# Vor-IPA Dokumentation





Ein Bild, das ClipArt enthält.

Automatisch generierte Beschreibung



**Firmenname** Technische Fachschule Bern – Abteilung Informatik  
**Berufsschule** Technische Fachschule Bern  
**Experten** Raffaele Zompicchiatti, Matthias Heimberg,   
 Claude Fankhauser, Cuno Vollenweider   
**ver. Fachkraft** Raffaele Zompicchiatti  
**Fachrichtung** Betriebsinformatik  
**Projektmethode** Hermes  
**Ausgabedatum** Vor-IPA 2021 Kanton Bern

**Autor** Michael Schmid  
**Version** V 0.0  
**Status** In Bearbeitung

# Teil 1: Ablauf und Umgebung

**Vor-IPA Projektname:** Datenbank Web Applikation als Azure Cloud Docker Container  
**Autor:**  Michael Schmid





Ein Bild, das ClipArt enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

# Kurzfassung des Vor-IPA-Berichts

Dieser Abschnitt dient als eine Zusammenfassung für die Vor-IPA und richtet sich an Fachpersonen.

## Ausgangssituation

Im Rahmen des Werkstattunterrichts sollen für die laufenden Projekte viele Webservices und Datenbanken etc. in Zukunft in der Cloud laufen. Im Sinne der Wartbarkeit und Portabilität sollen diese Services als Docker Container laufen können. Um ausserdem die Sicherheit der verwendeten Datenbanken zu erhöhen, sollen diese hinter der Azure Firewall versteckt bleiben und nur von einem Webservice, der die einzige Schnittstelle zur Aussenwelt darstellt, ansteuerbar sein.

## Umsetzung

Im Rahmen dieser Vor-IPA wird eine Datenbank Webapplikation als Azure Cloud Docker Container eingerichtet. Die verwendeten Technologien sind vom Auftraggeber vorgeschrieben. Das gesamte Projekt wird mit der Projektmethode Hermes 5 IPA geplant und durchgeführt. Sämtliche benötigte Informationen und Schritte wie eine NodeJS/Express JavaScript Webapplikation als Dockercontainer in der Cloud zur Verfügung gestellt werden kann, sind in diesem Dokument festgehalten.

## Ergebnis

Die Datenbank Webapplikation wurde als Dockercontainer in der Azure Cloud zur Verfügung gestellt. Es können Daten per REST an die Webapplikation gesendet werde, welche diese in eine in Azure ansässigen MySQL Datenbank abspeichert.

# Inhaltsverzeichnis

[Vor-IPA Dokumentation 1](#_Toc88727782)

[Teil 1: Ablauf und Umgebung 2](#_Toc88727783)

[1. Kurzfassung des Vor-IPA-Berichts 3](#_Toc88727784)

[1.1 Ausgangssituation 3](#_Toc88727785)

[1.2 Umsetzung 3](#_Toc88727786)

[1.3 Ergebnis 3](#_Toc88727787)

[2. Inhaltsverzeichnis 4](#_Toc88727788)

[3. Aufgabenstellung 8](#_Toc88727789)

[3.1 Thematik 8](#_Toc88727790)

[3.2 Ausgangslage 8](#_Toc88727791)

[3.3 Detaillierte Aufgabenstellung 8](#_Toc88727792)

[3.4 Mittel und Methoden 9](#_Toc88727793)

[3.5 Vorkenntnisse 10](#_Toc88727794)

[3.6 Vorarbeiten 10](#_Toc88727795)

[3.7 Neue Lerninhalte 10](#_Toc88727796)

[3.8 Arbeiten in den letzten 6 Monaten 10](#_Toc88727797)

[4. Standards 11](#_Toc88727798)

[5. ISDS 12](#_Toc88727799)

[5.1 Informationsschutz & Datenschutz 12](#_Toc88727800)

[5.1.1 Schützenswerte Daten 12](#_Toc88727801)

[5.2 Schutzbedarfsanalyse 12](#_Toc88727802)

[5.2.1 Vor-IPA schützenswerte Daten 13](#_Toc88727803)

[5.3 Technische Massnahmen 14](#_Toc88727804)

[5.4 Organisatorische Massnahmen 14](#_Toc88727805)

[6. Organisation der Vor-IPA Ergebnisse 14](#_Toc88727806)

[6.1 Arbeitsplatz 14](#_Toc88727807)

[6.1.1 Arbeitsnotebook: 15](#_Toc88727808)

[6.2 Dokumentablage 16](#_Toc88727809)

[6.3 Dokumentsicherung der Vor-IPA 17](#_Toc88727810)

[6.3.1 OneDrive 17](#_Toc88727811)

[6.3.2 Wiederherstellung der Daten mit OneDrive 17](#_Toc88727812)

[6.3.3 Git 18](#_Toc88727813)

[6.3.4 Wiederherstellung der Daten mit Git 18](#_Toc88727814)

[6.4 Versionsverwaltung 19](#_Toc88727815)

[6.4.1 Zurücksetzen der Daten mit Git 19](#_Toc88727816)

[6.4.2 Vorher 19](#_Toc88727817)

[6.4.3 Hard Reset 20](#_Toc88727818)

[6.4.4 Nachher 21](#_Toc88727819)

[7. Projektvorgehen 21](#_Toc88727820)

[7.1 Projektmethode 22](#_Toc88727821)

[7.2 Szenarien 22](#_Toc88727822)

[7.3 Die verschiedenen Phasen im Überblick 22](#_Toc88727823)

[7.3.1 Initialisierung 22](#_Toc88727824)

[7.3.2 Konzept 22](#_Toc88727825)

[7.3.3 Realisierung 23](#_Toc88727826)

[7.3.4 Einführung 23](#_Toc88727827)

[7.4 Module 23](#_Toc88727828)

[8. Vor-IPA Projektorganisation u. Projektrollen 24](#_Toc88727829)

[8.1 Projektorganisation 25](#_Toc88727830)

[8.2 Projektrollen 25](#_Toc88727831)

[9. Risikoanalyse 27](#_Toc88727832)

[9.1 Legende 27](#_Toc88727833)

[9.1.1 Schadensausmass 27](#_Toc88727834)

[9.1.2 Eintrittswahrscheinlichkeit 27](#_Toc88727835)

[9.2 Risikograph 28](#_Toc88727836)

[9.3 Risikomatrix 29](#_Toc88727837)

[10. Zeitplan 30](#_Toc88727838)

[11. Arbeitsjournale 31](#_Toc88727839)

[11.1 Montag, 15. November 2021 31](#_Toc88727840)

[11.2 Dienstag, 16. November 2021 32](#_Toc88727841)

[11.3 Mittwoch, 17. November 2021 34](#_Toc88727842)

[11.4 Freitag, 19. November 2021 35](#_Toc88727843)

[11.5 Montag, 22. Juni 2021 38](#_Toc88727844)

[11.6 Dienstag, 23. Juni 2021 39](#_Toc88727845)

[12. Abschlussberichte (Beispiele) 40](#_Toc88727846)

[12.1 Schlussbericht Ist/Soll 40](#_Toc88727847)

[12.2 Erkenntnisse 41](#_Toc88727848)

[12.2.1 Anforderungen 41](#_Toc88727849)

[12.2.2 Zeit 41](#_Toc88727850)

[12.2.3 Einsatzmittel 41](#_Toc88727851)

[12.2.4 Qualität der Arbeit 41](#_Toc88727852)

[12.1 Persönliches Fazit / Schlussreflexion 41](#_Toc88727853)

[Teil 2: Projektdokumentation 43](#_Toc88727854)

[13. Initialisierung 44](#_Toc88727855)

[13.1 Beschreibung Ist-Zustand / Soll-Zustand 44](#_Toc88727856)

[13.1.1 Istzustand 44](#_Toc88727857)

[13.1.2 Sollzustand 44](#_Toc88727858)

[13.2 Ziele 44](#_Toc88727859)

[13.2.1 Smart-Ziele 45](#_Toc88727860)

