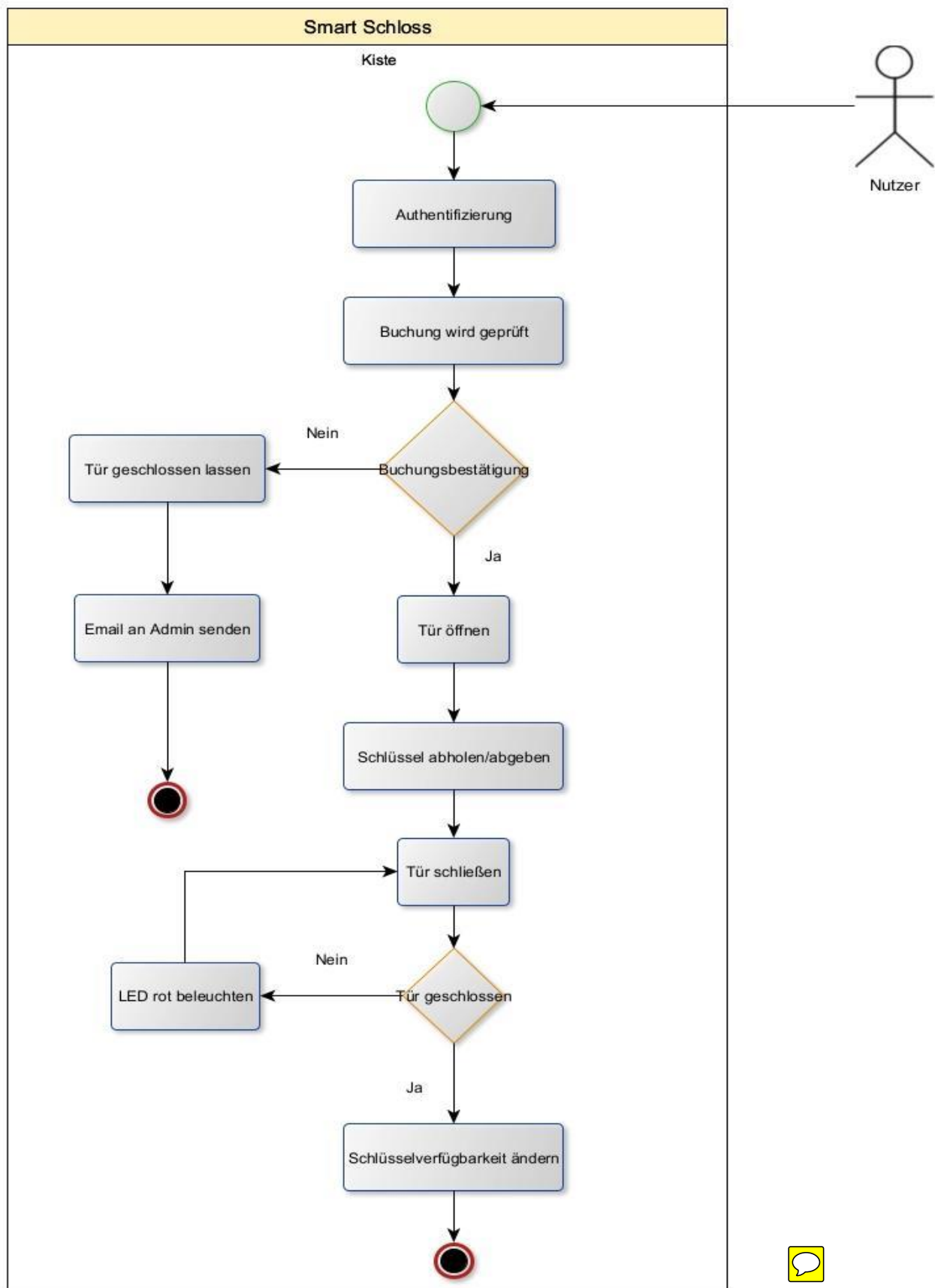


Abbildung 2: Schlüssel Abholung/ Abgabe

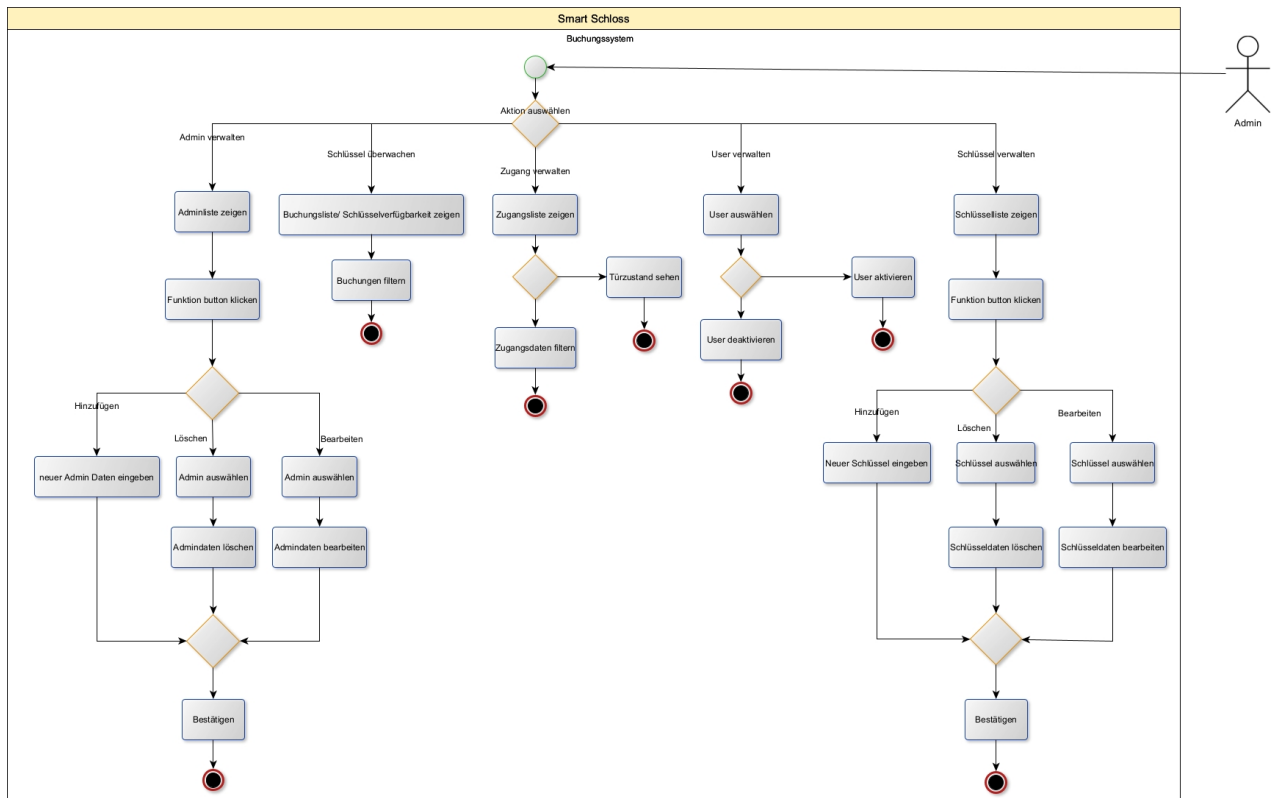


Abbildung 3: Admin Funktionen





## 5 Funktionalität

### 5.1 Überblick - Use Case Diagramm

.svg-datei: [Smart Schloss UCD](#)

