

**Semesterendprüfung**

Name, Vorname	Klasse	Punkte	Note

Zeit	90 Minuten
Total Punktzahl	90
Hilfsmittel	Die Prüfungsmodalitäten sind im jeweiligen Infoblatt SWEN1 aufgeführt.
Abgabe	Füllen Sie dieses Deckblatt aus. Schreiben Sie Ihre Lösungen direkt in das Aufgabebblatt, wo Platz dafür vorgesehen ist. Weitere Lösungsblätter sind unbedingt mit Ihrem Namen, Vornamen, ZHAW Kürzel und der Aufgabennummer ausgefüllt abzugeben.
Hinweise	<p>Lesen Sie alle Aufgabenstellungen sorgfältig durch, bevor Sie mit der Bearbeitung beginnen.</p> <p>Wenn eine Aufgabenstellung unklar ist, treffen Sie vernünftige Annahmen und <b>dokumentieren</b> Sie diese. Die Prüfungsaufsicht wird keine Fragen beantworten.</p> <p>Schreiben Sie weder mit Bleistift noch mit roter Farbe.</p> <p>Wir wünschen Ihnen viel Erfolg für die Prüfung!</p>

# 1 Anforderungsanalyse: Hotelsystem (42 P)

## 1.1 Einleitung und Problembeschreibung

Eine kleine Hotelkette hat festgestellt, dass der grösste Teil seiner Kunden junge Leute sind, für die (neben einer gewissen Grundqualität des Angebots) nur der Preis relevant ist. Deshalb möchte sie verschiedene ihrer Hotels so umbauen, dass der **Kunde** den Check-In ohne Personal an einem Terminal in der Eingangshalle durchführen kann. Sie wurden als Projektleiter für die Entwicklung einer neuen Software („**Hotelsystem**“) bestimmt, die das ermöglichen soll und haben bereits mit den Hotelverantwortlichen gesprochen und erste Abklärungen gemacht. Dazu haben Sie die folgenden Notizen angelegt.

Die Hotelkette arbeitet ausschliesslich mit **externen Buchungs-Plattformen** (eBP) wie Bookings oder Expedia zusammen. Über das Terminal in der Hotelhalle, wo die Kunden sich einchecken können, können keine Buchungen getätigt werden.

Das „Hotelsystem“ dient nun dazu, einen Kunden zu identifizieren und ihm gemäss seiner **Buchung** konkrete **Zimmer** zuzuweisen. Die Identifikation geschieht über die Buchungsnummer, die der Kunde beim Buchen über die eBP erhalten hat.

Die Hotelkette hat schon früh ein **automatisches Zimmerverschlusssystem** (ZVS) in all seinen Hotels installiert. Technisch sind es **Karten**, die mit einer RFID ausgestattet sind. An jeder Türe ist ein Leser montiert, der die ID (128 Bit Wert) der Karte kontaktlos lesen kann und diese Nummer bei einer Öffnungsanfrage dem zentralen Server des Zimmerverschlusssystems übermittelt. Daraufhin prüft dieser die Berechtigungen und gibt der Türe den Befehl zu öffnen oder eine Fehlermeldung („Pieps“) zu geben. Das ZVS arbeitet autonom und muss vom „Hotelsystem“ gleich nach der Zimmerzuteilung lediglich wissen, welche Karte wann zu welchem Zimmer **Zutrittsberechtigung** hat. Das Hotel hat einen Hauslieferanten für das ZVS und zeigt keine Absicht, dies zu ändern.

Für die eBP soll eine Web-Service Schnittstelle vorgesehen werden, wo die fertigen Buchungen dem „Hotelsystem“ übermittelt werden. Die Angebote, die die eBP von der Hotelkette haben, werden von einem Hotelangestellten eingegeben, der dafür sorgt, dass es in den eBP nicht mehr Angebote hat wie das Hotel aufnehmen kann. Diese Angebote beziehen sich aber immer auf eine bestimmte **Kategorie von Zimmern**. Deshalb muss während dem Check-In des Kunden noch eine Zimmerzuteilung durchgeführt werden. Der Check-In und Check-Out Zeitpunkt ist übrigens fix und nimmt auch keine Rücksicht auf Kundenwünsche oder auf den Reinigungszustand der Zimmer. Sollte trotzdem kein Zimmer der gewünschten Kategorie mehr frei sein, muss die **Zentrale** benachrichtigt werden, die dann mit dem Kunden direkt per Telefon in Kontakt tritt und die Zimmerzuteilung vornimmt.