[13.2.2 Persönliche Vorgehensziele 45](#_Toc88727861)

[13.2.3 Projektziele 46](#_Toc88727862)

[13.3 Anforderungen 46](#_Toc88727863)

[13.3.1 Funktionale Anforderungen 46](#_Toc88727864)

[13.3.2 Nicht funktionale Anforderungen 47](#_Toc88727865)

[13.4 Abgrenzungskriterien 47](#_Toc88727866)

[13.5 Schnittstellen und Umfeld 47](#_Toc88727867)

[13.6 Rahmenbedingungen 48](#_Toc88727868)

[13.6.1 Technische Rahmenbedingungen 48](#_Toc88727869)

[13.6.2 Organisatorische Rahmenbedingungen 48](#_Toc88727870)

[14. Konzept 50](#_Toc88727871)

[14.1 Erläuterung der verwendeten Mittel und Methoden 50](#_Toc88727872)

[14.2 Systemarchitektur 50](#_Toc88727873)

[14.2.1 Konzept Client 51](#_Toc88727874)

[14.2.2 Konzept Azure Cloud 52](#_Toc88727875)

[14.2.3 Softwarearchitekture 56](#_Toc88727876)

[14.2.4 Zugriffsberechtigungen 58](#_Toc88727877)

[14.2.5 Deploymentkonzept 59](#_Toc88727878)

[14.3 Anwendungsfälle 60](#_Toc88727879)

[14.4 Datenbankkonzept / DV-Konzept 61](#_Toc88727880)

[14.4.1 ERM 61](#_Toc88727881)

[14.5 Testkonzept 62](#_Toc88727882)

[14.6 Testart 62](#_Toc88727883)

[Testziele 62](#_Toc88727884)

[14.6.1 Testobjekte 62](#_Toc88727885)

[14.6.2 Testarten 63](#_Toc88727886)

[14.6.3 Testfälle 63](#_Toc88727887)

[14.6.4 Testvoraussetzungen 63](#_Toc88727888)

[14.6.5 Mängeklassifizierung 63](#_Toc88727889)

[14.6.6 Testumgebung 64](#_Toc88727890)

[14.6.7 Testinfrastruktur 64](#_Toc88727891)

[14.6.8 Testfall Vorlage 65](#_Toc88727892)

[15. Realisierung 65](#_Toc88727893)

[16. Abbildungsverzeichnis 66](#_Toc88727894)

[17. Tabellenverzeichnis 66](#_Toc88727895)

[18. Quellenverzeichnis 67](#_Toc88727896)

[19. Glossar 67](#_Toc88727897)

[20. Anhänge 67](#_Toc88727898)

# Aufgabenstellung

In der Aufgabenstellung sind alle Informationen enthalten, welche für die Realisation der Vor-IPA notwendig sind.

## Thematik

Datenbank Web Applikation als Azure Cloud Docker Container

## Ausgangslage

Im Rahmen des Werkstattunterrichts sollen für die laufenden Projekte viele Webservices und Datenbanken etc. in Zukunft in der Cloud laufen. Im Sinne der Wartbarkeit und Portabilität sollen diese Services als Docker Container laufen können. Um ausserdem die Sicherheit der verwendeten Datenbanken zu erhöhen, sollen diese hinter der Azure Firewall versteckt bleiben und nur von einem Webservice, der die einzige Schnittstelle zur Aussenwelt darstellt, ansteuerbar sein.

## Detaillierte Aufgabenstellung

**Zielbestimmungen:**

**Musskriterien:**

|  |  |
| --- | --- |
| **ID** | **MK** |
| 1 | Die in dieser Arbeit verwendeten Features von Docker Instanzen auf Azure Cloud sind analysiert und dokumentiert |
| 2 | Der Fokus des Dockercontainers liegt auf einer NodeJS/Express JavaScript Web Applikation |
| 3 | Die im Docker laufende Applikation muss von aussen per REST ansteuerbar sein |
| 4 | Es soll eine für Lernende des 3. Lehrjahrs der Abteilung I verständliche, Schritt für Schritt Anleitung geschrieben werden, wie eine NodeJS/Express JavaScript Webapplikation als Dockercontainer in der Cloud verfügbar gemacht werden kann |
| 5 | Es soll eine einfache MySQL Datenbank in Azure konfiguriert werden, welche von der Webapplikation angesteuert wird |
| 6 | Man muss Daten von aussen per REST an die Webapplikation senden können (Post), welche diese in der MySQL Datenbank abspeichert |
| 7 | Man muss von aussen dieselben Daten, welche zuvor in der Datenbank gespeichert wurden, wieder per REST (Get) abholen können |
| 8 | Die Webapplikation muss mit Unittests getestet sein. |

**Abgrenzungskriterien:**

* Es reicht, wenn die Webapplikation einen einzigen REST Endpunkt hat, da der Fokus auf der Machbarkeit und nicht auf der eigentlichen Funktion liegt.
* Das Bereitstellen des Containers mit der Webapplikation in der Cloud kann wahlweise manuell oder per Skript erfolgen
* Der Aufbau der MySQL Tabellen und Daten sind nicht wichtig und daher frei wählbar. Diese müssen nur von der Webapplikation verwendet werden können.
* Das Ansteuern der REST Schnittstellen kann irgendwie erfolgen (z.B. Postman). Es muss nur deren korrekte Funktion gezeigt werden können.

**Produkteinsatz:**

**Anwendungsbereiche:**

* Grundstein zur Verwendung cloudbasierter Lösung für laufende Projekte in der Werkstatt

**Zielgruppe:**

* Lernende Abteilung I

**Produktumgebung:**

**Software:**

* JavaScript
* Azure Cloud

**Hardware:**

* Arbeitslaptop

**Produktfunktionen:**

## Mittel und Methoden

* Projektmethode Hermes 5.1
* Azure Docker
* MySQL Datenbank
* JavaScript

## Vorkenntnisse

|  |  |
| --- | --- |
| **Bereich** | **Kenntnisse** |
| MySQL | In mehreren Informatikmodulen wurde mit MySQL Datenbanken gearbeitet. Darüber hinaus wurde mit anderen SQL basierten Datenbanken gearbeitet. |
| JavaScript | In zwei Webmodulen kam JavaScript zum Einsatz, darüber hinaus in einigen Werkstatt-Projekten. |
| NodeJS/Express | Im Rahmen der Werkstatt 3. Lehrjahr wurde teilweise mit NodeJS/Express gearbeitet. |
| PHP | Im Modul 153 wurde ich mit den Basics von PHP vertraut gemacht. |

Tabelle 1: Vorkenntnisse

## Vorarbeiten

Erstellen eines eigenen Azure Active Directory und das Verschieben der Schullizenz darauf. Neue Lerninhalte.

## Neue Lerninhalte

Azure Cloud, Docker

## Arbeiten in den letzten 6 Monaten

Initialisierungsphase Projekt „Lernende Dashboard“

# Standards

In diesem Kapitel werden verschieden Standards hinsichtlich der Formatierung und des Projektvorgehens festgelegt. Einige Standards wie z.B. die Projektmethode werden von der Technischen Fachschule vorgegeben, währenddem andere vom Kandidaten festgelegt werden.

|  |  |
| --- | --- |
| **Standards** | **Beschreibung** |
| **Dokumentvorlage** | Die Dokumentvorlage wurde vom Kandidaten erstellt. Der Kandidat hat sich beim Erstellen der Vorlage von anderen Dokumentationen inspirieren lassen |
| **Projektmethode** | Die Technische Fachschule Bern schreibt für die Vor-IPA die Projektemethode Hermes 5 IPA vor |
| **Testkonzept** | Das Testkonzept erfolgt nach der HERMES Projektmethode |
| **Clean Code** | Die Realisierung wird mit Clean Code ausgeführt. Diese basieren auf den Grundprinzipien des Buches: «Clean Code, A Handbook of Agile Software Craftsmanship» |

# ISDS

In diesem Abschnitt wird festgehalten, ob es im Zusammenhang mit der Vor-IPA schützenswerte Daten gibt. Zudem werden die Anforderungen an die Informationssicherheit und Datenschutz dokumentiert.