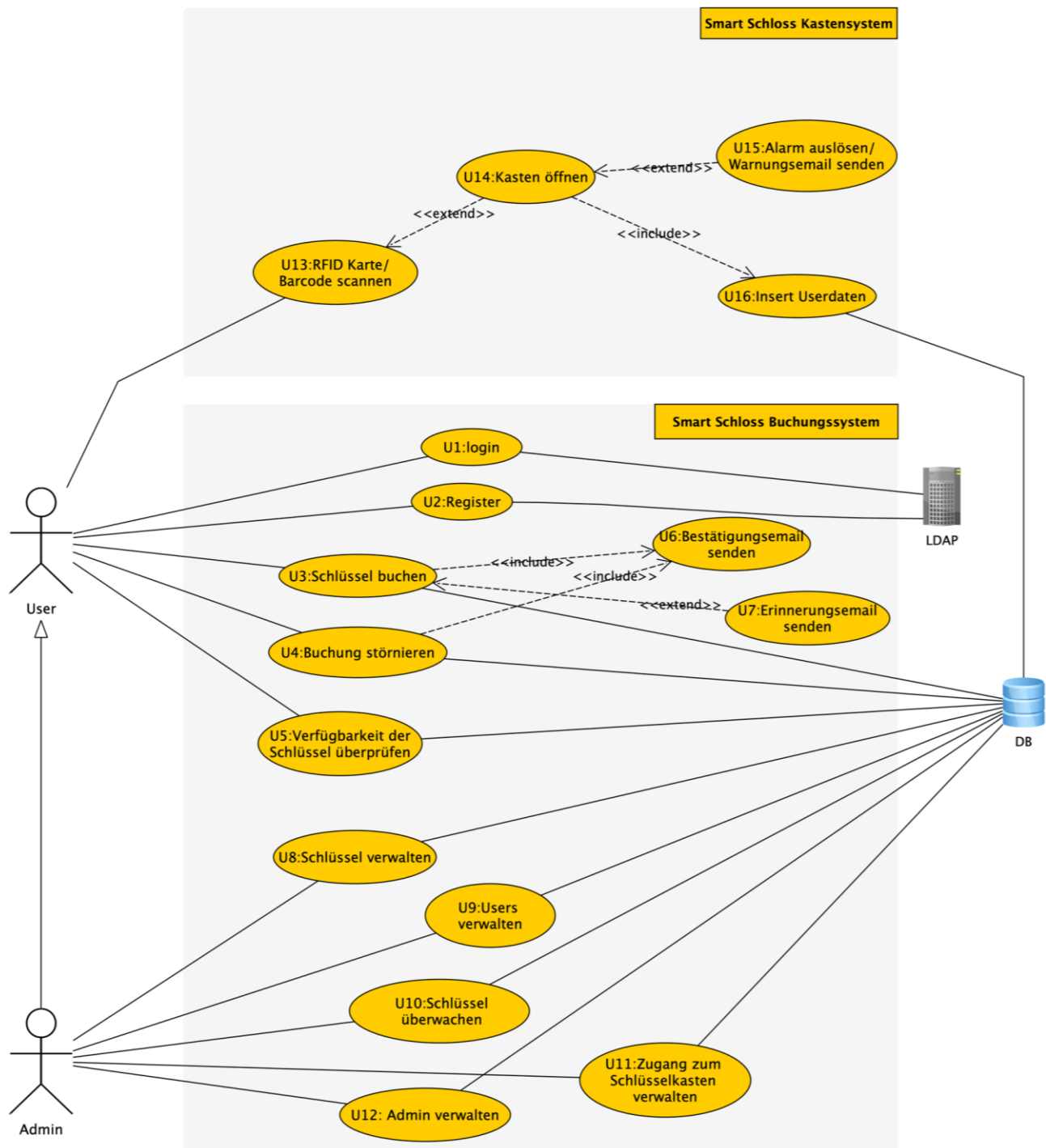


Abbildung 4: Use Case Diagramm



## 5.2 U1: Login

Zweck/Ziel	Mit dieser Funktion kann der Benutzer auf das Webportal zugreifen oder nicht.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• User</li> <li>• Admin</li> </ul>
Berechtigung	Users, die einen HTW-Account besitzen (Studenten, Lehrende...)
WF-Referenz	Buchungsprozess WF
Vorbedingung	<ul style="list-style-type: none"> <li>• Verbindung zum Internet</li> <li>• HTW Vpn (CISCO)</li> <li>• HTW-Account</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>• Benutzername (Matrikelnummer)</li> <li>• Passwort</li> <li>• Schnittstellen: DB und LDAP-Server der HTW</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Eingabe des Benutzernamens und Passworts</li> <li>2. Das System überprüft, ob der Benutzer einen gültigen HTW-Account besitzt und in unserer DB registriert ist.</li> <li>3. Bei erfolgreicher Überprüfung öffnet das System die Webseite und gewährt dem Benutzer Zugriff auf alle Funktionen und Daten der Webseite.</li> </ol>
Ergebnis	Bei erfolgreichem Login hat der User Zugriff auf alle Funktionen und Daten der Webseite.
Plausibilitäten	<ul style="list-style-type: none"> <li>• Validierung, ob der eingegebene Benutzername tatsächlich einem vorhandenen HTW-Account entspricht.</li> <li>• Bei fehlerhafter Eingabe der Login-Daten fordert das System den Benutzer zur erneuten Eingabe auf.</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>• Bei fehlerhafter Eingabe von Benutzernamen oder Passwort wird der Benutzer aufgefordert, die korrekten Daten einzugeben.</li> </ul>
Folgeprozess	Nach dem erfolgreichen Login des Users öffnet sich die Home-Seite.
Anforderung	<b>FA-2.3</b>
Release	Release im Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>• T-U1.1: Erfolgreicher Login: Der Benutzer gibt korrekte Login-Daten ein und erhält Zugriff auf das Webportal.</li> <li>• T-U1.2: Fehlgeschlagener Login: Der Benutzer gibt falsche oder ungültige Login-Daten ein und erhält keinen Zugriff auf das Webportal.</li> </ul>

Tabelle 4: U1: Login

Mockup: [Anmeldung.png](#)



## 5.3 U2: User Registrieren

Zweck/Ziel	Mit dieser Funktion kann sich der User registrieren, falls er noch kein Konto im Buchungssystem hat.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• User</li> <li>• Admin</li> </ul>
Berechtigung	Users, die einen HTW-Account besitzen (Studenten, Lehrende...)
WF-Referenz	Buchungsprozess WF
Vorbedingung	<ul style="list-style-type: none"> <li>• Es liegt keine vorherige Registrierung für den Benutzer vor.</li> <li>• Verbindung zum Internet</li> <li>• HTW Vpn (CISCO)</li> <li>• HTW-Account</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>• Benutzername: Matrikelnummer</li> <li>• Passwort</li> <li>• Name</li> <li>• Schnittstellen: DB und LDAP-Server der HTW</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Der Benutzer gibt seine Registrierungsdaten ein, einschließlich Benutzername, Passwort, und Name</li> <li>2. Es wird gecheckt, ob der User ein gültiges HTW-Konto hat.</li> <li>3. Wenn die Eingaben gültig sind, wird ein neues Benutzerkonto erstellt und die Registrierungsdaten werden im System gespeichert.</li> </ol>



	4. Nach erfolgreicher Aktivierung kann sich der Benutzer mit seinen Registrierungsdaten einloggen und auf alle Funktionen und Daten der Webseite zugreifen.
Ergebnis	Das Ergebnis dieser Funktion ist die erfolgreiche Registrierung des Benutzers, nach der er Zugriff auf alle Funktionen und Daten der Webseite hat.
Plausibilitäten	<ul style="list-style-type: none"> <li>Validierung, ob der eingegebene Benutzername tatsächlich einem vorhandenen HTW-Account entspricht.</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>Bei fehlerhafter Eingabe von Benutzernamen oder Passwort wird der Benutzer aufgefordert, die korrekten Daten einzugeben.</li> </ul>
Folgeprozess	U1: Login
Anforderung	<b>FA-2.3</b>
Release	Release im Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>T-U2.1: Erfolgreiche Registrierung</li> <li>T-U2.2: Fehlermeldung bei fehlerhafter Eingabe von Daten.</li> </ul>

Tabelle 5: U2: User Registration

Mockup: [Registrieren.png](#)

## 5.4 U3: Schlüssel buchen

Zweck/Ziel	Mit dieser Funktion kann der User die Schlüssel für eine bestimmte Zeit buchen
Akteur/Auslöser	<ul style="list-style-type: none"> <li>User</li> <li>Admin</li> </ul>
Berechtigung	Leute, die ein HTW-Konto haben (Studenten, Lehrende...)
WF-Referenz	Buchungsprozess WF
Vorbedingung	<ul style="list-style-type: none"> <li>Verbindung zum Internet</li> <li>Erfolgreicher Login</li> <li>Der Schlüssel muss für den angeforderten Zeitraum verfügbar sein, um eine Buchung durchzuführen.</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>Buchungszeitraum: Der Benutzer muss den Zeitraum angeben, für den er den Schlüssel buchen möchte. Dies kann ein bestimmter Tag oder ein Zeitraum von mehreren Tagen sein.</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>Der Benutzer meldet sich auf dem Web-Portal an und authentifiziert sich erfolgreich.</li> <li>Der Benutzer wählt den gewünschten Schlüssel aus der Liste der verfügbaren Optionen aus.</li> <li>Der Benutzer gibt Details zur Buchung ein, z.B. Datum und Uhrzeit der Buchung.</li> <li>Das System überprüft die Verfügbarkeit des Schlüssels sowie die Gültigkeit der Buchungsdetails.</li> <li>Das Buchungssystem fordert den Benutzer auf, die Buchung zu bestätigen, um sicherzustellen, dass er die Schlüssel tatsächlich buchen möchte.</li> </ol>
Ergebnis	Das System bestätigt dem Benutzer die Buchung und sendet eine Bestätigungsemail
Plausibilitäten	<ul style="list-style-type: none"> <li>Das System muss die Verfügbarkeit des Schlüssels aktualisieren, um sicherzustellen, dass der Schlüssel für andere Benutzer verfügbar ist.</li> <li>Die Bestätigungsmeldung sollte klar und verständlich sein, um dem Benutzer zu bestätigen, dass die Buchung erfolgreich war.</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>Wenn der Benutzer fehlende oder ungültige Buchungsdetails eingibt, z.B. ein ungültiges Datum oder eine ungültige Uhrzeit, sollte das System eine entsprechende Fehlermeldung anzeigen und den Benutzer bitten, die erforderlichen Details zu korrigieren.</li> <li>Wenn der ausgewählte Schlüssel zu einem bestimmten Zeitpunkt bereits reserviert ist, sollte das System dem Benutzer eine Fehlermeldung anzeigen und ihn bitten, einen anderen Zeitpunkt auszuwählen.</li> </ul>
Folgeprozess	U6 und danach Homepage
Anforderung	<b>FA-2.1</b>
Release	Release im Sprint 2

Test-Cases	<ul style="list-style-type: none"> <li>• T-U3.1: Der Benutzer meldet sich auf dem Web-Portal an und bucht einen verfügbaren Schlüssel für einen bestimmten Zeitraum. Die Buchung wird erfolgreich bestätigt</li> <li>• T-U3.2: Ein Benutzer versucht, einen Schlüssel für einen Zeitraum zu buchen, für den keine Schlüssel verfügbar sind. Der Benutzer erhält eine Fehlermeldung mit dem Hinweis, dass keine Schlüssel verfügbar sind und wird aufgefordert, einen anderen Zeitraum zu wählen.</li> </ul>
------------	---

Tabelle 6: U3: Schlüssel buchen

Mockup: [Buchen.png](#)