Das Check-In Terminal ist das Benutzerinterface des „Hotelsystems“ für den Kunden und soll ein grosser Bildschirm mit Touch-Screen, ohne eigene Tastatur oder Maus, sein. Darauf sollte die Benutzerführung natürlich Rücksicht nehmen.

Gleich daneben liegt ein Stapel von **Zimmerkarten**, die im Moment nicht in Gebrauch sind. Der Kunde kann so während dem Check-In die notwendige Anzahl Karten nehmen und Sie dann zum richtigen Zeitpunkt an den Leser halten, der beim Check-In Terminal vorhanden ist.

Der Gast soll auf alle Fälle die Korrektheit seiner Buchungsdaten (Datum, Dauer, Personendaten) beim Check-In bestätigen.

## **1.2 Domänenmodell (DM) (16 P)**

Analysieren Sie die Informationen aus der Problembeschreibung und leiten Sie davon ein DM ab. Belegen Sie (mittels kurzer Notizen und/oder Referenzen), welche Elemente des DM sie aus welchen Elementen der Problembeschreibung ableiten und wo Sie allgemeines Wissen über die relevanten Geschäftsprozesse einfließen lassen.

## **1.3 Anwendungsfall „Check-In“ (12 P)**

Schreiben Sie einen ausführlichen Anwendungsfall (**fully dressed use case**) für den Check-In. Legen Sie auch grossen Wert darauf, die wichtigsten Ausnahmen und Erweiterungen zu dokumentieren. Wo es keine konkreten Hinweise in der Problembeschreibung gibt, machen Sie bitte vernünftige Annahmen.

## **1.4 Weitere Anwendungsfälle (4 P)**

Schreiben Sie noch weitere 2 kurze (brief) Anwendungsfälle und dokumentieren Sie, wo in der Problembeschreibung die Notwendigkeit dieser Anwendungsfälle gegeben ist.

## **1.5 Anwendungsfalldiagramm (3 P)**

Zeichnen Sie mit den Anwendungsfällen aus Aufgabe 1.3 und 1.4 ein Anwendungsfalldiagramm.

## **1.6 Benutzerfreundlichkeit (3 P)**

Leiten Sie aus der Problembeschreibung alle Aspekte ab, die die nicht funktionale Anforderung „Hohe Benutzerfreundlichkeit“ betrifft. Beschreiben Sie diese dann noch etwas genauer.

## **1.7 Architekturanalyse (4P)**

Analysieren Sie die Anforderungen zur Ansteuerung des automatischen Zimmerverschlusssystems (ZVS) und diskutieren Sie diese unter dem Aspekt von Variations- und Entwicklungspunkten.