## Informationsschutz & Datenschutz

Es muss geprüft werden, ob und wie im Zusammenhang mit der Vor-IPA der Informationsschutz und Datenschutz gewährleistet ist, sofern dies notwendig ist.

### Schützenswerte Daten

* Personendaten: Alle Angaben, die sich auf eine bestimmte oder bestimmbare Person beziehen.
* Dabei besonders schützenswerte Daten umfassen:
  + Religiöse, weltanschauliche, politische oder gewerkschaftliche Ansichten und Tätigkeiten
  + Die Gesundheit, die Intimsphäre oder die Rassenzugehörigkeit
  + Massnahmen der sozialen Hilfe
  + Administrative oder strafrechtliche Verfolgungen und Sanktionen

## Schutzbedarfsanalyse

|  |  |  |  |
| --- | --- | --- | --- |
| **Kriterien** | **Fragen** | **Antworten** | **Begründung** |
| **Vertraulichkeit** | Gibt es in der Vor-IPA schützenswerte Daten? | **Ja** | Lediglich die Azure Logindaten sind schützenswert, da sie mit dem Active Directory der Abteilung verknüpft sind. |
| Gibt Daten, welche nicht in der Dokumentation ersichtlich sind? | **Ja** | Meine Azure Logindaten und die Datenbankzugangsdaten. Datenbankzugangsdaten. |
| Werden innerhalb der Applikation sensible bzw. schützenswerte Daten als Eingaben von Benutzern aufgenommen? | **Nein** | Es werden nur Testdaten aufgenommen. |
| **Verfügbarkeit** | Muss die Applikation rund um die Uhr verfügbar sein? | **Nein** | Die Webapplikation muss dann verfügbar sein, wenn der Berufsbildner dies wünscht. Da das Guthaben auf Azure beschränkt ist, ist die Applikation nicht immer verfügbar. |
| Sind im Falle eines Notfalls Massnahmen nötig? | **Nein** | Da es sich nicht um ein System im produktiven Betrieb handelt, sind im Falle eines Notfalls nicht zwingend Massnahmen nötig. |
| **Integrität** | Muss die Echtzeit, Korrektheit und/oder Unversehrtheit der Daten gewährleistet werden können? | **Nein** | Da die eingegebenen Daten nur für Testzwecke verwendet werden, findet keine Validierung statt, bevor die eingegeben Daten in die Datenbank gespeichert werden. |
| **Nachvollziehbarkeit** | Müssen bestimmte Arbeitsvorgänge aufgezeichnet und nachgewiesen werden können? | **Ja** | Es muss eine Anleitung erstellt werden, in welcher nachvollziehbar ist, wie eine Webapplikation als Container in der Cloud verfügbar gemacht werden kann. |

In dieser Vor-IPA wird nur ein Testsystem entwickelt. Die Überführung in den produktiven Betrieb ist im Rahmen dieser Vor-IPA nicht vorgesehen. Die Benutzer, die verwendet werden, sind nur Testdaten, deswegen müssen weder Massnahmen noch ein ISDS-Konzept definiert werden für die Benutzerdaten.

### Vor-IPA schützenswerte Daten

* Azure Login
* Datenbank Zugangsdaten

## Technische Massnahmen

Die Zugangsdaten für die Datenbank werden in einem separaten File für Umgebungsvariablen gespeichert. Dieses File ist im git ignore File hinterlegt.

## Organisatorische Massnahmen

Es werden weder die Azure-Login Daten noch die Zugangsdaten für die Datenbank in der Dokumentation erwähnt.

# Organisation der Vor-IPA Ergebnisse

In diesem Abschnitt wird alles rund um die Organisation der Vor-IPA aufgezeigt und beschrieben. Dies beinhaltet unter anderem Datenablage, Datenspeicherung und Einrichtung des Arbeitsplatzes.

## Arbeitsplatz

Das Projekt wurde am gewohnten Arbeitsplatz des Kandidaten in der Werkstatt im Raum LOA002 durchgeführt. Leider ist es nicht möglich, ein ruhiges Arbeitszimmer zu reservieren, da es nicht genügend freie Zimmer hat verteilt auf so viele Lernende.  
Der Stuhl des Kandidaten kann individuell angepasst werden.



Abbildung 1: Arbeitsplatz

### Arbeitsnotebook:

Die Technische Fachschule Bern stellt den Lernenden der Abteilung Informatik ein Arbeitsnotebook zur Verfügung.

.

|  |  |  |
| --- | --- | --- |
| Darstellung | Kategorie | Beschreibung |
| Ein Bild, das Text, Elektronik, computer, Computer enthält.  Automatisch generierte Beschreibung | OS | Microsoft Windows 10 Pro |
| Prozessor | Intel(R) Core(TM) i7-8550U CPU @  1.80GHz  1.99 GHz |
| Arbeitsspeicher | 16 GB |
| Hersteller | HP |
| Modellname | HP EliteBook850 G5 |

Tabelle 2: Arbeitsnotebook

## Dokumentablage

Sämtliche für die IPA relevante Dokumente sind lokal auf meinem Arbeitsnotebook abgelegt. Für jede Version der Dokumentation und jede Phase des Projekts wurde ein Unterordner angelegt.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 2: Dokumentablage

## Dokumentsicherung der Vor-IPA

Die Dokumentsicherung erfolgt einerseits über OneDrive. Zusätzlich wurde eine externe SSD angeschafft, auf welcher täglich am Ende des Tages die gesamte Ordnerstruktur der Vor-IPA abgespeicher wird, dafür wird die Backupsoftware Duplicati verwendet.

### OneDrive

Alle Relevanten Dokumente im Zusammenhang mit der Vor-IPA sind in einem Verzeichnis abgelegt, welches mit OneDrive Synchronisiert wird. OneDrive lädt alle Dateien in ihre Cloud hoch. Solange man über eine Internetverbindung und die notwendigen Zugangsdaten verfüge, kann man von überall auf die Cloud zugreifen. Jede einzelne Version der Doku ist in einem separaten Verzeichnis gespeichert, wodurch sichergestellt wird, dass auch auf jede Version zugegriffen werden kann.

### Wiederherstellung der Daten mit OneDrive

Angenommen, das Arbeitsnotebook ist nicht mehr verfügbar. In diesem Fall meldet man sich von einem anderen Gerät aus in der OneDrive Cloud an, um auf die Daten zuzugreifen.

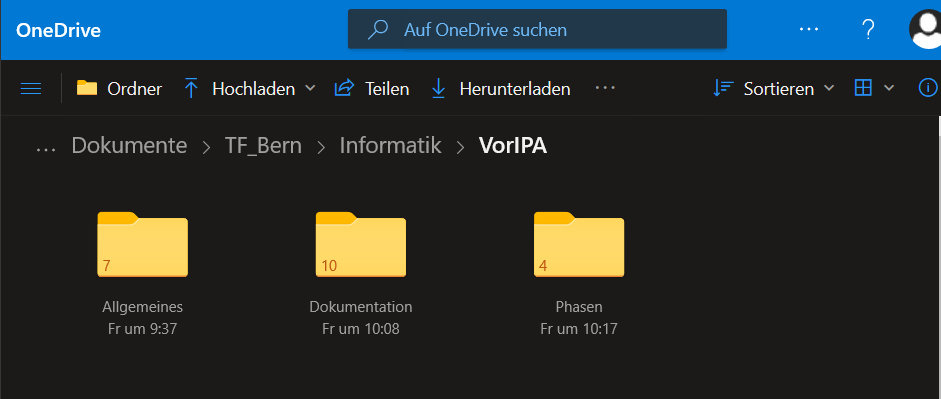


Abbildung 3: OneDrive Cloud

Um die Daten wieder lokal auf das Arbeitsgerät zu holen, wird auf dem Menü-Band die Schaltfläche Download ausgewählt. Nun ist unter «Downloads/OneDrive/Downloads»  
der Ordner VorIPA.zip vorhanden. Wird dieser entpackt, so können die Dokumente wieder lokal bearbeitet werden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 4: OneDrive Datenwiederherstellung

### Git

Neben OneDrive werden zusätzlich alle für die Vor-IPA relevanten Dokumente mit der Software Git gesichert. Gibt es irgendwelche Änderungen, wird sofort ein Commit und ein Push gemacht.