## 5.5 U4: Buchung stornieren

Zweck/Ziel	Mit dieser Funktion kann der User seine Buchungen stornieren
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• User</li> <li>• Admin</li> </ul>
Berechtigung	Users, die einen HTW-Account haben (Studenten, Lehrende...)
WF-Referenz	—
Vorbedingung	<ul style="list-style-type: none"> <li>• Erfolgreicher Login</li> <li>• Es muss bereits eine bestehende Buchung für den Schlüssel vorliegen, die der Benutzer stornieren möchte, und nur nicht abgeholte Buchungen können storniert werden.</li> <li>• Der Benutzer muss über einen Internetzugang und einen Webbrowser verfügen, um auf das Buchungssystem zugreifen zu können.</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>• BuchungsID</li> <li>• Schnittstelle: DB</li> </ul>
Verarbeitungsschritte	<ul style="list-style-type: none"> <li>• Der Benutzer muss sich zunächst im Buchungssystem authentifizieren, um auf seine Buchungen zugreifen zu können.</li> <li>• Der Benutzer wählt die Buchung aus, die er stornieren möchte, in der Regel durch Eingabe der Buchungs-ID oder durch Auswahl aus einer Liste.</li> <li>• Das Buchungssystem überprüft die ausgewählte Buchung auf Gültigkeit und stellt sicher, dass sie storniert werden kann.</li> <li>• Das Buchungssystem fordert den Benutzer auf, die Stornierung zu bestätigen, um sicherzustellen, dass er die Buchung tatsächlich stornieren möchte.</li> <li>• Wenn der Benutzer die Stornierung bestätigt hat, führt das Buchungssystem die Stornierung durch und ändert den Status der Buchung entsprechend.</li> </ul>
Ergebnis	Das System bestätigt dem Benutzer die Stornierung seiner Buchung und sendet eine Bestätigungsemail
Plausibilitäten	<ul style="list-style-type: none"> <li>• Vor der endgültigen Stornierung sollte das Buchungssystem eine Bestätigung vom Benutzer einholen, um sicherzustellen, dass er die Buchung wirklich stornieren möchte.</li> <li>• Überprüfen, ob die ausgewählte Buchung gültig ist und tatsächlich existiert.</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>• Wenn der Benutzer versucht, eine bereits stornierte Buchung zu stornieren, sollte das Buchungssystem eine Fehlermeldung anzeigen und dem Benutzer mitteilen, dass die Buchung bereits storniert wurde.</li> </ul>
Folgeprozess	U6 und danach Homepage
Anforderung	<b>extra FA</b>
Release	Release im Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>• T-U4.1: Erfolgreiche Stornierung</li> <li>• T-U4.2: Abbruch der Stornierung</li> </ul>

Tabelle 7: U4: Buchung stornieren

Mockup: [Edit Buchung.png](#)

## 5.6 U5: Verfügbarkeit der Schlüssel überprüfen

Zweck/Ziel	Mit dieser Funktion kann der User sehen, welche Schlüssel verfügbar sind.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• User</li> <li>• Admin</li> </ul>
Berechtigung	Users, die einen HTW-Account besitzen (Studenten, Lehrende...)
WF-Referenz	Buchungsprozess
Vorbedingung	<ul style="list-style-type: none"> <li>• Erfolgreicher Login.</li> <li>• Schlüsseldaten sind vorhanden.</li> <li>• Internet vorhanden.</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>• Schlüssel ID und Daten</li> <li>• Schnittstelle: DB</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Der Benutzer authentifiziert sich im System.</li> <li>2. Das System ruft die Informationen zu den verfügbaren Schlüsseln ab.</li> <li>3. Die Informationen werden dem Benutzer angezeigt.</li> </ol>
Ergebnis	Die Anzeige der verfügbaren Schlüssel für den Benutzer
Plausibilitäten	<ul style="list-style-type: none"> <li>• Das System sollte sicherstellen, dass die angezeigten Informationen zu den verfügbaren Schlüsseln aktuell und korrekt sind.</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>• Ungültige Eingabe: Wenn der Benutzer eine ungültige Eingabe macht, beispielsweise eine falsche Schlüssel-ID oder falsche Daten, kann das System eine Fehlermeldung anzeigen und den Benutzer zur erneuten Eingabe auffordern.</li> </ul>
Folgeprozess	Homepage
Anforderung	<b>FA-2.1</b>
Release	Release im Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>• T-U5.1: Keine verfügbaren Schlüssel</li> <li>• T-U5.2: Erfolgreiche Anzeige der verfügbaren Schlüssel</li> </ul>

Tabelle 8: U5: Verfügbarkeit der Schlüssel überprüfen

Mockup: [Verfügbarkeit.png](#)

## 5.7 U6: Bestätigungsmail senden

Zweck/Ziel	Diese Funktion besteht darin, dem Benutzer eine Bestätigungsmail für eine erfolgreiche Buchungs- oder Stornierungsanfrage zu senden.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• System</li> </ul>
Berechtigung	Nur Benutzer, die eine gültige Buchungs- oder Stornierungsanfrage gestellt haben, bekommen eine Mail.
WF-Referenz	Buchungsprozess
Vorbedingung	<ul style="list-style-type: none"> <li>• Erfolgreicher Login.</li> <li>• Benutzer haben eine Buchung oder Stornierung vorgenommen.</li> <li>• Buchungs- oder Stornierungsanfrage wurde erfolgreich bearbeitet.</li> </ul>
Daten-Input	keine
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Das System prüft, ob der Benutzer eine gültige Buchungs- oder Stornierungsanfrage gestellt hat.</li> <li>2. Das System generiert automatisch eine Bestätigungsmail mit den relevanten Informationen zur Buchungs- oder Stornierungsanfrage.</li> <li>3. Die Bestätigungsmail wird an die E-Mail-Adresse des Benutzers gesendet.</li> </ol>
Ergebnis	Der Versand einer Bestätigungsmail an den Benutzer.
Plausibilitäten	<ul style="list-style-type: none"> <li>• Eine E-Mail wird gesendet, solange eine Buchung/Stornierung erfolgreich ist.</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>• bei falscher Eingabe erneute Aufforderung zur Eingabe richtiger Daten</li> <li>• Daten unvollständig in diesem Schritt, Meldung Daten vervollständigen.</li> </ul>
Folgeprozess	Homepage
Anforderung	<b>FA-2.7</b>
Release	Release im Sprint 2

Test-Cases	<ul style="list-style-type: none"> <li>• T-U6.1: Erfolgreicher Versand der Bestätigungsmail</li> <li>• T-U6.2: Fehler beim Versand der Bestätigungsmail</li> </ul>
------------	--

Tabelle 9: U6: Bestätigungsmail senden

## 5.8 U7: Erinnerungsmail senden

Zweck/Ziel	Erinnerungsmails werden gesendet, wenn der User vergisst, einen Schlüssel zurückzugeben.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• System</li> </ul>
Berechtigung	Nur Benutzer, die eine gültige Buchungs- oder Stornierungsanfrage gestellt haben, bekommen eine Mail.
WF-Referenz	
Vorbedingung	<ul style="list-style-type: none"> <li>• Es wurde erkannt, dass der Benutzer einen Schlüssel noch nicht zurückgegeben hat.</li> </ul>
Daten-Input	Keine
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Das System identifiziert den Benutzer, der den Schlüssel rechtzeitig nicht zurückgegeben hat.</li> <li>2. Das System generiert automatisch eine Erinnerungsmail mit relevanten Informationen zur Schlüssel-Rückgabe.</li> <li>3. Die Erinnerungsmail wird an die E-Mail-Adresse des Benutzers gesendet.</li> </ol>
Ergebnis	Das Ergebnis dieser Funktion ist der Versand einer Erinnerungsmail an den Benutzer.
Plausibilitäten	<ul style="list-style-type: none"> <li>• Überprüfung, ob der Benutzer existiert und eine gültige E-Mail-Adresse hinterlegt hat.</li> <li>• Überprüfung, ob der Benutzer tatsächlich noch ausstehende Schlüssel hat.</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>• Falls der Benutzer nicht gefunden wird oder keine gültige E-Mail-Adresse hinterlegt hat, wird eine Fehlermeldung generiert.</li> </ul>
Folgeprozess	Homepage
Anforderung	<b>FA-2.7</b>
Release	Release im Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>• T-U7.1: Ein Benutzer hat ausstehende Schlüssel und erhält eine Erinnerungsmail.</li> <li>• T-U7.2: Ein Benutzer hat keine ausstehenden Schlüssel und erhält keine Erinnerungsmail.</li> </ul>

Tabelle 10: U7: Erinnerungsmail senden

## 5.9 U8: Schlüssel verwalten

Zweck/Ziel	Die Funktion ermöglicht es dem Administrator, Schlüsselinformationen zu verwalten.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• Admin</li> </ul>
Berechtigung	Der Benutzer muss ein Admin sein, der an der HTW teilnimmt, um diese Funktion auszuführen
WF-Referenz	Admin Funktionen
Vorbedingung	<ul style="list-style-type: none"> <li>• Erfolgreicher Login als Admin.</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>• Schlüsselname/ID: Informationen zum Schlüssel, der verwaltet werden soll.</li> <li>• Aktion: Die gewünschte Aktion, z.B. "Schlüssel hinzufügen", "Schlüssel löschen", "Schlüssel bearbeiten".</li> <li>• Schnittstelle: GUI - DB</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Der Administrator meldet sich erfolgreich auf dem Webportal an.</li> <li>2. Der Administrator navigiert zur Funktion "Schlüssel verwalten".</li> <li>3. Je nach ausgewählter Aktion führt der Administrator die entsprechenden Schritte aus: <ul style="list-style-type: none"> <li>• Schlüssel hinzufügen: Der Administrator gibt die erforderlichen Informationen zum neuen Schlüssel ein und fügt ihn in der Schlüsseltabelle in der DB hinzu.</li> <li>• Schlüssel löschen: Der Administrator wählt den zu löschenden Schlüssel aus der Liste aus und bestätigt die Löschung.</li> </ul> </li> </ol>