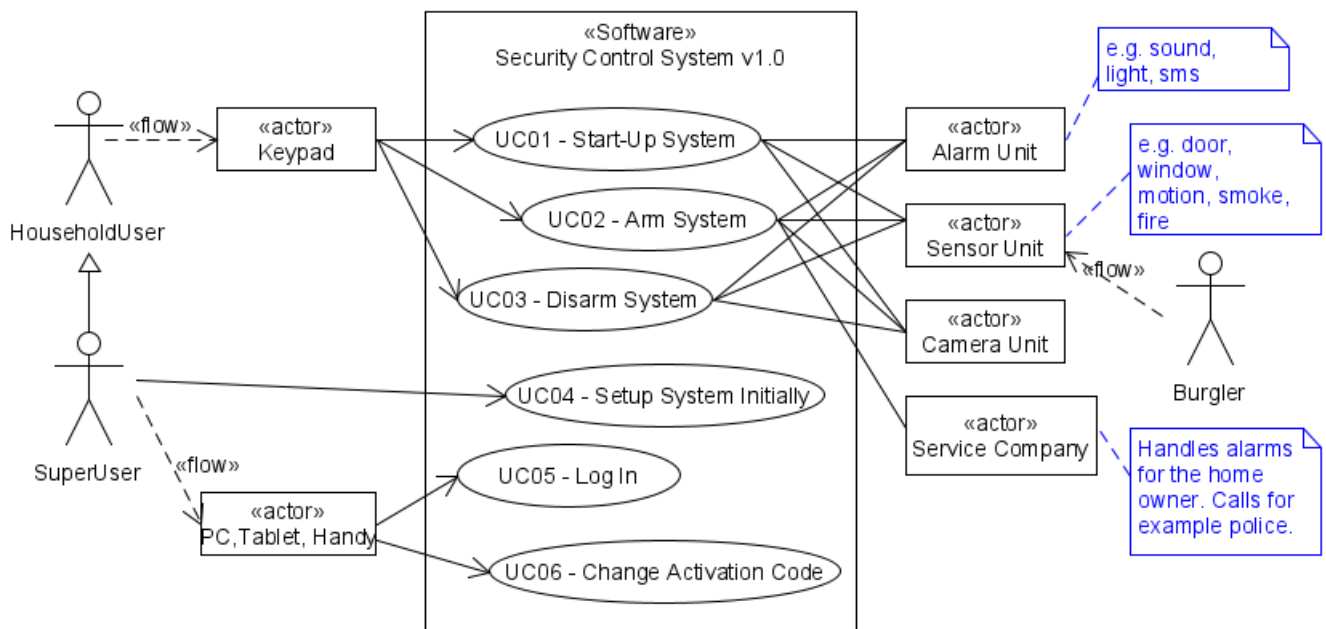
## 2 Design: SafeHome (48 P)

### 2.1 Einleitung und Problemstellung

Die fiktive Firma SafeHome erstellt Sicherheitssysteme im Kontext von Heimautomation. Als Heimautomation bezeichnet man die Gesamtheit von Überwachungs-, Steuer-, Regel- und Optimierungseinrichtungen in Privathäusern und -wohnungen.

Bei einem Sicherheitssystem geht es darum, Einbruch, Wasser, Feuer und Luftqualität zu überwachen und Alarme auszulösen, wenn ein Einbrecher detektiert oder bestimmte Schwellwerte überschritten werden.

Für eine bestehende Produktlinie soll eine neue (objektorientierte) Software-Plattform entwickelt werden. Dabei soll die bestehende Hardware (Sensoren, Aktoren etc.) weiterverwendet werden. Zur Orientierung und als Ausgangspunkt für das erste Release der neuen Plattform dient ein Use-Case-Modell.

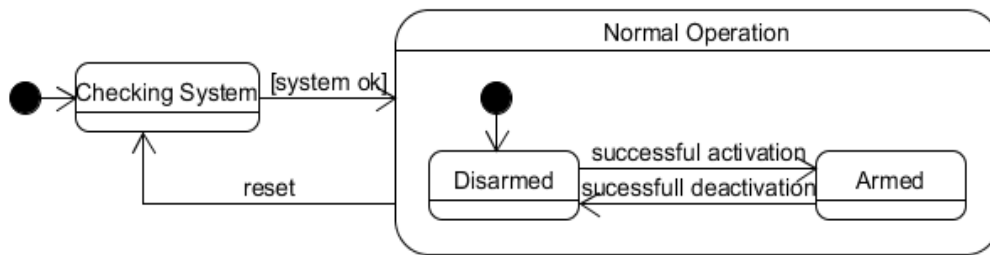


UC02 und UC03 sind die Haupt-Anwendungsfälle. Um das Sicherheitssystem zu aktivieren (engl. to arm) oder zu deaktivieren (engl. to disarm), muss der Hausbewohner einen Aktivierungscode eingeben.

Ein Super-User kann zusätzlich zu Hause oder ausser Haus (extern) sich in das Sicherheitssystem einloggen (Handy App, PC etc.) und neben der Aktivierung oder Deaktivierung des Sicherheitssystems auch den Aktivierungscode ändern.