### Wiederherstellung der Daten mit Git

Um eine ältere Version herunterzuladen, ohne den jetzigen Stand zurückzusetzen, kann man auf dem Git-Server zur gewünschten Version navigieren und diese als Zip herunterladen.

Ein Bild, das Text, Screenshot, Monitor, Bildschirm enthält.

Automatisch generierte Beschreibung

Abbildung 5: Datenwiederherstellung mit Git

Nun sieht man auf dem lokalen PC unter Downloads die Version als Zip Ordner.

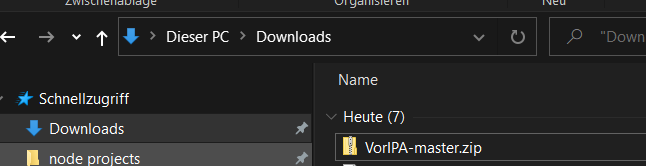


Abbildung 6: Version als Zip Ordner

## Versionsverwaltung

Während der Vor-IPA muss ein Restore aller relevanten Versionen sichergestellt sein.   
Zusätzlich ist es notwendig, dass Informationen zum Zustand der Version erfasst werden sollen. Der Kandidat hat sich für die Versionsverwaltungssoftware Git entschieden.

### Zurücksetzen der Daten mit Git

Es ist notwendig, dass das Repository auf jeden Commit zurückgesetzt werden kann. So können zum Beispiel Fehler zurückgesetzt werden.

### Vorher

Auf dem untenstehenden Bild ist links ein Ausschnitt aller Commits sichtbar.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 7: Git Übersicht Commits

### Hard Reset

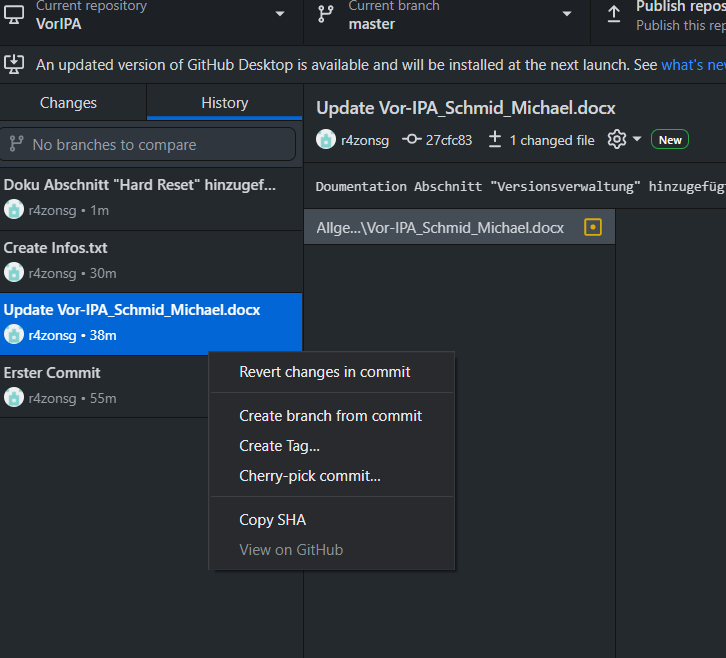
Für einen Hard Reset wird der gewünschte Commit angewählt dann «Revert Changes in Commit». Nun wird lokal auf den Stand dieses Commits zurückgesetzt. Daher werden alle Änderungen zwischen dem ausgewählten und dem neusten Commit gelöscht.  
  


Abbildung 8: Git Hard Reset

### Nachher

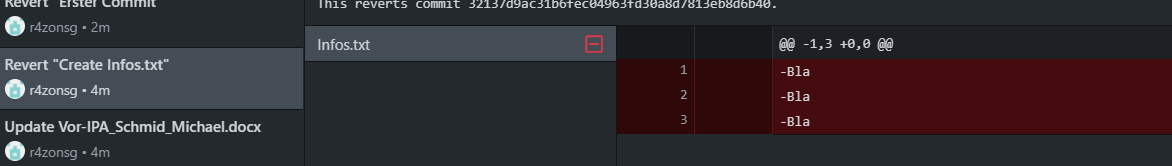


Abbildung 9: Git Hard Reset nacher

Wie man im obenstehenden Bild sieht, wurde auf den Stand des ausgewählten Commits zurückgesetzt. Nun ist die File Infos.txt verschwunden.

# Projektvorgehen

In diesem Abschnitt wird die für die Vor-IPA verwendete Projektmethode definiert und zusammengefasst.

## Projektmethode

Während des ganzen Projekts wurde die Projektmethode HERMES 5.1 verwendet.

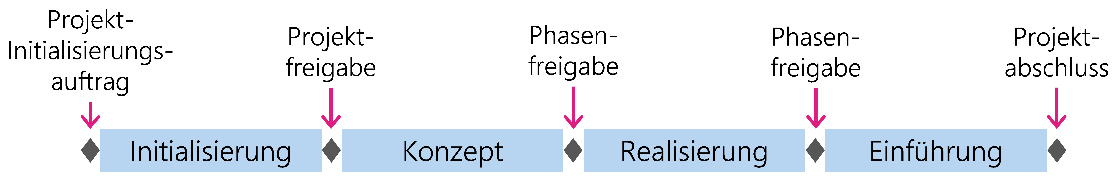


Abbildung 10: Hermes Phasen

## Szenarien

Hermes 5.1 bietet verschiedene Szenarien. Ein Szenario beinhaltet die Module, welche für das jeweilige Szenario relevant sind.  
Das für dieses Projekt verwendete Szenario ist, IT-Individualanwendung, weil im Rahmen dieses Projekts eine Software entwickelt wird.

## Die verschiedenen Phasen im Überblick

Folgenden Abschnitt werden die einzelnen Phasen der Projektmethode Hermes 5 IP erläutert und erklärt.

### Initialisierung

Nachdem der Projektauftrag erteilt wurde, müssen die Projektgrundlagen erarbeitet werden. Sobald dies gemacht wurde, ist eine definierte Ausgangslage sichergestellt. Diese ist die Voraussetzung für die nächste Phase, das Konzept.

### Konzept

In dieser Phase werden die in der Initialisierung gewählten Varianten konkretisiert und auch die Anforderungen werden konkretisiert und vervollständigt. Das Konzept muss für ausstehende Personen mit Fachwissen nachvollziehbar sein. Die Realisierung stützt sich stets auf das Konzept.

### Realisierung

Das IT-System wird realisiert und parallel getestet. Stetiges Ziel dabei ist es, alles zu unternehmen, damit es eine reibungslose Einführung gibt.

### Einführung

Das neue System wird eingeführt und während der Übergangsphase betreut. Die Dokumentation fungiert später als Hilfestellung und Nachschlagewerk für den Kunden.  
Nach der Einführung wird das Projekt abgeschlossen.

Achtung in diesem Projekt ist keine Einführungsphase vorgesehen, da die Applikation im Rahmen dieser Vor-IPA nicht für den produktiven Betrieb entwickelt wird.

