



Übung 8: Use Cases

Sie haben den Auftrag erhalten, die Software für einen neuen Bankautomaten zu entwickeln. Wie nicht anders zu erwarten, soll es möglich sein, Geld abzuheben – und zwar mit einer EC Karte aber auch mit einer Kreditkarte. In beiden Fällen muss eine PIN angegeben werden, damit die Karte des Bankkunden geprüft wird.

Mit dem Bankautomaten sollen aber noch mehr Dienstleistungen möglich sein. Da der Automat eh mit dem Buchungssystem der Bank verbunden ist, soll es insbesondere dem Bankkunden möglich sein, den aktuellen Kontostand abzurufen oder einen Kontoauszug ausdrucken zu lassen. Auch seine/ihre Mobiltelefonkarte soll mit dem Bankautomaten aufgeladen werden können.

Je nachdem ob mit EC- oder Kreditkarte Geld abgehoben wird, wird der Betrag im Buchungssystem der Bank beziehungsweise im jeweiligen Kreditkartensystem verbucht. Wenn der Bankautomat nach der Auszahlung feststellt, dass keine ausreichende Menge an Geldscheinen mehr vorhanden ist, wird der Bankautomat deaktiviert. Dies wird in einem Systemprotokoll festgehalten. Ein Bankangestellter muss also erst wieder Vorort Geldscheine auffüllen und den Bankautomaten initialisieren. Im Übrigen soll auch ein Bankangestellter jederzeit den Bankautomaten deaktivieren können.

Gerade solche Betriebsstörungen interessieren natürlich das Wartungsteam der Bank. Ein Administrator kann daher jederzeit das Systemprotokoll abrufen und selbstverständlich auch den Bankautomaten wieder zum Laufen bringen (initialisieren).

Aufgabe 1 – Anwendungsfälle aus Text

Identifizieren Sie alle Anwendungsfälle, welche Sie aus der obigen Beschreibung des Bankautomaten entnehmen können.

- a) Geben Sie dazu den Anwendungsfällen sinnvolle, sprechende Bezeichnungen!

***Hinweis:** Verwenden Sie Verben in den Bezeichnungen der Anwendungsfälle. Anhand der Bezeichnung sollte schon möglichst klar sein, was der Benutzer erreichen möchte (zum Beispiel „Geld abheben“).*

- b) Überlegen Sie sich die Systemgrenze und welche Anwendungsfälle nicht relevant für Ihr Bankautomat-System sind!

Aufgabe 2 – Akteure identifizieren

Akteure sind die Benutzer des Softwaresystems oder externe Systeme, welche mit dem Softwaresystem interagieren.

Überlegen Sie, wer Ihr Bankautomat-System gemäß der obigen Beschreibung verwendet beziehungsweise mit ihm interagiert. Dokumentieren Sie die Akteure in einer Tabelle mit den Beziehungen zu den in Aufgabe 1 gefundenen Anwendungsfällen!

Aufgabe 3 – Anwendungsfallübersicht erstellen (Use Case Diagram)¹

Erstellen Sie schließlich ein Anwendungsfalldiagramm für das oben beschriebene Bankautomat-System! Für die Erstellung des Diagramms können Sie wieder MS-Visio oder Papier und Bleistift verwenden. Sie dürfen aber auch gerne jedes andere UML-Modellierungswerkzeug ausprobieren.

Zur Erinnerung: Ein Anwendungsfalldiagramm für ein System besteht aus der Systemgrenze, Akteuren, Anwendungsfällen, Beziehungen zwischen Akteuren und Beziehungen zwischen Anwendungsfällen (Einschluss «*include*»², Erweiterung «*extend*», Generalisierung).

Aufgabe 4 – Dokumentieren eines Anwendungsfalls (Use Case)

Modellieren Sie den Anwendungsfall „Geld mit EC Karte abheben“. Der Anwendungsfall startet damit, dass ein Nutzer seine Bankkarte in den Automaten steckt. Zum Geldabheben benötigt er eine elektronisch lesbare, gültige Bankkarte und ein nicht überzogenes Konto, weiterhin muss er seine PIN wissen. Der Anwendungsfall verläuft dann so, wie Sie es von Ihrer Bank her kennen.

Beschreiben Sie den Use Case mittels der in der Vorlesung vorgestellten Schablone!

Hinweis: Verwenden Sie bei der Beschreibung des Ablaufs Aktiv-Formulierungen. Beschreiben Sie insbesondere den kompletten Ablauf (also auch Erweiterungen, Alternativen), und nicht nur den regulären Ablauf.

¹ Sie können Ihr Ergebnis mit der Checkliste von Scott Ambler prüfen, siehe:
<http://www.agilemodeling.com/style/useCaseDiagram.htm>.

² Manche Werkzeuge verwenden dafür auch den «*uses*» Stereotyp.