Die Software im Hauptgerät im Haus besteht aus zwei Hauptkomponenten: der Steuerungssoftware für die Überwachung und einer Webapplikation mit Webserver für den lokalen und entfernten Zugriff auf das Sicherheitssystem. Für das erste Release wird aber nur die Steuerungssoftware neu entwickelt.

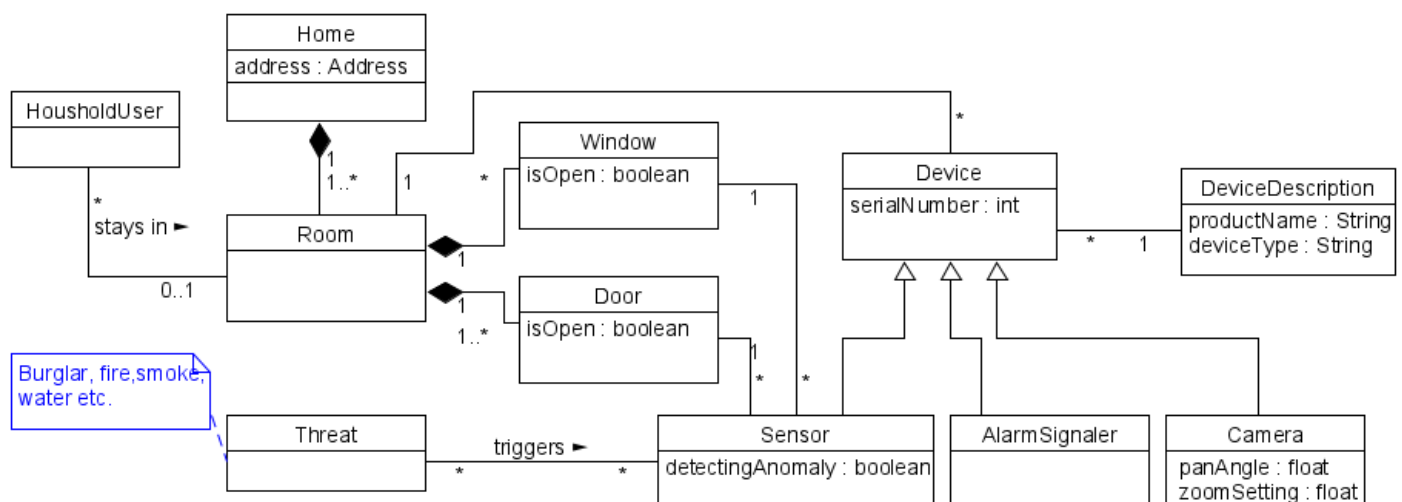
Das prinzipielle Verhalten der Steuerungssoftware ist im folgenden UML-Zustandsdiagramm dargestellt.



Nach dem Einschalten (UC01) und einer erfolgreichen Prüfung des Systems (Checking System) wechselt das System aufgrund von Benutzer-Aktionen (UC02, UC03) zwischen Disarmed (nicht aktiviert) und Armed (aktiviert) hin und her. Im Zustand Armed ("scharf" gestellt) muss das System auf getriggerte Sensoren reagieren und einen Alarm auslösen.

## 2.2 Design-Klassendiagramm (16 P)

Als Ausgangspunkt für das Design der Klassen für die Systemüberwachung dient das folgende Domänenmodell.



Folgende Anforderungen des Produktmanagers der Produktlinie müssen dabei berücksichtigt werden.

Das Layout des Hauses bzw. der Wohnung – Räume, Türen und Fenster – müssen nicht abgebildet werden. Um ein Gerät (Sensor, Aktor) bei einem Alarm zu lokalisieren, reicht ein Attribut mit einer Beschreibung im Gerät (z.B. "Fenster im 1. Stock").

Das Sicherheitssystem kann durch den Hausbewohner auf zwei Arten aktiviert werden. Der Hausbewohner geht weg (armed-away) oder der Hausbewohner ist nach der Aktivierung weiter im Haus (armed-stay). Bei der zweiten Aktivierung werden dann z.B. die Bewegungssensoren nicht aktiviert, so dass der Hausbewohner sich im Haus aufhalten kann, ohne einen Alarm auszulösen. Jedes

Gerät muss darum zwei Eigenschaften haben, ob es bei Abwesenheit und/oder Anwesenheit getriggert werden können soll.