## Module

Hermes 5.1 gibt basierend auf der auf dem gewählten Szenario verschiedene Module vor, welche auf die verschiedenen Phasen verteilt werden.   
In unterstehender Grafik ist festgehalten, welches Modul in welcher Phase vorkommt im Szenario «IT-Individualanwendung»

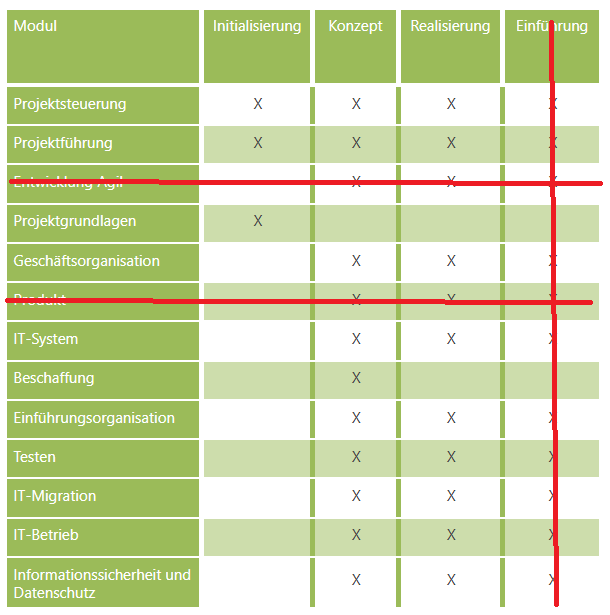


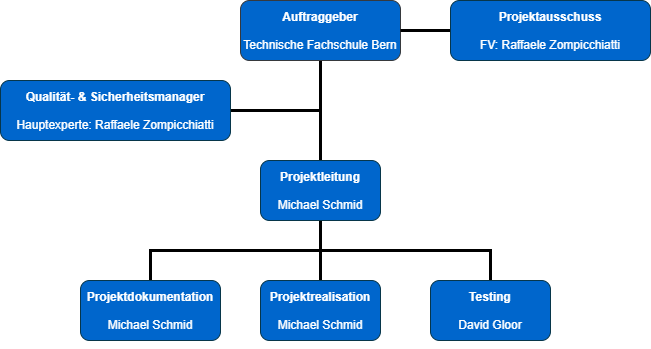
Abbildung 11: Hermes Module

# Vor-IPA Projektorganisation u. Projektrollen

Im folgenden Kapitel sind die Projektrollen und die Projektorganisation aufgeführt.   
Da es sich hier um die Vor-IPA handelt gibt es nur einen Experten, welcher gleichzeitig der Fachvorgesetzte ist.

## Projektorganisation

Abbildung 12: Projektorganisation

In der untenstehenden Grafik ist die Projektorganisation als Diagramm ersichtlich.

## Projektrollen

In der untenstehenden Tabelle sind die Projektrollen aufgeführt.

|  |  |
| --- | --- |
| **Rolle** | **Beschreibung** |
| **Auftraggeber**  Technische Fachschule Bern | Der Projektauftraggeber ist die Technische Fachschule Bern. |
| **Projektausschuss** Raffaele Zompicchiatti | Die Rolle des Projetsauschusses übernimmt der Fachvorgesetzte Raffaele Zompicchiatti. |
| **Qualität- & Sicherheitsmanager** | Raffaele Zompicchiatti übernimmt als Experte die Rolle des Qualität- & Sicherheitsmanagers. |
| **Projektleitung**  Michael Schmid | Die Projektleitung wird durch Michael Schmid besetzt. |
| **Projektrealisation**  Michael Schmid | Der Realisator des Projekts ist Michael Schmid |
| **Projektdokumentation**  Michael Schmid | Die Projektdokumentation wird von Michael Schmid erstellt. |
| **Testing**  David Gloor | David Gloor ist verantwortlich für das Testing. |

Tabelle 3: Projektrollen

# Risikoanalyse

Die Risikoanalyse verschafft einen Überblick von möglichen Risiken und deren Auswirkungen auf das Projekt ohne und mit Massnahmen.   
Bei jedem Risiko ist zusätzlich die Eintrittswahrscheinlichkeit, wie auch das Schadensausmass ersichtlich.  
Durch Massnahmen kann die Eintrittswahrscheinlichkeit von Problemen reduziert werden wie auch das Schadensausmass.

## Legende

### Schadensausmass

|  |  |
| --- | --- |
| **Labeling** | **Beschrieb** |
| S1 | keine Abwertung |
| S2 | Abwertung bis zu 1.0 Notenpunkte |
| S3 | Abwertung über 1.0 Notenpunkte |
| S4 | führt zu Nichtbestehen |

Tabelle 4: Schadensausmass

### Eintrittswahrscheinlichkeit

|  |  |
| --- | --- |
| **Labeling** | **Beschrieb** |
| E1 | unvorstellbar |
| E2 | unwahrscheinlich |
| E3 | wahrscheinlich |
| E4 | sehr wahrscheinlich |

Tabelle 5: Eintrittswahrscheinlichkei

## Risikograph

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nr.** | **Risikobeschreib** | **Auswirkung** | **Schadensausmass (vor Massnahme)** | **Eintrittswahrscheinlichkeit (vor Massnahme)** | **Massnahmen** | **Schadensausmass (nach Massnahme)** | **Eintrittswahrscheinlichkeit (nach Massnahme)** |
| R1 | Krankheit/Unfall und andere nicht geplante Ausfälle | Das Projekt kann nicht fristegerecht abgegeben werden. | S4 | E2 | Der Experte wird sofort kontaktiert und es wird ein Arztzeugnis eingeholt. | S1 | E2 |
| R2 | Zeit reicht nicht aus | Der Abgabetermin wird nicht eingehalten. | S3 | E3 | Es wurde ein Zeitplan erstellt, der streng befolgt werden muss. | S2 | E2 |
| R3 | Netzwerkausfall | Der Kandidat kann nicht aufs Internet zugreifen. | S4 | E2 | Es wird auf ein alternatives Netzwerk umgestellt, der verantwortliche Experte wird bei Bedarf informiert | S2 | E2 |
| R4 | Höhere Gewalt | Schaden an Informatikinfrastruktur. | S3 | E2 | Es wird auf die neuste Sicherung zugegriffen. Wenn nötig, wird der Experte kontaktiert. | S2 | E2 |
| R5 | Azure Guthaben reich nicht aus | Die Applikation kann nicht auf Azure betrieben werden. | S4 | E3 | Die Infrastruktur auf Azure läuft nur wenn der Kandidat sie benötigt. | S1 | E2 |
| R6 | Nicht vorgesehene Probleme bei der Realisierung | Verzögerung des Abschlusses. | S3 | E3 | Das Problem wird so schnell wie möglich gelöst, falls nötig wird externe Hilfe in Anspruch genommen. | S2 | E3 |
| R7 | Ausfall Online-Dienste von Drittanbietern | Die Arbeit kann unter Umständen nicht fortgeführt werden. | S3 | E2 | Bei Bedarf wird der Experte kontaktiert und das weitere Vorgehen besprochen. | S2 | E2 |

Tabelle 6: Risikograph

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Vor Massnahme** | | | | |  |  | **Nach Massnahme** | | | | |
| **Eintrittswahrscheinlichkeit** | W4 |  |  |  |  |  | **Eintrittswahrschein** | W4 |  |  |  |  |
| W3 |  |  | R2, R5, R6 |  |  | W3 |  | R6 |  |  |
| W2 |  |  | R4, R7 | R1, R3 |  | W2 | R1 | R2, R3, R4, R5, R7 |  |  |
| W1 |  |  |  |  |  | W1 |  |  |  |  |
|  | | S1 | S2 | S3 |  |  |  | | S1 | S2 | S3 | S4 |
| **Schadensausmass** | | | |  | **Schadensausmass** | | | |

