

Transferaufgabe APDE Diplomsemester A (DA)

Hotel Grand Pilatus

E-Mail-Kampagnen Management

Semester:	3. Studienjahr, Diplomsemester A
Sozialform:	Bewertete Einzel- oder Gruppenarbeit ausserhalb der Unterrichtszeit
Modul/e:	APDE – Application Design
Gesamtpunkte:	100 Punkte zählt zu 100% als Fachnote
genügende Note ab:	≥ 60 Punkte
Prüfungsform:	Schriftliche Prüfung, bestehend aus mehreren Teilaufgaben
Bewertung:	Gemäss Bewertungsraster
Seitenzahl:	Siehe Details in Aufgabenstellungen
Abgabe der Arbeit:	Schriftliche Arbeit als PDF und Sourcen als ZIP-Datei termingerecht im OpenOlat hochladen
Speichername:	Jedes Gruppenmitglied speichert alle Prüfungsteile je Modul jeweils unter: «APDE_Nachname_Vorname_Trainingsfirma»

Autor/in: WISS
Version: 1.0
Hinweise: Keine

Inhaltsverzeichnis

1.	Beachte / Hinweis.....	3
2.	Ausgangslage.....	3
3.	Formale Beurteilungskriterien und korrektes Zitieren.....	3
4.	Aufgabenstellungen.....	4
4.1.	Softwarearchitektur, technologisches Design und Entwicklungsumgebung	4
4.2.	Server-App.....	6
4.3.	Client-Apps	7
4.4.	Deployment und Test.....	8

1. Beachte / Hinweis

Bitte beachte, dass alle Aufgaben immer im Hinblick auf deine gewählte Trainingsfirma zu lösen sind.

2. Ausgangslage

Der Hintergrund für diese Transferaufgabe bildet die «Hotel Grand Pilatus» gemäss der Trainingsfirmen-Beschreibung der WISS. Das Hotel plant, ein innovatives E-Mail-Kampagnen-Management-System zu implementieren, das dem Marketingteam ermöglicht, gezielte Kampagnen effizient zu gestalten und die Kundenansprache zu verbessern.

Die Lösung umfasst Funktionen, die den gesamten Ablauf einer E-Mail-Kampagne abdecken: von der Bestimmung und Segmentierung der Zielgruppen über die Erstellung ansprechender Landing-Pages bis hin zur Durchführung der Kampagne und der Auswertung der Reaktionen. Die Segmentierung wird es dem Marketingteam ermöglichen, Prospects (potenzielle Gäste) spezifisch anzusprechen und gezielt auf deren Interessen einzugehen.

Diese Single-Page-Aplication (SPA) wird für eine intuitive Bedienung im Responsive Design entwickelt, sodass Marketingmitarbeitende jederzeit und von verschiedenen Geräten aus Zugriff auf die Kampagnen-Tools und -Daten haben. Die Applikation wird zwei Benutzergruppen umfassen: Prospects als Nutzniesser der Kampagnen und Marketingmitarbeitende als Administratoren. Um die Datenverwaltung zu optimieren, wird die Plattform auf verschiedene APIs zugreifen können. Eine interne API der ERP-Software stellt Kundenadressen bereit, während eine externe API der Firma Küba AG Marktdaten wie Adressen potenzieller Neukunden bereitstellt. Zudem wird die Plattform ein API zur Verfügung stellen, die Performance-Daten der Kampagnen für ein externes Dashboard liefert, um eine umfassende Analyse und Berichterstattung zu ermöglichen.

Durch die Einführung dieser Lösung sollen die Effizienz und Zielgenauigkeit des Marketings von Hotel Grand Pilatus gesteigert und die Grundlage für eine datengetriebene Marketingstrategie geschaffen werden.

