

Laborprojekt Carsharing System

Softwaretechnik 2

Sommersemester 2022

Prof. Dr.-Ing. Jasminka Matevska

(jasminka.matevska@hs-bremen.de)



- Die Rechte an geschützten Marken liegen bei den jeweiligen Markeninhabern
- Die Rechte an referenzierte Literatur, Folien, Notizen und sonstigen Materialien liegen bei den jeweiligen Autoren
- Alle Rechte an den Materialien zu dieser Veranstaltung liegen bei ihrem Autor, Prof. Dr.-Ing. Jasminka Matevska
- Jede Form der teilweisen oder vollständigen Weitergabe, Speicherung auf Servern oder Nutzung in Lehrveranstaltungen, die nicht von dem Autor selbst durchgeführt werden, erfordert seine schriftliche Zustimmung. Eine schriftliche Zustimmung ist darüber hinaus für jede kommerzielle Nutzung erforderlich
- Für inhaltliche Fehler kann keine Haftung übernommen werden

- Ausgangssituation
- Auftragsbeschreibung
- Carsharing
- Kundenanforderungen
- Rahmenbedingungen
- Aufgaben

- Sie sind ein Unternehmen, das seit mehr als 10 Jahren Software-Kommunikationssysteme entwickelt, vertreibt und administriert
- Sie sind ein mittelständisches Unternehmen in Deutschland
- Kundenkreis (bisher): Internet Provider
- Stagnierende Umsätze veranlassen Sie zur Ausweitung Ihres Geschäftsfeldes

- Ihr Kunde ist eine mittelständische Autovermietung
- Das Unternehmen möchte in das Carsharing-Geschäft einsteigen
- Es soll ein Softwaresystem entwickelt werden, das eine automatische Abwicklung der Reservierungen, Verwaltung der Fahrzeuge, Mitglieder und Abrechnungen ermöglicht

- Die Fahrzeuge einer Carsharing-Organisation sind entweder auf fest angemieteten Parkplätzen über eine Stadt oder einen größeren Ort verteilt oder parken im öffentlichen Straßenraum.
 - Die festen Mietstationen befinden sich häufig an Verkehrsknotenpunkten des öffentlichen Verkehrs wie Flughäfen, Bahnhöfen, Tramknoten, Endstationen von Buslinien, wo sie von den Nutzern gut erreichbar sind.
 - Die vorab reservierten Fahrzeuge werden meist benutzt, um von diesen Knotenpunkten aus abgelegene Ziele zu erreichen.
 - Das Fahrzeug muss nach Ablauf der gebuchten Zeit wieder an seinen ursprünglichen Standort zurückgebracht werden.
 - Der Nutzer muss vorher entscheiden, an welchem Standort des Carsharing-Anbieters das Fahrzeug abgeliefert werden soll.
 - Das Fahrzeug muss in einem definierten Bereich abgestellt werden.
- [de.wikipedia.org]

Kundenanforderungen (Auszug aus dem Lastenheft)

1. Die fertige Anwendung muss über eine URL erreichbar sein
2. Es soll folgende Benutzer geben: Gast, Mitglied, Mitarbeiter und Admin
3. Nur Mitglieder sollen ein Fahrzeug ausleihen können
4. Die Beantragung bzw. Verwaltung einer Mitgliedschaft und die Verwaltung der Reservierungen soll über folgende Wege möglich sein:
 1. Vor Ort in einer Filiale durch einen Mitarbeiter
 2. Telefonisch über ein CallCenter
 3. Durch das Mitglied selbst über eine Web-Schnittstelle und/oder eine App
5. Die Verwaltung der Fahrzeuge im Pool, der Belegung der Ausleihstationen und der Abrechnungen soll durch die Mitarbeiter durchgeführt werden, wobei diese auch über eine Web-Schnittstelle auf das System zugreifen können sollen
6. Die Daten sollen in einer Datenbank persistent gespeichert werden
7. Jedes Mitglied soll eine programmierbare Karte bekommen, die über ein RF-Signal (oder vergleichbar) das reservierte Fahrzeug bei Abholung öffnen und bei Rückgabe schließen kann

Kundenanforderungen (Auszug aus dem Lastenheft)

8. Jedes Fahrzeug besitzt ein GPS-System und soll jederzeit geortet werden können
9. Alle Daten müssen verschlüsselt übermittelt werden
10. Die Software soll ausfallsicher sein
11. Eine Anbindung an das bestehende Buchhaltungssystem im Unternehmen wäre wünschenswert
12. Es wäre gut, wenn das Gesamtsystem (Server und Clients) für Adaption an verschiedene Orte und Länder ausgelegt ist
13. Automatisierte Workflows bei Verspätungen und vorzeitiger Rückgabe von Fahrzeugen wären wünschenswert
14. Erfassung der Schadens-, Wartungs- und Reinigungs-Managementdaten ist für die erste Lieferung optional
15. In der initialen Version soll es drei Tarife geben: Basic, Ermäßigt und Exklusiv
16. Abbildung komplexer Tarifmodelle und Unterstützung verschiedener Zahlungsverfahren in einer „Billing-Engine“ wären hilfreich

- Der Auftragnehmer ist für die Entwicklung, Beschaffung, Integration und Test der benötigten Hardware und Software zuständig
- Die benötigte Hardware und Software soll durch den Auftragnehmer vollständig installiert und in Betrieb genommen werden
- Die Wartung inclusive Erweiterung des Systems ist auch unter Verantwortung des Auftragnehmers für die Dauer von 5 Jahren nach der erfolgreichen Inbetriebnahme
- Die Wartung der Software erfolgt nach den vertraglich festgelegten Service Level Agreements (z.B. Verfügbarkeit von 95%)