Tabelle 7: Risikomatrix

## Risikomatrix

# Zeitplan



# Arbeitsjournale

## Montag, 15. November 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tätigkeit | | Beteiligte Personen | Aufwand geplant | Aufwand effektiv |
| **Zeitplan erstellen**  Als erstes habe ich den Zeitplan erstellt. Er hilft mir, einen Überblick über das Projekt zu halten und dient mir als Leitfaden. | | M. Schmid | 2 | 2 |
| **Ausgangsituation verfassen**  Die Ausganssituation habe ich aus der Aufgabenstellung übernehmen können. | | M. Schmid | 0.25 | 0.25 |
| **Standards deklarieren**  Gewisse von der Technischen Fachschule Bern vorgegebene Standards habe ich übernommen, zudem habe ich eigene hinzugefügt. | | M. Schmid | 0.25 | 0.25 |
| **Dokumentation überarbeiten**  Ich habe die einzelnen Punkte fortlaufend dokumentiert. | | M. Schmid | 1 | 1 |
| **Arbeitsjournal schreiben.**  Am Ende des Tages habe ich das Arbeitsjournal des ersten Tages verfasst. | | M. Schmid | 0.5 | 0.5 |
| **Total** | | | **4.0** | **4.0** |
| Hilfestellungen | | | | |
| Für den Zeitplan habe ich auf die Vorlage von Edson zurückgegriffen und diese überarbeitet. Sie befindet sich im Anhang. Auch für die Arbeitsjournale greife ich auf eine Vorlage der Technischen Fachschule Bern zurück, die ich jedoch stark angepasst habe. | | | | |
| Positives | Negatives | | | |
| * Gut im Zeitplan * Konzentration * Motivation | * Wenig Vorkenntnisse im Dokumentieren * Kleine Formatierungsfehler im Zeitplan | | | |
| Reflexion | | | | |
| Der erste Tag verlief sehr zügig. Ich konnte mich an den Zeitplan halten. Obwohl ich etwas nervös war, konnte ich mich gut konzentrieren und habe effizient gearbeitet. Leider habe ich erst nach der Abgabe des Zeitplans bemerkt, dass es im Zeitplan minimale Formatierungsfehler hat. Diese werde ich noch korrigieren. | | | | |
| Weiteres Vorgehen | | | | |
| * ISDS erstellen * Organisation der IPA-Ergebnisse erstellen * Projektvorgehen dokumentieren * Projektorganisation dokumentieren | | | | |

Tabelle 8: Tagesjournal 15.11.21

## Dienstag, 16. November 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tätigkeit | | Beteiligte Personen | Aufwand geplant | Aufwand effektiv |
| **ISDS definieren**  Als erste Tätigkeit habe ich das ISDS Kapitel erstellt. Innerhalb dieses Kapitels befindet sich die Vor-IPA Schutzbedarfsanalyse. | | M. Schmid, | 0.5 | 1 |
| **Organisation der Vor-IPA Ergebnisse**  Als nächstes habe ich den Abschnitt «Organisation der Vor-IPA Ergebnisse» hinzugefügt. Dies beinhaltete auch Unterkapitel wie Arbeitsplatz, Datensicherung, Wiederherstellung der Daten und Versionierung. | | M. Schmid, | 1 | 1 |
| **Projektvorgehen dokumentieren**  Das Projektvorgehen wurde dokumentiert und die wichtigsten Elemente wurden grob erklärt | | M. Schmid | 0.5 | 0.25 |
| **Projektorganisation verfassen**  Danach habe ich die Projektorganisation erfasst. Die Projektorganisation ist nun sowohl als Organigramm als auch als Tabelle abgezeichnet. | | M. Schmid | 0.5 | 0.25 |
| **Dokumentation überarbeiten**  Ich habe die einzelnen Punkte fortlaufend dokumentiert. | | M. Schmid, | 1 | 1 |
| **Arbeitsjournal schreiben**  Zum zweiten Mal das Arbeitsjournal schreiben, endlich mal etwas repetitive Arbeit! | | M. Schmid | 0.5 | 0.5 |
| **Total** | | | **4.0** | **4.0** |
| Hilfestellungen | | | | |
| Für die Schutzbedarfsanalyse innerhalb des ISDS habe ich eine Vorlage verwendet von unserem Dashboard-Team. Die Vorlage befindet sich im Anhang. | | | | |
| Positives | Negatives | | | |
| * Projektvorgehen und Projektorganisation schneller erstellt als gedacht. * Nach wie vor im Zeitplan | * Mehr Zeit fürs ISDS als gedacht * Ablenkung | | | |
| Reflexion | | | | |
| Den Projektstart habe ich hinter mir, es ist bereits der zweite Projekttag. Glücklicherweise bin ich immer noch im Zeitplan. Ich merkte, dass ich etwas abgelenkt vom Lärm der anderen Klasse war. Unglücklicherweise hatte ich vergessen, meine Kopfhörer mitzunehmen; ich konnte mich also nicht so gut abschirmen. Insgesamt bin ich aber zufrieden mit dem heutigen Forschritt. | | | | |
| Weiteres Vorgehen | | | | |
| * Technische Risikoanalyse * Phasenfreigabe Initialisierung * Ist/Soll analysieren * Abgrenzungskriterien dokumentieren | | | | |

Tabelle 9: Tagesjournal 16.11.21

## Mittwoch, 17. November 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tätigkeit | | Beteiligte Personen | Aufwand geplant | Aufwand effektiv |
| **Technische Risikoanalyse**  Als erste Tätigkeit des Tages habe ich die Technische Risikoanalyse erstellt. Insgesamt habe ich 7 Risiken definiert. | | M. Schmid | 1 | 1 |
| **Phasenfreigabe: Phase Initialisierung**  Da in der Projektmethode Hermes 5.1 die Phase Initialisierung die erste Phase ist braucht es eigentlich keine Phasenfreigabe; In der Vor-IPA ist dies anders, da es noch einen formalen Teil gibt, welcher abgenommen werden muss, ehe die Phase Initialisierung gestartet werden kann. Cuno Vollenweider hat mir die Phasenfreigabe mündlich erteilt. Er hat gewünscht, dass ich den vom Detailbeschreib übernommene in gewissen Punkten konkretisiere. | | M. Schmid | 0.25 | 0.25 |
| **Ist/Soll Studie**  Die erste Tätigkeit in der Phase Initialisierung war die Ist/Soll Studie. Mit der Ist Situation bin ich als Lernender vertraut. Die Soll Situation konnte ich aus der Aufgabenstellung ziehen. | | M. Schmid | 0.5 | 0.5 |
| **Ziele definieren**  Als nächstes habe ich mich an das Definieren von Zielen gemacht. Die Ziele habe ich unterteilt in persönliche Ziele und Projektziele. | | M. Schmid | 0.5 | 0.75 |
| **Dokumentation überarbeiten**  Ich habe die einzelnen Punkte fortlaufend dokumentiert. | | M. Schmid | 1 | 1 |
| **Arbeitsjournal schreiben**  Am Schluss des Tages habe ich das lang ersehnte AJ geschrieben, die einzige repetitive Arbeit des Tages. | | M. Schmid | 0.5 | 0.5 |
| **Total** | | | **3.75** | **4** |
| Hilfestellungen | | | | |
| Hermes Website  Cuno Vollenweider | | | | |
| Positives | Negatives | | | |
| * Phasenfreigabe wurde erteilt * Fokussiert gearbeitet * Motivation | * Ich bin mir teilweise unsicher in gewissen Punkten im Zusammenhang mit Hermes und muss oft deren Website aufschlagen. | | | |
| Reflexion | | | | |
| Allmählich komme ich in den Arbeitsfluss der Vor-IPA. Es fällt mir weniger schwer mich zu fokussieren als am Anfang des Projekts, was sehr positiv ist. Ebenfalls war ich auch froh, dass Cuno Vollenweider weitgehend zufrieden mit dem Formalen Teil war und mir schliesslich die Phasenfreigabe erteilt hat. Leider bin ich minimal im Verzug, denn ich hätte heute bereits die Abgrenzungskriterien dokumentieren sollen. Weil ich diese jedoch direkt aus der Aufgabenstellung übernehmen kann, ist das nicht so schlimm. | | | | |
| Weiteres Vorgehen | | | | |
| * Abgrenzungskriterien dokumentieren * Anforderungen definieren * Schnittstellen und Umfeld dokumentieren * Rahmenbedingungen dokumentieren * Phasenfreigabe: Phase Konzept * Systemarchitektur entwerfen * Anwendungsfälle definieren | | | | |