	<ul style="list-style-type: none"> <li>• Schlüssel bearbeiten: Der Administrator wählt den zu bearbeitenden Schlüssel aus der Liste aus und aktualisiert die entsprechenden Informationen.</li> </ul>
Ergebnis	<ul style="list-style-type: none"> <li>• Schlüssel hinzufügen: Der neue Schlüssel ist in der Schlüsselliste verfügbar.</li> <li>• Schlüssel löschen: Der gelöschte Schlüssel ist nicht mehr in der Schlüsselliste vorhanden.</li> <li>• Schlüssel bearbeiten: Die aktualisierten Informationen des bearbeiteten Schlüssels sind in der Schlüsselliste sichtbar.</li> </ul>
Plausibilitäten	<ul style="list-style-type: none"> <li>• Überprüfung, ob der Schlüssel ID eine gültige Eingabe enthält</li> <li>• Überprüfung, ob der Schlüssel ID eindeutig ist</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>• Bei fehlerhaften oder ungültigen Eingaben werden entsprechende Fehlermeldungen angezeigt, und der Administrator wird zur erneuten Eingabe aufgefordert.</li> </ul>
Folgeprozess	Homepage
Anforderung	<b>FA-2.3</b>
Release	Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>• T-U8.1: Überprüfen, ob ein neuer Schlüssel erfolgreich hinzugefügt wird.</li> <li>• T-U8.2: Überprüfen, ob ein neuer Schlüssel erfolgreich gelöscht wird.</li> <li>• T-U8.3: Überprüfen, ob ein neuer Schlüssel erfolgreich bearbeitet wird.</li> <li>• T-U8.4: Überprüfen, ob das System fehlerhafte Eingaben korrekt behandelt.</li> </ul>

Tabelle 11: U8: Schlüssel verwalten

Mockup: [Schlüssel + Verwaltung list.png](#)

## 5.10 U9: Users verwalten

Zweck/Ziel	Diese Funktion ermöglicht es dem Administrator, Benutzerkonten zu verwalten und verschiedene Aktionen im Zusammenhang mit Benutzern durchzuführen.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• Admin</li> </ul>
Berechtigung	Der Benutzer muss ein Admin sein, der über ein HTW-Konto verfügt, um diese Funktion auszuführen.
WF-Referenz	Admin Funktionen
Vorbedingung	<ul style="list-style-type: none"> <li>• Erfolgreicher Login als Administrator.</li> <li>• Internet</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>• Benutzername: Der Administrator gibt den Benutzernamen des betreffenden Benutzers ein.</li> <li>• Der Admin kann über einen Button einen User blockieren.</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Der Admin authentifiziert sich im System als Administrator.</li> <li>2. Der gibt den Benutzernamen des betreffenden Benutzers ein oder wählt den User aus der Liste.</li> <li>3. Der Administrator kann verschiedene Aktionen ausführen, z.B. Aktivieren / Deaktivieren des Benutzerkontos.</li> </ol>
Ergebnis	Die gewünschten Aktionen im Zusammenhang mit dem Benutzerkonto werden vom Administrator durchgeführt.
Plausibilitäten	<ul style="list-style-type: none"> <li>• Die ausgewählte Aktion muss erfolgreich auf den ausgewählten User richtig ausgeführt werden.</li> </ul>
Fehlerhandling	<ul style="list-style-type: none"> <li>• Wenn der eingegebene Benutzername ungültig ist oder das Benutzerkonto nicht gefunden wird, zeigt das System eine entsprechende Fehlermeldung an.</li> </ul>
Folgeprozess	Homepage
Anforderung	<b>Extra FA</b>
Release	Release im Sprint 2
Test-Cases	<p>T-U9.1: Test, ob das Userkonto richtig ist.</p> <p>T-U9.2: Test, ob die ausgewählte Aktion richtig ausgeführt werden kann</p> <p>T-U9.3: Test, ob das Aktivieren/Deaktivieren erfolgreich ist.</p>

Tabelle 12: U9: Users verwalten

Mockup: [Users Verwaltung.png](#)

### 5.11 U10: Schlüssel überwachen

Zweck/Ziel	Der Admin kann über das Buchungssystem wissen, welcher Schlüssel aktuell verfügbar/nicht verfügbar ist, für welchen Raum/Schloss jeder Schlüssel ist und wann jeder Schlüssel abgeholt/zurückgegeben wird.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>Admin</li> </ul>
Berechtigung	Der Benutzer muss ein Admin sein, der über ein HTW-Konto verfügt, um diese Funktion auszuführen.
WF-Referenz	Admin Funktionen
Vorbedingung	<ul style="list-style-type: none"> <li>Der Admin soll ein Teilnehmer der HTW Berlin sein.</li> <li>Der Admin muss mit dem Internet verbunden und im Buchungssystem eingeloggt sein.</li> <li>Alle Schlüssel müssen mit ID und anderen Daten initialisiert sein.</li> <li>Die Sensordaten in der Datenbank sollten bereits durch das Kastensystem aktualisiert sein.</li> <li>Die anderen Funktionen sollten nicht in Bearbeitung sein, wenn diese Funktion aufgerufen wird (z.B. Speichern von Buchungsdaten...).</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>Sensordaten zur Erkennung, ob der Schlüssel da ist, oder nicht und wann ein Schlüssel abgeholt/zurückgegeben wurde.</li> <li>Die Schlüsseldaten sind in einer Datenbank gespeichert. Jeder Schlüssel hat ein ID (INT). Zeitstempel hat den Datentyp (DATETIME) und der Status der Schlüssel hat den Datentyp (ENUM ('available', 'not available')).</li> <li>Notwendige Schnittstelle: DB mit Schlüssel-/Sensordaten.</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>Wenn der Admin im System eingeloggt ist, klickt er auf dem Navbar auf Schlüssel überwachen.</li> <li>Das Buchungssystem ruft die Daten aus der Datenbank auf.</li> <li>Es öffnet sich eine neue Seite und dort kann der Admin in Form einer Tabelle sehen, alle Schlüssel mit ihren Daten (ID, Schloss/Raum, Platz in Kasten...), welcher Schlüssel verfügbar/nicht verfügbar ist und wann der Schlüssel abgeholt wurde und wann er zurückgegeben wird.</li> <li>Der Admin kann auch das Ergebnis nach (Raum, Abholzeit, Rückgabezeit, ID...) filtern.</li> <li>Er kann auch das Ergebnis über einen Button neu laden und aktualisieren.</li> </ol>
Ergebnis	Eine Tabelle mit den Daten der Schlüssel und aktuellen Sensordaten aus dem Kastensystem. Diese Daten sind als Werte in den Tabellen einer DB gespeichert.
Plausibilitäten	<ul style="list-style-type: none"> <li>Alle Daten eines Schlüssels müssen gezeigt werden.</li> <li>Alle gezeigten Daten müssen aktuell sein.</li> <li>Alle Sensordaten müssen richtig sein.</li> </ul>
Fehler Handling	<ul style="list-style-type: none"> <li>Bei falscher Ausgabe der Daten kann der Admin die Seite neu laden.</li> <li>Wenn alle Daten nicht gezeigt werden, wird eine Fehlermeldung im Log für Bugfixes und Vervollständigen gespeichert.</li> <li>Bei nicht aktuellen Sensordaten wird eine E-Mail an den Vertreter des Kastensystems gesendet.</li> </ul>
Folgeprozess	Verschiedene Funktionen können über das GUI danach ausgeführt werden.
Anforderung	<b>FA-2.2</b>
Release	Release im Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>T-U10.1: Test, ob normale Benutzer keinen Zugriff auf diese Funktion haben.</li> <li>T-U10.2: Test, ob Admins Zugriff auf diese Funktion haben.</li> <li>T-U10.3: Test, dass die Webseite dieser Funktion nicht ohne Login über eine direkte URL-Eingabe sich öffnet.</li> <li>T-U10.4: Test, ob die Sensordaten in der DB mit dem aktuellen Zustand in dem Kasten übereinstimmen.</li> <li>T-U10.5: Test, ob die gezeigten Daten auf der Webseite mit den Daten in der DB übereinstimmen.</li> <li>T-U10.6: Test, ob alle Schlüssel-Daten in der DB vollständig sind.</li> <li>T-U10.7: Test, ob die vorherige Funktion bis zum Ende durchgelaufen ist, bevor diese Funktion startet.</li> </ul>



Tabelle 13: U10: Schlüssel überwachen

(Siehe GUI von U11)