- Mitgliedskonten anlegen
- Mitgliedsdaten ändern
- Login/Logout
- Neue Reservierungen hinzufügen
- Reservierungen einsehen
- Reservierungen löschen

- L01: Pflichtenheft (150 P)
 - L02: Projektplan (100 P)
 - L03: Final Presentation (100 P)
 - L04: Technische Dokumentation (250 P)
 - L05: Implementierung, Test und Abnahme (300 P)
-
- **Hinweis: Einzelleistungen sind in den Abgaben deutlich zu kennzeichnen!**

L01: Pflichtenheft (150 Punkte)

- Aufgabe: Erstellen Sie ein Pflichtenheft für Ihr Projekt
 - Vorlage aus der Vorlesung
 - Anwendungsfälle mit UML Use Case Diagrammen
 - Use Case Beschreibung (Spezifikationsschablone aus der Vorlesung), wichtigste Use Cases, eine Schablone pro Teammitglied
 - Beschreiben Sie die verschiedenen Anforderungen und Randbedingungen
 - Klassifizieren Sie die Anforderungen nach dem Muster: muss, soll, kann
 - Notieren Sie eine Verifikationsart zu jeder Anforderung (Test (T), Inspektion (I), Analyse (A) oder Review of Design (RoD))
- Abgabe: AULIS, KW 20
- Abgabeform
 - Ein PDF-Dokument, max. 25 Seiten ohne Verzeichnisse
 - Dateiname: L01_Pflichtenheft_Projektname.pdf
- Vorstellung: KW 20, 15 Min.

L02: Projektplan (100 Punkte)

- Aufgabe: Erstellen Sie ein Projektplan für Ihr Projekt bestehend aus:
 - Projektname und Logo
 - Zieldefinition: übergeordnetes Ziel, Leistungs-, Termin- und Kostenziele (grob)
 - Stakeholder Analyse: Stakeholder, -Typ, und –Matrix
 - Risiko-Analyse: 5 Projektrisiken, Bewertungsmatrix, Schwierigkeitsgrad und Gegenmaßnahme
 - Disziplinatorientierter Projektstrukturplan (Work Breakdown Structure) für das gesamte Projekt
 - Meilensteine für das Gesamtprojekt
 - Projektzeitplan (GANTT Chart) nur für das aktuelle Semester
- Abgabe, AULIS, KW 23
- Abgabeform
 - Ein PDF-Dokument, max. 15 Seiten ohne Verzeichnisse
 - Dateiname: L02_Projektplan_Projektname.pdf
- Vorstellung: KW 23, 15 Min.

L03: Final Presentation (100 Punkte)

- Aufgabe: Stellen Sie Ihr Projekt vor
 - Erstellen Sie eine Präsentation zum gesamten Projekt
 - Führen Sie Ihre Implementierung vor
- Vorstellung: KW 28
 - Dauer der Präsentation: 15 Min.
 - Dauer der Vorführung: 5 Min.
- Abgabe: AULIS, KW 28
- Abgabeform
 - Ein PDF-Dokument
 - Dateiname: L04_FinalPresentation_Projektname.pdf
- **Wichtig: jedes Team-Mitglied soll vortragen!**

L04: Technische Dokumentation (250 Punkte)

- Aufgabe: Erstellen Sie eine technische Dokumentation für das System
 - Product Backlog für die Mindestanforderungen
 - Beschreibung der ausgewählten Technologien und Werkzeuge
 - Versionsmanagement: Beschreibung der git Repository Struktur
 - Architekturbeschreibung für das Gesamtsystem
 - Komponentendiagramm und Kurzbeschreibung)
 - Modellierung der statischen und dynamischen Sicht (Klassen, ER, wichtigste Funktionalitäten, Ablauf bzw. Interaktion/Sequenzdiagramme und Beschreibung)
 - Beschreibung der Komponenten/Module und Schnittstellen, die implementiert wurden, ggf. Screenshots (bitte Mindestanforderungen beachten!)
 - Verifikation
 - Beschreibung der durchgeführten Unit- und System Tests
 - Beschreibung der Testergebnisse
 - Verification Matrix (Welche Anforderungen aus dem Backlog wurden wie verifiziert?)
- Abgabe: AULIS, KW 29
- Abgabeform
 - Ein PDF-Dokument, max. 40 Seiten ohne Verzeichnisse
 - Dateiname: L04_Technische_Dokumentation_Projektname.pdf

L05: Implementierung, Test und Abnahme (300 Punkte)

- Aufgabe:
 - Erstellen Sie ein Git-Repository mit Ihrem kompletten Projekt (Implementierung und Tests) auf dem HSB Git-Server
 - Vergeben Sie alle notwendigen Rechte an Herrn Raven und Prof. Matevska (user: rnoah und matevska)
 - Erstellen Sie eine README-Datei mit einer Anleitung zum Auffinden, Bauen und Ausführen Ihres Projekts
 - Die fertige Anwendung muss lauffähig und über eine URL erreichbar sein
 - Führen Sie die Installation und Ihre Implementierung vor
- Abgabe: AULIS, KW 29
- Abgabeform
 - Ein PDF-Dokument, Dateiname: L05_README_Projektname.pdf
 - Git-Repository auf dem HSB-Server, KW 29 (Gesamtprojekt inkl. Dokumentation)
- Abnahme: KW 29, 30 Min. pro Team