Tabelle 10: Tagesjournal 17.11.21

## Freitag, 19. November 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tätigkeit | | Beteiligte Personen | Aufwand geplant | Aufwand effektiv |
| **Abgrenzungskriterien dokumentieren**  Nun sind alle wichtigen Abgrenzungskriterien in der Dokumentation festgehalten. Ich habe sie aus der Aufgabenstellung übernommen. | | M. Schmid | 0.5 | 0.5 |
| **Anforderungen definieren**  Als nächstes habe ich die Anforderungen dokumentiert, welche in der Aufgabenstellung ersichtlich sind. | | M. Schmid | 0.5 | 0.5 |
| **Schnittstellen und Umfeld analysieren**  Die Schnittstellen und das Umfeld, in welchem meine Applikation läuft sind nun dokumentiert. Ich habe eine Grafik eingefügt, welche die Beschreibung bildlich veranschaulicht. | | M. Schmid | 1 | 1 |
| **Rahmenbedingungen definieren**  Die letzte Aufgabe in der Phase Initialisierung, war das Erfassen von Rahmenbedingen. | | M. Schmid | 0.5 | 0.5 |
| **Phasenfreigabe**  Bevor ich mit der Konzeptphase loslegen konnte, hat Matthias Heimberg die Initialisierungsphase abgenommen und mir die Phasenfreigabe für die Phase Konzept erteilt. | | M. Schmid | 0.25 | 0.25 |
| **Systemarchitektur dokumentieren**  Ich war nun in der Phase Konzept, wo ich als erstes die Systemarchitektur entworfen und anschliessend in die Dokumentation eingefügt habe. | | M. Schmid | 1.5 | 2.5 |
| **Anwendungsfälle erstellen**  Als letzte Phasenbasierte Aufgabe habe ich die Anwendungsfälle erstellt. Verwendet habe ich dabei das Programm Draw.io | | M. Schmid | 1 | 1 |
| **Dokumentation überarbeiten**  Ich habe die einzelnen Punkte fortlaufend dokumentiert. | | M. Schmid | 1 | 1 |
| **Arbeitsjournal schreiben**  Am Schluss des Tages habe ich wie immer das Tagesjournal geschrieben. | | M. Schmid | 0.5 | 0.5 |
| **Aufräumen und Putzen** Wie jeden Freitag, haben wir eine Viertelstunde vor Schluss noch die Werkstatt geputzt. | | Inf19 | 0.25 | 0.25 |
| **Total** | | | **7.0** | **8.0** |
| Hilfestellungen | | | | |
| Matthias Heimberg | | | | |
| Positives | Negatives | | | |
| * Motivation * Keine grossen Pannen * Fast im Zeitplan | * Konzentration schwindet im Verlauf des Nachmittags | | | |
| Reflexion | | | | |
| Heute war ein eher anstrengender und langer Tag, weil wir den ganzen Tag an der vIPA gearbeitet hatte. Am Morgen war ich noch produktiv aber am Nachmittag merkte ich, wie ich den Fokus langsam am Verlieren war. Das Projekt befindet sich nun endlich in der Konzeptphase, das heisst: Die Realisierungsphase ist nicht mehr fern. | | | | |
| Weiteres Vorgehen | | | | |
| * Datenbankkonzept erarbeiten * Testkonzept * Phasenfreigabe Phase Realisierung | | | | |

Tabelle 11: Tagesjournal 19.11.21

## Montag, 22. Juni 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tätigkeit | | Beteiligte Personen | Aufwand geplant | Aufwand effektiv |
| **Datenbankkonzept erarbeiten**  Nun sind alle wichtigen Abgrenzungskriterien in der Dokumentation festgehalten. Ich habe sie aus der Aufgabenstellung übernommen. | | M. Schmid | 1 | 1 |
| **Testkonzept erstellen**  Als nächstes habe ich die Anforderungen dokumentiert, welche in der Aufgabenstellung ersichtlich sind. | | M. Schmid | 1.5 | 1.25 |
| **Besprechung Phasenfreigabe mit Expert**  Die Schnittstellen und das Umfeld, in welchem meine Applikation läuft sind nun dokumentiert. Ich habe eine Grafik eingefügt, welche die Beschreibung bildlich veranschaulicht. | | M. Schmid | 0.25 | 0.25 |
| **Dokumentation überarbeiten**  Ich habe die einzelnen Punkte fortlaufend dokumentiert. | | M. Schmid | 1 | 1 |
| **Arbeitsjournal schreiben**  Am Schluss des Tages habe ich wie immer das Tagesjournal geschrieben. | | M. Schmid | 0.5 | 0.5 |
| **Total** | | | **4.25** | **4** |
| Hilfestellungen | | | | |
| Raffaele Zompicchiatti | | | | |
| Positives | Negatives | | | |
| * Motivation * Gute Konzentration | * Phasenfreigabe wurde nicht erteilt. | | | |
| Reflexion | | | | |
| Heute wollte ich eigentlich die Konzeptphase abschliessen, jedoch ist meine Systemarchitektur laut Experte zu mager. Morgen werde ich die Systemarchitektur nochmals überarbeiten. | | | | |
| Weiteres Vorgehen | | | | |
| * Systemarchitektur überarbeiten * Anwendungsfälle überarbeiten | | | | |

Tabelle 11: Tagesjournal 19.11.21

## Dienstag, 23. Juni 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tätigkeit | | Beteiligte Personen | Aufwand geplant | Aufwand effektiv |
| **Systemarchitektur verändert**  Die Systemarchitektur wurde dahingehend verändert, dass nicht nur die Kontextsicht ersichtlich ist, sondern die einzelnen Systembausteine. | | M. Schmid | 1.5 | 2 |
| **Anwendungsfälle**  Als nächstes wurden die Anwendungsfälle überarbeitet. Ich habe den Anwendungsfall Unit Test hinzugefügt. | | M. Schmid | 0.75 | 0.5 |
| **Dokumentation überarbeiten**  Ich habe die einzelnen Punkte fortlaufend dokumentiert. | | M. Schmid | 1 | 1 |
| **Arbeitsjournal schreiben**  Am Schluss des Tages habe ich wie immer das Tagesjournal geschrieben. | | M. Schmid | 0.5 | 0.5 |
| **Total** | | | **3.75** | **4.0** |
| Hilfestellungen | | | | |
| Heute habe ich keine Hilfestellungen verwendet. | | | | |
| Positives | Negatives | | | |
| * Motivation * Keine grossen Pannen * Fast im Zeitplan |  | | | |
| Reflexion | | | | |
| Ich finde, dass mein Konzept nun besser ist, weil ich nun tief ins Detail gegangen bin. Ich habe die Umgebung in Azure aufgezeigt und auch die Softwarearchitektur erstellt. | | | | |
| Weiteres Vorgehen | | | | |
| * Phasenfreigabe Realisierung * Realisierung starten | | | | |

Tabelle 11: Tagesjournal 19.11.21

# Abschlussberichte (Beispiele)

Das Projekt ist nun vollendet. Ich empfand die letzten 6 Tage als ziemlich streng, dafür habe ich vieles im Zusammenhang mit Projektdokumentation gelernt. Demnächst werde ich mit Giulio das Projekt auswerten, ich bin gespannt auf seine Rückmeldung.

## Schlussbericht Ist/Soll

Das Projekt wurde wie geplant umgesetzt. Zuerst wurde ein Konzept erstellt, danach wurde die Realisierung vorgenommen. Die in der Aufgabenstellung ersichtlichen Ziele wurden alle erreicht. Es kann auf die gewünschten Einträge in der Datenbank zugegriffen werden und diese werden dann auf der Schulhomepage angezeigt.

## Erkenntnisse

### Anforderungen

Die Anforderungen des Projekts wurden, soweit ich es beurteilen kann, erfüllt.  
Die Testfälle haben keine Fehler verursacht.