## 5.12 U11: Zugang zum Schlüsselkasten verwalten

Zweck/Ziel	Diese Funktion ermöglicht, dass der Admin sieht, wann und von wem der Kasten geöffnet wurde. Außerdem kann er sehen, ob der Kasten aktuell zu oder auf ist und wer den Kasten auf Vergessen hat. Er kann auch alle Buchungen, die gemacht wurden, sehen.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>Admin</li> </ul>
Berechtigung	Der Benutzer muss ein Admin sein, der über ein HTW-Konto verfügt, um diese Funktion auszuführen.
WF-Referenz	Admin Funktionen
Vorbedingung	<ul style="list-style-type: none"> <li>Der Admin soll ein Teilnehmer der HTW Berlin sein.</li> <li>Der Admin muss mit dem Internet verbunden und im Buchungssystem eingeloggt sein.</li> <li>Der Zeitpunkt des Öffnens des Schlüsselkastens sollte bereits erfasst und in der Datenbank gespeichert sein.</li> <li>Der aktuelle Zustand der Kastentür muss live als Telemetrie gesendet werden.</li> <li>Die User, die den Kasten geöffnet haben, müssen identifiziert und in der DB gespeichert sein.</li> <li>Die anderen Funktionen sollten nicht in Bearbeitung sein, wenn diese Funktion aufgerufen wird (z.B. Speichern von Buchungsdaten...).</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>Türzustand als Sensordaten in Form von Telemetrie zu erkennen, ob die Tür auf/zu ist.</li> <li>Userdaten, um zu erkennen, welcher User den Kasten geöffnet hat.</li> <li>Zeitstempel, um zu erkennen, wann der Kasten geöffnet wurde.</li> <li>Datentypen: <ul style="list-style-type: none"> <li>Türzustand: String Telemetrie</li> <li>Userdaten: (ID/Matrikelnummer(INT), Name(VARCHAR), Email(VARCHAR)...) -</li> <li>Zeitstempel: Date time.</li> </ul> </li> <li>Notwendige Schnittstelle: DB und Kommunikationsprotokoll (HTTP) zwischen dem Kastensystem und der Webseite.</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>Wenn der Admin im System eingeloggt ist, klickt er auf dem Navbar auf Kasten-Zugang verwalten.</li> <li>Das Buchungssystem ruft die Daten aus der Datenbank auf.</li> <li>Es öffnet sich eine neue Seite und dort kann der Admin in Form einer Tabelle sehen, welcher User den Kasten geöffnet hat und wann das gemacht wurde. Der Admin kann auch dort sehen, ob die Tür momentan geschlossen oder nicht geschlossen ist.</li> <li>Der Admin kann auch das Ergebnis nach (User-ID, User-Name, Zeitstempel) filtern.</li> <li>Er kann auch das Ergebnis über einen Button neu laden und aktualisieren.</li> <li>Der Admin kann auch dort sehen, welcher User die Tür auf Vergessen hat und ihm eine Nachricht schreiben.</li> </ol>
Ergebnis	Eine Tabelle mit den Daten des Schlüsselkastens und des Users. Außerdem eine Signaldarstellung in Form von Text oder grafisch, was dem Admin zeigt, ob der Kasten auf/zu ist.
Plausibilitäten	<ul style="list-style-type: none"> <li>Alle Daten aus dem Kastensystem müssen gezeigt werden.</li> <li>Die User, die den Kasten benutzt haben, müssen in DB gespeichert und angezeigt werden.</li> <li>Die Kommunikation zwischen dem Kastensystem und der Webseite muss zuverlässig sein.</li> <li>Der aktuelle Zustand der Kastentür muss live als Telemetrie gesendet werden.</li> </ul>

Fehler Handling	<ul style="list-style-type: none"> <li>• Bei falscher Ausgabe der Daten kann der Admin die Seite neu laden.</li> <li>• Wenn alle Daten nicht gezeigt werden, wird eine Fehlermeldung im Log für Bugfixes und Vervollständigen gespeichert.</li> <li>• Bei nicht richtigen Daten aus dem Kastensystem wird eine E-Mail an den Vertreter des Kastensystems gesendet.</li> </ul>
Folgeprozess	Verschiedene Funktionen können über das GUI danach ausgeführt werden.
Anforderung	<b>FA-2.5 &amp; FA-2.6</b>
Release	Release im Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>• T-U11.1: Test, ob normale Benutzer keinen Zugriff auf diese Funktion haben.</li> <li>• T-U11.2: Test, ob Admins Zugriff auf diese Funktion haben.</li> <li>• T-U11.3: Test, dass die Webseite dieser Funktion nicht ohne Login über eine direkte URL-Eingabe sich öffnet.</li> <li>• T-U11.4: Test, ob die Daten aus dem Kastensystem in der DB mit dem aktuellen Zustand in dem Kasten übereinstimmen.</li> <li>• T-U11.5: Test, ob die gezeigten Daten auf der Webseite mit den Daten in der DB übereinstimmen.</li> <li>• T-U11.6: Test, ob die gezeigten Users registriert sind.</li> <li>• T-U11.7: Test, ob die Telemetrie mit dem aktuellen Zustand im Kastensystem übereinstimmen.</li> <li>• T-U11.8: Test, ob der Zeitstempel richtig ist.</li> <li>• T-U11.9: Test, ob die vorherige Funktion bis zum Ende durchgelaufen ist, bevor diese Funktion startet.</li> </ul>

Tabelle 14: U11: Zugang zum Schlüsselkasten verwalten

Mockup: [Kasten Überwachung Dashboard.png](#)

### 5.13 U12: Admin verwalten

Zweck/Ziel	Diese Funktion ermöglicht, dass der Admin andere Admins über die Plattform hinzufügt. Der Admin kann auch die Daten von den anderen Admins verändern oder die anderen Admins löschen.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• Admin</li> </ul>
Berechtigung	Der Benutzer muss ein Admin sein, der über ein HTW-Konto verfügt, um diese Funktion auszuführen.
WF-Referenz	Admin Funktionen
Vorbedingung	<ul style="list-style-type: none"> <li>• Der Admin soll ein Teilnehmer der HTW Berlin sein.</li> <li>• Der Admin muss mit dem Internet verbunden und im Buchungssystem eingeloggt sein.</li> <li>• Der Admin muss in der Admin-Tabelle in der DB gespeichert sein.</li> <li>• Die anderen Funktionen sollten nicht in Bearbeitung sein, wenn diese Funktion aufgerufen wird (z.B. Speichern von Buchungsdaten...).</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>• Admins Daten (ID, Name, Beitrittsdatum, Matrikelnummer).</li> <li>• Datentypen: ID: INT, Name: String, Beitrittsdatum: Date, Matrikelnummer: String</li> <li>• Schnittstellen: DB</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Wenn der Admin im System eingeloggt ist, klickt er auf dem Navbar auf Admin Verwalten.</li> <li>2. Er öffnet sich eine Seite mit einer Liste allen Admins und Ihre Daten.</li> <li>3. Der Admin hat dann die Möglichkeit, einen neuen Admin hinzuzufügen, die Daten von einem Admin zu bearbeiten oder die Admin-Daten zu löschen.</li> <li>4. Falls der Admin einen neuen Admin hinzufügen möchte, klickt er auf den entsprechenden Button, danach öffnet sich die Seite, auf der die Daten von dem neuen Admin eingetippt werden können.</li> <li>5. Falls der Admin die Daten eines alten Admins bearbeiten möchte, klickt er auf den entsprechenden Button, danach öffnet sich die Seite, auf der die Daten von dem alten Admin stehen und bearbeitet werden können.</li> </ol>

	<p>6. Falls der Admin einen alten Admin löschen möchte, klickt er auf den entsprechenden Button, danach öffnet sich die Seite, auf der die Daten von dem alten Admin stehen, und er soll das Löschen bestätigen.</p> <p>7. Beim Löschen und Bearbeiten soll ein anderer zweiter Admin diesen Vorgang bestätigen, um mögliche Fehler zu verhindern.</p>
Ergebnis	(Insert, Select, Update oder Delete) - Funktion in der DB
Plausibilitäten	<ul style="list-style-type: none"> <li>• Alle Admin-Daten in der DB müssen gezeigt werden.</li> <li>• Alle Daten in der DB müssen vollständig sein.</li> <li>• Die neuen Admin-Daten müssen richtig gespeichert werden.</li> <li>• Wenn ein zweiter Admin das Löschen oder Bearbeiten der Admin-Daten nicht bestätigt, müssen die Daten in der DB nicht geändert werden.</li> </ul>
Fehler Handling	<ul style="list-style-type: none"> <li>• Bei falscher Ausgabe der Daten kann der Admin die Seite neu laden.</li> <li>• Wenn alle Daten nicht gezeigt werden, wird eine Fehlermeldung im Log für Bugfixes gespeichert.</li> </ul>
Folgeprozess	Verschiedene Funktionen können über das GUI danach ausgeführt werden.
Anforderung	<b>Extra FA</b>
Release	Release im Sprint 2
Test-Cases	<ul style="list-style-type: none"> <li>• T-U12.1: Test, ob normale Benutzer keinen Zugriff auf diese Funktion haben.</li> <li>• T-U12.2: Test, ob Admins Zugriff auf diese Funktion haben.</li> <li>• T-U12.3: Test, dass die Webseite dieser Funktion nicht ohne Login über eine direkte URL-Eingabe sich öffnet.</li> <li>• T-U12.4: Test, ob die gezeigten Daten auf der Webseite mit den Daten in der DB übereinstimmen.</li> <li>• T-U12.5: Test, ob der neue Admin hinzugefügt wird.</li> <li>• T-U12.6: Test, ob Löschen/Bearbeiten ohne Bestätigung nicht erfolgreich ist.</li> <li>• T-U12.7: Test, ob die vorherige Funktion bis zum Ende durchgelaufen ist, bevor diese Funktion startet.</li> </ul>

Tabelle 15: U12: Admin verwalten

Mockup: [Admin Verwaltung.png](#)