3. Formale Beurteilungskriterien und korrektes Zitieren

Ein wichtiges Beurteilungskriterium über die gesamte Transferaufgabe stellen die Vorgaben des schriftlichen Leitfadens sowie des Zitierleitfadens dar. Folgende Kriterien werden in der Transferaufgabe in Bezug auf die formalen Bedingungen sowie dem korrekten Zitieren bewertet:

Beurteilungskriterien	Maximal 10 Punkte mit folgender Gewichtung:
	<ul style="list-style-type: none"> ▪ Neutrale Schriftsprache und angemessene Wortwahl sowie korrekte Grammatik, Rechtschreibung und Zeichensetzung (40 %). ▪ Korrekte Formatierung gemäss schriftlichem Leitfaden (10 %). ▪ Die inhaltlichen Elemente gemäss schriftlichem Leitfaden sind vorhanden (10 %). ▪ Bezugnahme oder Übernahme von fremden Texten, Darstellungen oder Gedankengut wurde im redaktionellen Text gemäss Zitierleitfaden korrekt zitiert (40 %).

4. Aufgabenstellungen

4.1. Softwarearchitektur, technologisches Design und Entwicklungsumgebung

Im Rahmen des vorgesehenen Entwicklungsprojektes hat die Projektleitung dich beauftragt, eine tragfähige Softwarearchitektur und technologisches Design der verteilten Webapplikation zu erstellen. Weiter ist die konkrete Entwicklungsumgebung mit den notwendigen Tools festzulegen, um eine optimale Zusammenarbeit im Team und eine höchstmögliche Effizienz der Systementwicklung zu gewährleisten. Berücksichtige dabei insbesondere die Informationen in der Ausgangslage.

Aufgabe(n)	<ol style="list-style-type: none"> 1. Begründete Wahl des Technologie-Stacks: Begründe deine Auswahl der Technologie-Stack-Elemente unter Berücksichtigung von Skalierbarkeit, Performance, Kosten und Wartbarkeit. Zudem sollte auf die spezifischen Anforderungen der Applikation (verteilte Architektur, Client- und Server-Kommunikation) eingegangen werden. 2. Design der Softwarearchitektur für die verteilte Applikation: Entwurf einer Architektur, die die Trennung von Client und Server klar definiert. Es muss ein adäquates Diagramm ausgewählt und angewendet werden, um die Beziehungen zwischen den Komponenten und ihren Verantwortlichkeiten darzustellen. 3. Toolbasiertes Design und Dokumentation der APIs: Begründete Wahl der API-Technologie sowie deren Dokumentations-Tools. Vollständige Erstellung des API-Designs für die gemäss der Ausgangslage angebotene Web-API. Dabei ist aber nur ein Datenobjekt vorzusehen. 4. Auswahl einer geeigneten Entwicklungsumgebung unter begründeter Berücksichtigung der KI-Unterstützung.
Form der Antwort	<p>Die Struktur der Antwort muss sich an die Aufgabenstruktur halten und eine entsprechende Titelstruktur aufweisen.</p> <p>Die geforderten Auswahlen sind jeweils mit einer fundierten Herleitung und mit mindestens je 2 Quellen (mind. je 1 aktuelles Fachbuch sowie je 1 Internetquelle) gut begründet darzulegen.</p> <p>Jedes Element des geforderten Diagramms muss kurz (in 2-3 Sätzen) beschrieben sein.</p> <p>Das API-Design sollte in einem maschinenlesbaren Format (z.B. YAML) dokumentiert werden, um eine einfache Integration und spätere Erweiterungen zu ermöglichen.</p>
Umfang der Antwort	<p>Das Dokument muss ca. 7, jedoch max. 11 Seiten umfassen, wobei die Begründungen je ca. $\frac{3}{4}$ - 1 A4-Seite sein müssen.</p>
Hinweise	<p>Berücksichtige für den Technologie-Stack folgende Aspekte:</p> <ul style="list-style-type: none"> - Serverseitig: Middleware, Frameworks, Programmiersprache, Datenbank - Clientseitig: Frameworks, Programmiersprache, Runtime-Umgebungen <p>Die Softwarearchitektur muss folgende Aspekte abdecken:</p> <ul style="list-style-type: none"> - Server- und Clientseitige Architektur mit deren Pakete und Komponenten - Integration mit den Umsystemen über adäquate Schnittstellen <p>Berücksichtige für die Entwicklungsumgebung Tools für folgende Aspekte:</p> <ul style="list-style-type: none"> - UI-Wireframing - IDE - Unit-Testing - API-Testing - KI-Unterstützung - Versionskontrolle