Ein Alarm kann z.B. eine Kombination von Sound, Licht und Benachrichtigung einer Service-Firma sein.

Entwerfen Sie ein UML-Design-Klassendiagramm (DCD) für die Realisierung der Systemüberwachung.

Erläutern Sie zusätzlich im Design, welche Entwurfsmuster (GRASP, GoF) Sie eingesetzt haben und warum (Begründung).

## **2.3 Verhalten während der Systemüberwachung (8 P)**

Wenn ein Bewegungssensor eine Bewegung detektiert (z.B. einen Einbrecher), wird nicht sofort ein Alarm erzeugt. Die Bewegung kann ja durch den Hausbewohner beim Heimkehren und der Deaktivierung des Sicherheitssystems erzeugt worden sein.

Erst nach 45 Sekunden geht das System in einen Zustand höherer Sensitivität. In diesem Zustand wird bei einer weiteren Bewegung sofort Alarm ausgelöst. Das System bleibt in diesem Zustand für 5 Minuten bis es wieder in den Normalzustand (kein Sensor getriggert) wechselt.

Wenn irgend ein anderer Sensor ausgelöst wird (oder der Bewegungssensor im Zustand höher Sensitivität ist), wird ein Alarm ausgelöst. Das System startet in diesem Zustand einen Timer und wenn in einer benutzerdefinierbaren Zeit kein weiterer Sensor ausgelöst wird, erlischt der Alarm.

Dies verhindert ein wiederholtes Auslösen bei einem Falsch-Alarm. Wenn aber in diesem Zustand weitere Sensoren ausgelöst werden, wird der Timer wieder zurückgesetzt, so dass der Alarm weiter läuft.

Erstellen Sie ein UML-Zustandsdiagramm, das das Verhalten im Zustand der Überwachung (Armed) modelliert.

## **2.4 Szenario Systemüberwachung aktivieren (4 P)**

Modellieren Sie die Aktivierung der Systemüberwachung für den Fall, dass sich kein Hausbewohner mehr im Haus befindet (armed-away) mit einem UML-Sequenzdiagramm. Falls sich neue Erkenntnisse im Klassen-Design ergeben, führen Sie die im DCD nach ;-)

## **2.5 Szenario Einbrecher detektiert (6 P)**

Modellieren Sie das Szenario, dass das Sicherheitssystem aktiviert ist (armed-away) und ein Bewegungssensor durch einen Einbrecher getriggert wird mit einem UML-Sequenzdiagramm. Falls sich neue Erkenntnisse im Klassen-Design ergeben, führen Sie die im DCD nach ;-)

## **2.6 Zusätzliche Anforderungen (10 P)**

Der Produktmanager möchte, dass das System schon im ersten Release so entworfen wird, dass erstens mehrere Konfigurationen - Geräte und deren Verwendung - erstellt werden können und zweitens die verschiedenen Geräte (Sensoren, Aktoren) auch zu Zonen und Sub-Zonen in einer Konfiguration gruppiert werden können. Als Beispiel könnte man eine Zone „1. Stock“ konfigurieren mit den Sub-Zonen Küche, Wohnraum und WC und mit den entsprechenden Geräten.

Dies erlaubt in grösseren Gebäuden den Alarm schneller zu identifizieren und auch einzelne Zonen selektiv aktivieren und deaktivieren zu können.

Entwerfen Sie eine flexiblere Variante mit Konfigurationen und Sicherheitszonen für grössere zu überwachende Gebäude. Dokumentieren Sie die Erweiterungen im DCD (aus Aufgabe 2.2) und evtl. Annahmen. Erläutern Sie zusätzlich, welche Entwurfsmuster (GRASP, GoF) Sie eingesetzt haben und warum (Begründung).

## **2.7 Fehlerszenarien (4 P)**

Identifizieren Sie mögliche Fehlerszenarien, die sich aus fehlerhafter Hardware ergeben und beschreiben Sie in ein paar Sätzen, wie Sie diese lösen.