## 5.14 U13: RFID-Karte/ Barcode scannen

Zweck/Ziel	Diese Funktion ermöglicht, dass der registrierte User seine HTW-Karte und einen ID-Barcode scannt, um die Tür des Kastensystems zu öffnen und sich anzumelden.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>• User</li> <li>• Admin</li> </ul>
Berechtigung	Usern und Admins, die über HTW-Konto verfügen, können diese Funktion benutzen.
WF-Referenz	Schlüssel Abholung / Schlüssel Abgabe
Vorbedingung	<ul style="list-style-type: none"> <li>• Das Kastensystem muss mit dem Internet verbunden sein.</li> <li>• Das Kastensystem muss über Strom verfügen.</li> <li>• Der Admin/User soll ein Teilnehmer der HTW Berlin sein.</li> <li>• Der Admin/User soll in der DB registriert sein.</li> <li>• Die Kastentür muss geschlossen sein.</li> <li>• Eine Buchung eines Schlüssels muss vorhanden sein</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>• Eine HTW-Karte, die einen RFID-Chip enthält</li> <li>• Ein Barcode, der den User identifiziert und ermöglicht den 2 Faktor Authentifizierung.</li> <li>• Userdaten aus der Datenbank, um zu wissen, ob der User berechtigt ist.</li> <li>• Signal vom Sensor an der Tür, um zu wissen, ob die Tür zu ist.</li> <li>• Datentypen: -</li> <li>• Türzustand: Digital Input (DI) - Signal (HIGH/LOW).</li> <li>• Userdaten: (ID(String), ID-Barcode (INT), gebuchter Schlüssel-ID(INT)).</li> <li>• Barcode/ RFID-Karte: Barcode (INT) /UID als HEX-String.</li> </ul>

Verarbeitungsschritte	<ul style="list-style-type: none"> <li>Notwendige Schnittstelle: DB, CAM, RFID-Leser, Türsensor</li> </ul> <ol style="list-style-type: none"> <li>Es wird geprüft, ob die Tür geschlossen ist.</li> <li>Der User stellt seine Karte vor.</li> <li>Der User stellt dann seinen Barcode vor.</li> <li>Das System ruft die Daten aus der Datenbank auf.</li> <li>Es wird verglichen, ob der User mit der HTW-Karte registriert ist.</li> <li>Es wird auch verglichen, ob der vorgestellte Barcode übereinstimmt.</li> <li>Es wird gecheckt, ob der User eine Buchung gemacht hat.</li> <li>Wenn einer der oben aufgeführten Vergleiche nicht erfolgreich ist, wird der Vorgang abgebrochen.</li> <li>Der Zugriffsversuch wird aufgezeichnet und in der DB gespeichert.</li> <li>Falls dieser Funktion erfolgreich ist folgt Funktion U12</li> </ol>
Ergebnis	Speicherung der Zugriffsversuche und folgt die Funktion U12
Plausibilitäten	<ul style="list-style-type: none"> <li>Die RFID-Karte/Barcode müssen richtig gelesen werden.</li> <li>Die Sensordaten müssen richtig sein.</li> <li>Alle Vergleiche müssen nacheinander durchlaufen und wenn ein Vergleich nicht erfolgreich ist, soll die Überprüfung abbrechen.</li> <li>Der Versuch muss in der Datenbank gespeichert werden.</li> </ul>
Fehler Handling	<ul style="list-style-type: none"> <li>Bei einem erfolglosen Versuch wird der User z.B. über LED/LCD benachrichtigt.</li> <li>Wenn der Kasten falsche Daten ausgibt, wird alles im Log aufgezeichnet und der Vertreter wird benachrichtigt.</li> <li>Wenn eine Schnittstelle nicht funktioniert, wird das System ausgeschaltet, bis der Fehler beseitigt wird.</li> <li>Wenn die Tür nicht zu ist, muss der User sie einfach schließen.</li> </ul>
Folgeprozess	U12 folgt diese Funktion
Anforderung	<b>FA-1</b>
Release	Release im Sprint 1
Test-Cases	<ul style="list-style-type: none"> <li>T-U13.1: Test, ob RFID-Leser richtig liest.</li> <li>T-U13.2: Test, ob die Kamera den Barcode aufnimmt.</li> <li>T-U13.3: Test, ob der Türsensor die richtigen Daten zurückgibt.</li> <li>T-U13.4: Test, ob die aus der Datenbank aufgerufenen Daten richtig sind.</li> <li>T-U13.5: Test, ob alle Vergleiche richtig ablaufen.</li> <li>T-U13.6: Test, ob der Versuch in der DB gespeichert wird.</li> <li>T-U13.7: Test, ob die Funktion U12 nach dieser Funktion ausgeführt wird.</li> <li>T-U13.8: Test, ob der Vorgang abbricht, wenn ein Vergleich erfolglos ist.</li> </ul>

Tabelle 16: U13: RFID-Karte/ Barcode scannen

## 5.15 U14: Kasten öffnen

Zweck/Ziel	Diese Funktion wird durch das Kastensystem ausgelöst. Sie beschreibt den Vorgang, wenn die Tür sich öffnet und was im Hintergrund passiert.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>Kastensystem</li> </ul>
Berechtigung	Eine erfolgreiche U11-Funktion löst diese Funktion aus und die gleiche Berechtigung für U11 gilt hier.
WF-Referenz	Schlüssel Abholung / Schlüssel Abgabe
Vorbedingung	<ul style="list-style-type: none"> <li>Die Kastentür muss geschlossen sein.</li> <li>Das Kastensystem muss mit dem Internet verbunden sein.</li> <li>Das Kastensystem muss über Strom verfügen.</li> <li>U11 soll erfolgreich durchgelaufen sein.</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>Ein Signal vom Sensor an der Tür, um zu wissen, ob die Tür zu ist.</li> <li>Datentypen: Türzustand: Digital Input (DI) - Signal (HIGH/LOW).</li> <li>Notwendige Schnittstelle: Türsensor</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>Es wird geprüft, ob die Tür geschlossen ist.</li> <li>Es wird gecheckt, dass die U11 erfolgreich war.</li> </ol>

	<ol style="list-style-type: none"> <li>Falls diese Funktion erfolgreich ist, öffnet sich das Schloss des Kastens</li> <li>Falls das Schloss immer noch zu ist, wird Schritt 3 mehrmals wiederholt</li> <li>Die Funktion U14 wird danach ausgelöst.</li> <li>Alternativ folgt die Funktion U13, wenn etwas erfolglos ist.</li> </ol>
Ergebnis	Entweder eine geöffnete Tür und die Funktion U14 oder die Funktion U13.
Plausibilitäten	<ul style="list-style-type: none"> <li>Die Sensordaten müssen richtig sein.</li> <li>Das Schloss muss sich öffnen und es muss gecheckt werden, dass das Schloss wirklich auf ist.</li> <li>U13 oder U14 muss nach dieser Funktion folgen.</li> </ul>
Fehler Handling	<ul style="list-style-type: none"> <li>Bei einer erfolglosen Durchführung wird der User z.B. über LED/LCD benachrichtigt.</li> <li>Wenn der Kasten falsche Daten ausgibt oder das Folgeprozess diese Funktion nicht folgt, wird alles im Log aufgezeichnet und der Vertreter wird benachrichtigt.</li> <li>Wenn das Schloss nach allen Versuchen nicht funktioniert, wird das System ausgeschaltet, bis der Fehler beseitigt wird.</li> <li>Wenn die Tür nicht zu ist, muss der User sie einfach schließen.</li> </ul>
Folgeprozess	Entweder U14 oder U13.
Anforderung	<b>FA-1</b>
Release	Release im Sprint 1
Test-Cases	<ul style="list-style-type: none"> <li>T-U14.1: Test, ob der Türsensor die richtigen Daten zurückgibt.</li> <li>T-U14.2: Test, ob die Funktion U11 erfolgreich durchgelaufen ist.</li> <li>T-U14.3: Test, ob das Schloss sich öffnet.</li> <li>T-U14.4: Test, ob es wiederholt wird, wenn das Schloss zu bleibt.</li> <li>T-U14.5: Test, ob U13 folgt, wenn etwas erfolglos ist.</li> <li>T-U14.6: Test, ob U14 ausgeführt wird, wenn alles erfolgreich durchläuft.</li> </ul>

Tabelle 17: U14: Kasten öffnen

### 5.16 U15: Alarm auslösen/ Warnung-E-Mail senden

Zweck/Ziel	Diese Funktion wird ausgelöst, wenn U12 erfolglos ist. Das heißt, auch wenn U11 erfolglos ist, wird diese Funktion durchlaufen. Ziel ist das, wenn ein von einiger Arbeitsschritte erfolglos ist oder ein Manipulationsversuch gemacht wird, wird ein Alarm ausgelöst und eine E-Mail wird dem Admin geschickt.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>Kastensystem</li> </ul>
Berechtigung	Das System löst diese Funktion aus, wenn U11 oder/und U12 erfolglos sind.
WF-Referenz	Schlüssel Abholung / Schlüssel Abgabe
Vorbedingung	<ul style="list-style-type: none"> <li>Das Kastensystem muss mit dem Internet verbunden sein.</li> <li>Das Kastensystem muss über Strom verfügen.</li> <li>U11 und U12 müssen durchlaufen, bevor diese Funktion ausgelöst wird.</li> <li>U11 oder/und U12 müssen erfolglos durchlaufen.</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>Fehlermeldung Signal aus U11 / U12 - Digital Input (HIGH/LOW).</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>Ein Fehlersignal löst diese Funktion aus</li> <li>Es wird danach gecheckt, welcher Fehler die Funktion ausgelöst hat.</li> <li>Einige Fehler lösen einen Alarm aus (z.B., wenn die Tür geöffnet wird oder wenn jemand eine falsche Karte benutzt...).</li> <li>Bei jedem Fehler wird dem Admin eine E-Mail gesendet.</li> <li>Nach dieser Funktion wird das Kastensystem auf Wartezustand eingeschaltet, damit anderen Users das System weiter nutzen können.</li> </ol>
Ergebnis	E-Mail und/oder Alarm danach wird das System auf Wartezustand geführt.
Plausibilitäten	<ul style="list-style-type: none"> <li>Alarm wird nach dem Fehler für eine bestimmte Zeit gelöst.</li> <li>E-Mail wird gesendet.</li> <li>Das System setzt sich zurück.</li> </ul>
Fehler Handling	<ul style="list-style-type: none"> <li>Wenn kein Alarm sich auslöst, wird dem Admin benachrichtigt.</li> </ul>