Beurteilungskriterien

Maximal 25 Punkte mit folgender Gewichtung:

- Wahl des Technologie-Stacks (25 %):
 - Literatur basierte, fundierte Begründung
 - Vollständigkeit und Adäquatheit der Elemente
- Design der Softwarearchitektur (35 %):
 - Vollständigkeit und Modularität der Architektur
 - Literatur basierte Auswahl eines geeigneten Diagramms und deren korrekte Anwendung
 - Pakete, Komponenten, Assoziationen und Interfaces sind korrekt vorhanden und beschrieben
- API-Design und Dokumentation (25 %):
 - Nachvollziehbarkeit der Wahl und Zweckmässigkeit der API-Technologie und des Dokumentationstools
 - Vollständigkeit, Konsistenz und Verständlichkeit der toolbasierten API-Spezifikation in einem technischen Datenformat (z.B. YAML)
- Auswahl und Konfiguration der Entwicklungsumgebung (15 %):
 - Vollständigkeit, Effizienz und Zweckmäßigkeit der gewählten Tools
 - Berücksichtigung und Begründung der passenden KI-Unterstützung

4.2. Server-App

In einem ersten Implementationsschritt soll eine serverseitige Basis-Applikation mit dem vorgeschlagenen Technologie-Stack umgesetzt werden. Diese bildet die Grundlage für die Client-Entwicklung und die Integration der Umsysteme.

Aufgabe	<ol style="list-style-type: none"> 1. Serverseitige Basis-Applikation: Implementiere gemäss dem in Aufgabe 1 gewähltem Technologie-Stacks die Basis-Applikation. Die nachfolgenden Aufgaben 2 und 3 sind in dieser Basis-Applikation einzubinden. 2. Einbindung der Persistenz: Entwirf und implementiere die in Aufgabe 1 gewählte Non-SQL-Datenbank. Gib in der Basis-Applikation einen Data-Access-Layer, in welchem Datenobjekte in einem festgelegten Format (z.B. JSON) eingefügt, gelesen, geändert und gelöscht werden können. 3. Web-API gemäss Design implementieren: Setze das in Aufgabe 1 entworfene API um und biete den Service in der Basis-Applikation an. Sichere den Zugriff vor nicht autorisierten Aufrufen. 4. Toolbasierter API-Tests: Erstelle mit dem in Aufgabe 1 gewählten API-Testing-Tool ein Set von Testfällen, um die API funktional vollständig zu testen. Dokumentiere und behebe gefundene Fehler.
Form der Antwort	<p>Jede Aufgabe ist mit einem Arbeitsprotokoll mit den essenziellen Schritten zu dokumentieren. Dabei sind die einzelnen Arbeitsschritte, welche zur Erfüllung der Aufgabe führen, kurz zu erläutern.</p> <p>Der Source-Code der Basis-Applikation ist als Zip-Format abzugeben.</p>
Umfang der Antwort	Die Dokumentation muss ca. 6, jedoch max. 10 Seiten umfassen.
Hinweise	Der Code muss einen für aussenstehende gut verständlichen Inline-Kommentar in Deutsch oder Englisch enthalten.
Beurteilungskriterien	<p>Maximal 25 Punkte mit folgender Gewichtung:</p> <ul style="list-style-type: none"> ▪ Erstellung der serverseitigen Basis-Applikation (30 %) <ul style="list-style-type: none"> ○ Verständliche und vollständige Dokumentation ○ Klarheit und Effizienz der Implementierung mit verständlichen Inline-Kommentaren ○ Korrekte Nutzung des gewählten Technologie-Stacks ○ Erfüllung der grundlegenden Anforderungen der Server-App ▪ Einbindung der Persistenz (25 %) <ul style="list-style-type: none"> ○ Verständliche und vollständige Dokumentation ○ Sinnvoller Entwurf für die Non-SQL-DB ○ Saubere Implementierung und Einbindung an die Server-App im Data-Access-Layer ▪ Web-API Implementierung (25 %) <ul style="list-style-type: none"> ○ Verständliche und vollständige Dokumentation ○ Einhaltung des API-Designs und funktionaler Anforderungen ▪ Toolbasierter API-Tests (20 %) <ul style="list-style-type: none"> ○ Verständliche und vollständige Dokumentation ○ Vollständigkeit und Tiefe der durchgeführten Tests ○ Klarheit und Nachvollziehbarkeit der Testergebnisse ○ Behebung und Dokumentation möglicher Probleme