	<ul style="list-style-type: none"> <li>Es wird im Log gespeichert und im Buchungssystem gezeigt, dass eine E-Mail gesendet wurde, damit der Admin später vergleichen kann, ob die E-Mail wirklich gesendet wurde.</li> <li>Sollte das System nicht mehr nutzbar sein, wird der Admin benachrichtigt.</li> </ul>
Folgeprozess	Wartezustand, bis ein User das System nochmal auslöst.
Anforderung	<b>FA-1</b>
Release	Release im Sprint 1
Test-Cases	<ul style="list-style-type: none"> <li>T-U15.1: Test, ob die Alarm-Funktion gut funktioniert.</li> <li>T-U15.2: Test, ob die E-Mail gesendet wird.</li> <li>T-U15.3: Test, ob das System sich auf Wartezustand einschaltet.</li> <li>T-U15.4: Test, ob es wiederholt wird, wenn das Schloss zu bleibt.</li> <li>T-U15.5: Test, ob diese Funktion nach U11/U12 kommt.</li> <li>T-U15.6: Test, ob das Fehlersignal wahr ist.</li> <li>T-U15.7: Test, ob die richtige Fehlermeldung zurückgegeben ist.</li> </ul>

Tabelle 18: U15: Alarm auslösen/ Warnung-Mail senden

### 5.17 U16: Insert Userdaten

Zweck/Ziel	Diese Funktion wird ausgelöst, wenn U12 erfolgreich ausgeführt wurde. Ziel ist es, wenn der Kasten erfolgreich geöffnet wird und U12 erfolgreich durchgeführt wird, werden die Daten des Benutzers, den der Kasten geöffnet hat, in der DB gespeichert. Es wird auch gespeichert, wann der Kasten geöffnet wurde, welcher Schlüssel abgeholt/zurückgegeben wurde und ob die Tür danach geschlossen wurde.
Akteur/Auslöser	<ul style="list-style-type: none"> <li>Kastensystem</li> </ul>
Berechtigung	Eine erfolgreiche U12-Funktion löst diese Funktion aus.
WF-Referenz	Schlüssel Abholung / Schlüssel Abgabe
Vorbedingung	<ul style="list-style-type: none"> <li>Die Kastentür muss geschlossen sein.</li> <li>Das Kastensystem muss mit dem Internet verbunden sein.</li> <li>Das Kastensystem muss über Strom verfügen.</li> <li>U12 soll erfolgreich ausgeführt sein.</li> <li>Eine Buchung eines Schlüssels muss vorhanden sein</li> </ul>
Daten-Input	<ul style="list-style-type: none"> <li>Erfolg Signal aus U12 - Digital Input (HIGH/LOW).</li> <li>Ein Signal aus dem Sensor, der uns sagt, welcher Schlüssel abgeholt / zurückgegeben wurde. (HIGH/LOW).</li> <li>Ein Signal aus dem Sensor, der uns sagt, welcher Schlüssel abgeholt / zurückgegeben wurde. (HIGH/LOW).</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>Ein Signal löst diese Funktion aus, wenn U12 erfolgreich ist.</li> <li>Userdaten, die gelesen wurden, wenn der Benutzer seine RFID-Karte / seinen Barcode gescannt hat, werden in der DB gespeichert.</li> <li>Der Zeitstempel wird in der Datenbank gespeichert.</li> <li>Es wird durch die Sensoren im Kasten erkannt, welcher Schlüssel abgeholt / zurückgegeben wurde und danach in der DB gespeichert.</li> <li>Es wird durch einen Sensor an der Tür erkannt, ob die Tür auf/zu ist und danach wird das als Telemetrie an das Buchungssystem gesendet und in der DB gespeichert.</li> <li>Nach dieser Funktion wird das Kastensystem auf Wartezustand eingeschaltet, damit anderen Users das System weiter nutzen können.</li> </ol>
Ergebnis	Speicherung aller Daten, die in Verarbeitungsschritte gelistet wurden, und danach wird das System auf Wartezustand geführt.
Plausibilitäten	<ul style="list-style-type: none"> <li>Die Sensordaten müssen richtig sein und in der DB gespeichert werden.</li> <li>Diese Funktion muss nur nach der U12 Funktion folgen.</li> <li>Der Türzustand muss als Telemetrie an das Buchungssystem in einem festen Intervall geschickt werden.</li> <li>Das System setzt sich zurück.</li> </ul>
Fehler Handling	<ul style="list-style-type: none"> <li>Die Datenbank meldet sich zurück, wenn die Daten gespeichert / nicht gespeichert sind. Falls die Daten nicht gespeichert sind, muss dies wiederholt werden.</li> </ul>



	<ul style="list-style-type: none"> <li>• Wenn die Daten nach allen Versuchen nicht gespeichert sind, wird das System ausgeschaltet, bis der Fehler beseitigt wird.</li> <li>• Wenn das System sich auf Wartezustand nicht zurücksetzt, wird das System ausgeschaltet und der Admin wird darüber informiert.</li> </ul>
Folgeprozess	Wartezustand, bis ein User das System nochmal auslöst.
Anforderung	<b>FA-1</b>
Release	Release im Sprint 1
Test-Cases	<ul style="list-style-type: none"> <li>• T-U16.1: Test, ob die Funktion U12 erfolgreich durchgelaufen ist.</li> <li>• T-U16.2: Test, ob die Daten aus dem RFID/Barcode-Leser richtig sind.</li> <li>• T-U16.3: Test, ob der Zeitstempel richtig gespeichert ist.</li> <li>• T-U16.4: Test, ob es richtig erkannt wurde, welcher Schlüssel abgeholt/zurückgegeben wurde.</li> <li>• T-U16.5: Test, ob der Türsensor richtig liest, ob die Tür auf/zu ist.</li> <li>• T-U16.6: Test, ob die Telemetriedaten richtig geschickt werden.</li> <li>• T-U16.7: Test, ob alle Daten in der DB gespeichert werden.</li> <li>• T-U16.8: Test, ob das System sich auf Wartezustand einschaltet.</li> </ul>

Tabelle 19: U16: Insert Userdaten

### Figma Prototype: [Smart Schloss Buchungssystem Prototype](#)

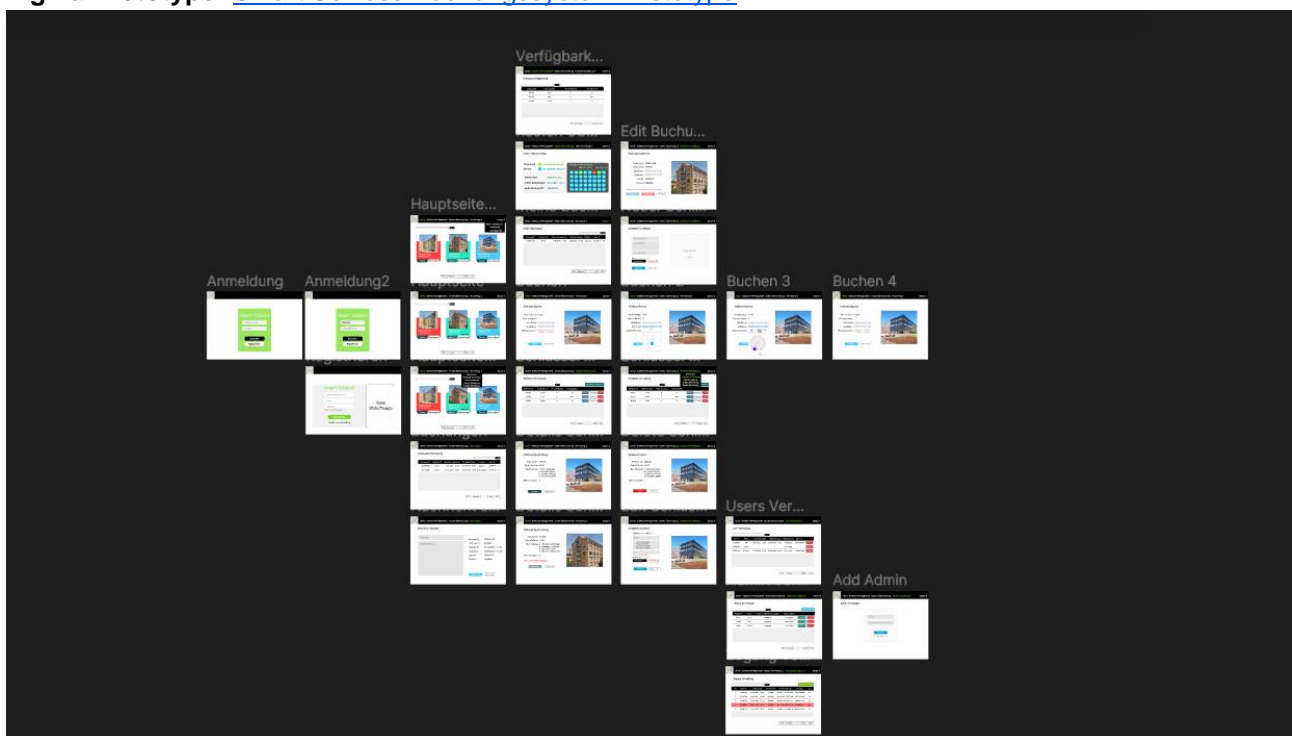


Abbildung 5: Figma Prototype (GUI-Design)