4.3. Client-Apps

Aufgabe	<ol style="list-style-type: none"> 1. Wireframe für SPA Web-App erstellen: Entwickle ein Wireframe mit dem in Aufgabe 1 gewählten Tool für die SPA Web-App, der die grundlegende Benutzeroberfläche, Navigation und ca. 5 der zentralen Formulare aus Sicht des Sachbearbeiters darstellt. Legen Sie in der Dokumentation die zu berücksichtigenden Quellen bzw. Vorgaben dar. 2. Wireframe für Mobile-App erstellen: Entwickle ein Wireframe mit dem in Aufgabe 1 gewählten Tool für die Mobile-App, der die grundlegende Benutzeroberfläche, Navigation und ca. 5 der zentralen Formulare darstellt. Legen Sie in der Dokumentation die zu berücksichtigenden Quellen bzw. Vorgaben dar. 3. SPA Web-App Prototyp entwickeln: Setze die Vorgabe gemäss dem in Teilaufgabe 1 erstellten Wireframe in dem gewählten Tool von Aufgabe 1 um.
Form der Antwort	<p>Das Resultat der Aufgabe 1 und 2 ist in einem Dokument darzulegen und die einzelnen Formulare bzw. Bildschirmseiten mit einem Print-Screen zu dokumentieren.</p> <p>In Aufgabe 3 sind die essenziellen Schritte mit einem Arbeitsprotokoll zu dokumentieren. Dabei sind die einzelnen Arbeitsschritte, welche zur Erfüllung der Aufgabe führen, kurz zu erläutern.</p> <p>Der Source-Code der SPA-Web-App ist als Zip-Format abzugeben.</p>
Umfang der Antwort	Das Dokument muss ca. 6, jedoch max. 10 Seiten umfassen.
Hinweise	<p>Berücksichtige im Wireframing die Grundsätze und Prinzipien des UI/UX-Designs und der bekannten UI-Guidelines.</p> <p>Der Source-Code muss einen für aussenstehende gut verständlichen Inline-Kommentar in Deutsch oder Englisch enthalten.</p>
Beurteilungskriterien	<p>Maximal 25 Punkte mit folgender Gewichtung:</p> <ul style="list-style-type: none"> ▪ Wireframe für SPA Web-App (30 %) <ul style="list-style-type: none"> ○ Bedienbarkeit und Ästhetik sind klar, verständlich, sinnvoll und gemäss den gewählten UI-Guidelines ○ UX-Prinzipien wie Navigation, Benutzerführung, Responsiveness usw. sind berücksichtigt ○ Vollständige Darstellung der wichtigsten Interaktionen und der ca. 5 zentralen Formulare ▪ Wireframe für Mobile-App (30 %) <ul style="list-style-type: none"> ○ Bedienbarkeit und Ästhetik sind klar, verständlich, sinnvoll und gemäss den Plattform-Vorgaben iOS und Android ○ Vollständige Darstellung der wichtigsten Interaktionen und der ca. 5 zentralen Formulare ▪ Prototyp der SPA Web-App (40 %) <ul style="list-style-type: none"> ○ Verständliches und vollständiges Arbeitsprotokoll ○ Vollständiger Source-Code mit Inline-Dokumentation ○ Implementierung der Inhalte gemäss dem in der Teilaufgabe 1 erstellten Wireframe ○ Technische Umsetzung mit dem in Aufgabe 1 gewählten Framework

4.4. Deployment und Test

Aufgabe	<ol style="list-style-type: none"> 1. Planung und Design des Deployments: Entwickle einen umfassenden Plan für das Deployment der Webanwendung. Dieser Plan sollte die fundiert begründete Wahl der Infrastruktur sowie die Auswahl geeigneter DevOps-Tools für Continuous Integration und Continuous Deployment (CI/CD) beinhalten. Der CI/CD-Prozess sollte die Automatisierung des Builds, der Tests und des Deployments umfassen. Verwende dabei relevante Quellen, um deine Entscheidungen zu stützen und Best Practices zu berücksichtigen. 2. Planung des Deployments der Mobile-Apps: Entwickle einen detaillierten Plan für das Deployment der Mobile-Apps auf den Plattformen iOS und Android. Achte auf die spezifischen Anforderungen beider Plattformen, einschließlich Zertifizierung, Code-Signing, Freigabeprozesse und die Veröffentlichung in den jeweiligen App-Stores. Nutze verfügbare offizielle Dokumentationen und Best Practices, um sicherzustellen, dass die Apps den Richtlinien der jeweiligen Plattformen entsprechen. Erkläre und begründe die relevanten Schritte im Deployment-Prozess. 3. Erstelle ein Testprotokoll des manuellen Systemtests des Prototyps der Web-App: Führe einen dokumentierten manuellen Systemtest des erstellten Prototyps durch. Teste alle wesentlichen Funktionen (4 grundlegende Basisfälle mit neu, suchen, ändern, löschen) der Webanwendung. Dokumentiere die Ergebnisse der Tests, gefundene Fehler und Verbesserungsvorschläge. Der Testbericht sollte alle durchgeführten Testfälle und deren Ergebnisse kurz und mit Print-Screens beschreiben.
Form der Antwort	Das Resultat der Aufgabe 1, 2 und 3 ist in einem Dokument darzulegen. Die Aufgabe 3 ist mit einem Arbeitsprotokoll zu dokumentieren. Dabei sind die einzelnen Arbeitsschritte, welche zur Erfüllung der Aufgabe führen, kurz zu erläutern.
Umfang der Antwort	Das Dokument muss ca. 6, jedoch max. 10 Seiten umfassen.
Hinweise	Beim Systemtest sind die entsprechenden Print-Screens im Testprotokoll zu hinterlegen.
Beurteilungskriterien	<p>Maximal 15 Punkte mit folgender Gewichtung:</p> <ul style="list-style-type: none"> ▪ Planung und Design des Deployments – (35 %) <ul style="list-style-type: none"> ○ Qualität und Detaillierung des Deployment-Plans ○ Begründete und mit Quellen belegte Auswahl der Infrastruktur und DevOps-Tools ○ Nachvollziehbarkeit und Effizienz des CI/CD-Prozesses ▪ Planung des Deployments der Mobile-Apps (35 %) <ul style="list-style-type: none"> ○ Vollständigkeit und Detailliertheit des mit Quellen belegten Deployment-Plans für die Mobile-Apps ○ Berücksichtigung der Plattformanforderungen (iOS, Android) ○ Realisierbarkeit und Einhaltung der Store-Richtlinien und Zertifizierungsprozesse ▪ Dokumentierter manueller Systemtest (30 %) <ul style="list-style-type: none"> ○ Vollständigkeit und Detailliertheit des Systemtests ○ Nachvollziehbarkeit der Testergebnisse und Berichterstellung ○ Korrektur und Vorschläge zur Optimierung basierend auf den Testergebnissen