*Manchmal wurde nur "User" geschrieben, aber das heißt, dass auch Admin die Funktion macht oder/und die Bedingungen auch für ihn gelten, weil der Admin gleichzeitig ein User ist. Das ist auch an dem UC-Diagramm deutlich dargestellt.*

### Kasten/ Fach Prototyp:

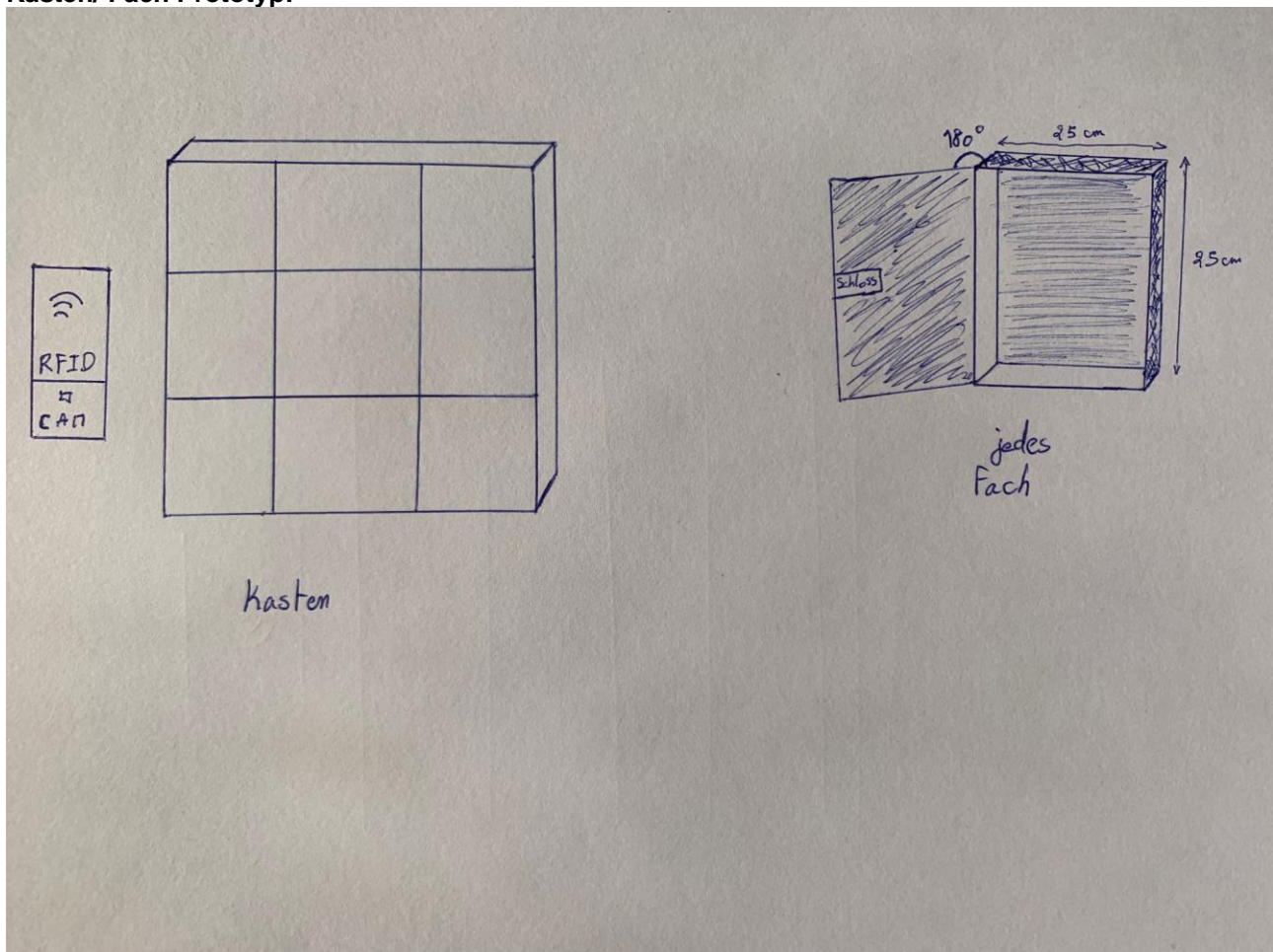
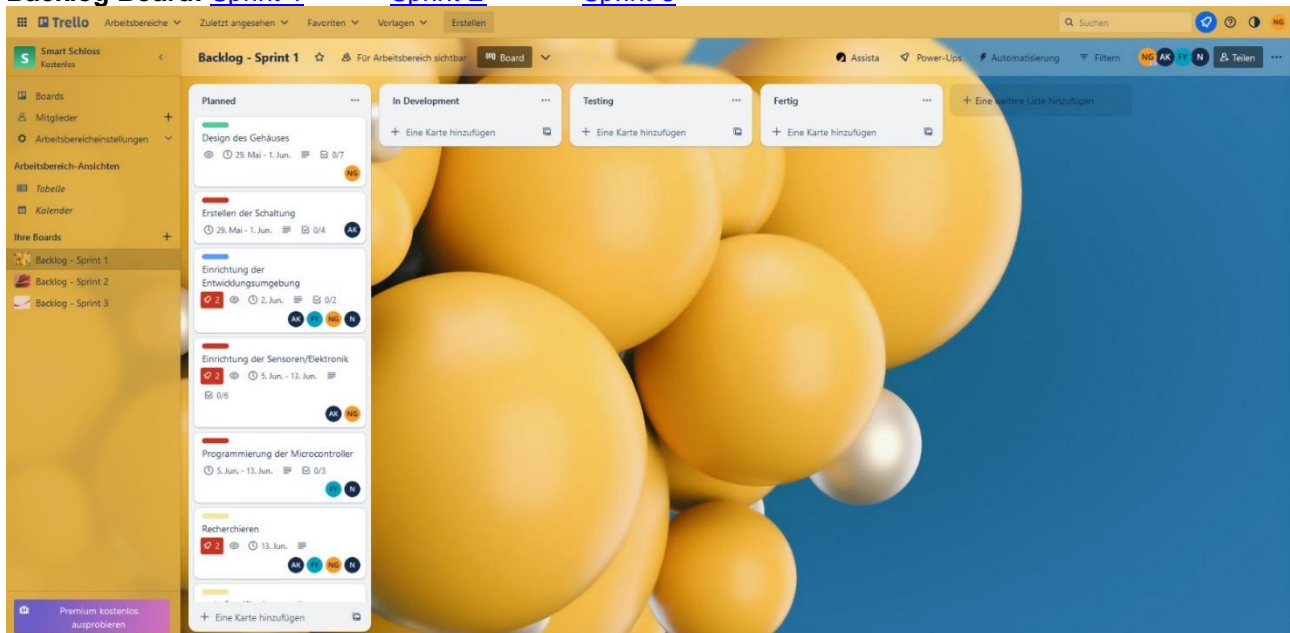


Abbildung 6: Kasten/ Fach Design

### Backlog Board: [Sprint-1](#) [Sprint-2](#) [Sprint-3](#)





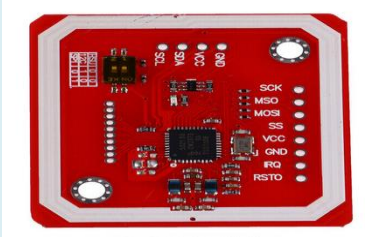



## 6 Offene Fragen

	Issue	Status	Owner	Deadline
1	Kommunikation zwischen User und Kasten über LCD oder LED?	abgeschlossen	Holger & Team	25.05.2023
2	Ist Blockieren von User auf das Buchungssystem erforderlich?	abgeschlossen	Holger & Team	25.05.2023
3	Wie viele Schlüssel müssen im Kasten sein?	abgeschlossen	Holger & Team	25.05.2023

*Tabelle 20: Offene Fragen*

## 7 Modulabhängigkeiten

Kategorie	Komponente	Bild
<b>Hardware:</b>		
<b>Mikrocontroller</b>	ESP32-CAM Modul	 <p><a href="#">ESP32-Cam</a></p>
<b>Schloss</b>	Elektromagnetisches Türschloss (z.B. 12V Elektromagnet)	 <p><a href="#">Link</a></p>
<b>RFID-/NFC-Lesegerät</b>	PN532 NFC-Modul	 <p><a href="#">Link</a></p>
<b>Fehler Ausgabe</b>	LED	


<b>Türzustand Erkennungssensor</b>	Magnetisch kontakt Sensor	 <a href="#">Link</a>
<b>Stromversorgung</b>	5V USB-Netzteil und 12V/1A DC-Netzteil	
<b>Software:</b>		
<b>Datenbank-Management- System</b>	MySQL	ERD: <a href="#">Wünsch vom externen Drupal Entwickler</a>
<b>Programmiersprachen</b>	C/C++ - Mikrocontroller- Programmierung PHP für das Drupal- Webportal  HTML, CSS und JavaScript für das Frontend-Design	
Kommunikationsprotokoll	Kommunikationsprotokoll (RESTful API - HTTP)  E-Mail-Protokoll (SMTP) User-Autorisierung Protokoll (LDAP)	

Tabelle 21: Modulabhängigkeiten

## 8 Wer hat was gemacht

Autor	Aufgabe/Kapitel	Anteil
Ahmed	Alle	25%
Firas	Alle	25%
Heltonn	Alle	25%
Nader	Alle	25%

Tabelle 22: Wer hat was gemacht