### Zeit

Ich hatte fast durchgehend Zeitdruck. Eine komplette IPA Dokumentation ohne Vorlage innert 6 Tagen bzw. Halbtagen nachzuahmen war eine grosse Herausforderung für mich.  
Ich hoffe, dass diese Dokumentation einige Elemente enthält, auf welche ich in weiteren Projekten zurückgreifen kann.

### Einsatzmittel

Ich habe lediglich die Einsatzmittel verwendet, welche ich auf Seite 6 aufgeführt habe.  
Es wurden keine weiteren Mittel verwendet.

### Qualität der Arbeit

Die Qualität der Dokumentation hätte ich gern ein bisschen besser gehabt, leider reicht es mir jedoch zeitlich nicht mehr. Mit dem Produkt an sich bin ich ganz zufrieden, wenn man möchte, könnte man die Darstellung noch ein bisschen verbessern, aber die Funktionalität ist gewährleistet.

## Persönliches Fazit / Schlussreflexion

Generell bin ich zufrieden mit diesem Projekt. Ich hoffe, dass dies Giulio auch so sieht. Ich denke, dass dies eine gute Gelegenheit war, um Projektdokumentation zu üben. Fast die ganze Zeit stand ich unter Zeitdruck, dies ist wohl normal bei einem solchen Projekt.

Mir ist jetzt auf jeden Fall bewusst geworden, das man mit Hermes, wenn man es sauber ausführt, ein Projekt effizient abwickeln kann und Fehler minimieren kann. Ich denke, dass ich mit Hermes auch bei meiner IPA arbeiten werde, da ich mich langsam mit dieser Projektmethode angefreundet habe.  
  
Während des Projekts musste ich mich in viele mir neue Bereiche einarbeiten, wie z.B. in das Erstellen von Konzepten. Ich hoffe, dass meine Konzepte nicht für die Tonne sind und bin gespannt auf die Meinung von Giulio.

# Teil 2: Projektdokumentation





Ein Bild, das Text, ClipArt enthält.

Automatisch generierte BeschreibungEin Bild, das ClipArt enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

# Initialisierung

## Beschreibung Ist-Zustand / Soll-Zustand

In diesem Abschnitt wird erläutert was der Ist-Zustand ist (Zustand bevor das Projekt abgeschlossen ist), anschliessend wird der Soll-Zustand erläutert (Gewünschter Nachdem

### Istzustand

Zurzeit wird in der Technischen Fachschule Bern keine cloudbasierte Lösung für laufende Projekte in der Werkstatt eingesetzt. Meistens werden Projekte lokal auf den Laptops der Lernenden entwickelt. Dies ist umständlich und erschwert die Zusammenarbeit, weil die Applikation nur lokal auf einem Laptop läuft. Zudem kann die Applikation nur dann ausgeführt werden, wenn auch der Laptop des Lernenden eingeschaltet ist und alle nötigen Dienste am Laufen sind.

### Sollzustand

In Zukunft sollen cloudbasierte Lösungen für laufende Projekte in der Werkstatt verwendet werden. Dies hat viele Vorteile für laufende Projekte, insbesondere dann wenn mehrere Personen im Projekt involviert sind. Nehmen wir das Beispiel des Projekts «Dashboard»:  
Sämtliche Micro-services können auf der Cloud abgelegt werden und sind bei Bedarf permanent und standortunabhängig verfügbar und wartbar (Portabilität und Wartbarkeit). Die Services können innert sehr kurzer Zeit gestartet und gestoppt werden.

## Ziele

In diesem Kapitel wird beschrieben nach welchen Kriterien die Ziele innerhalb dieser Vor-IPA definiert werden und was die jeweiligen Ziele sind.

### Smart-Ziele

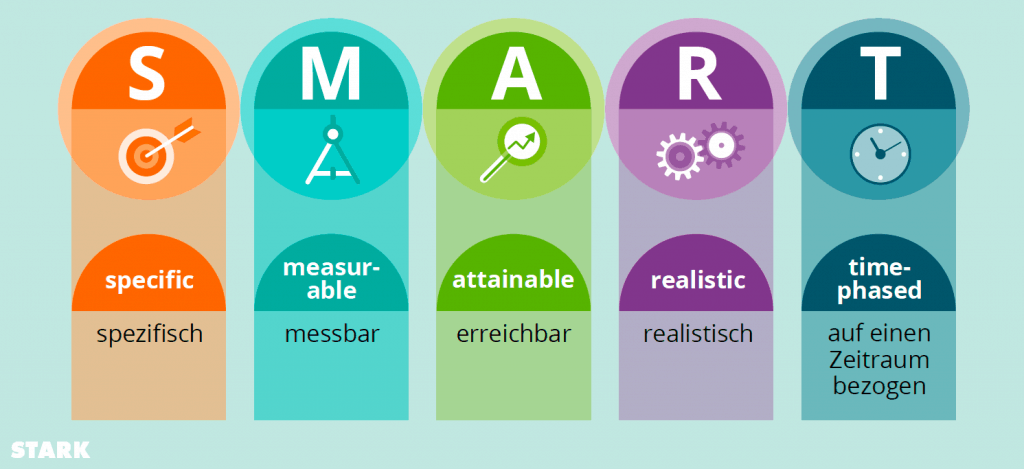
Um klare Ziele zu definieren, wird innerhalb dieses Projekts die SMART-Methode verwendet. Das Wort «SMART» setzt sich zusammen aus den Anfangsbuchstaben der englischen Wörter Specific, Measurable, Attainable, Realistic, Time-phased

Abbildung 13: Smartziele

Quelle: https://www.schultrainer.de/lernen-und-wissen/wirtschaft-politik-und-sozialkunde/die-smart-methode-einfach-erklaert/

### Persönliche Vorgehensziele

|  |  |
| --- | --- |
| **ID** | **Beschreibung** |
| 1 | Die Projektmethode HERMES 5.1 wird während des gesamten Projekts eingesetzt und korrekt angewandt. |
| 2 | Die gesetzten Meilensteine werden zeitlich eingehalten |
| 3 | Jede Version der Dokumentation stimmt mit dem Zeitplan überein |
| 4 | Alle Hilfestellungen und übernommene Elemente sind richtig angegeben |
| 5 | Das gesamte Projekt wird rechtzeitig abgegeben. |

Abbildung 14: Persönliche Vorgehensziele

### Projektziele

|  |  |
| --- | --- |
| **ID** | **Beschreibung** |
| 1 | Lernende des 3. Lehrjahres der Abteilung I sollen anhand der Anleitung des Kandidaten selbständig eine JavaScript Webapplikation als Docker Container in der Cloud verfügbar machen können. |
| 2 | Die Datenbank soll abgesehen von der definierten Schnittstelle abgekapselt sein |
| 3 | Die Webapplikation soll Benutzereingaben in einer Datenbank abspeichern können und diese wieder abrufen können. |
| 4 | Die Webapplikation funktioniert und ist sicher. |
| 5 | Der Code ist verständlich, er braucht keine Kommentare, sodass andere Lernende die Software unabhängig weiterentwickeln können. |

Abbildung 15: Projektziele

## Anforderungen

### Funktionale Anforderungen

* Die im Docker laufende Applikation muss von aussen per REST ansteuerbar sein
* Es soll eine einfache MySQL Datenbank in Azure konfiguriert werden, welche von der Webapplikation angesteuert wird
* Man muss Daten von aussen per REST an die Webapplikation senden können (Post), welche diese in der MySQL Datenbank abspeichert
* Man muss von aussen dieselben Daten, welche zuvor in der Datenbank gespeichert wurden, wieder per REST (Get) abholen können
* Die Webapplikation muss mit Tests getestet sein

### Nicht funktionale Anforderungen

* Die in dieser Arbeit verwendeten Features von Docker Instanzen auf Azure Cloud sind analysiert und dokumentiert
* Der Fokus des Docker Containers liegt auf einer NodeJS/Express JavaScript Web Applikation
* Es soll eine für Lernende des 3. Lehrjahrs der Abteilung I verständliche, Schritt für Schritt Anleitung geschrieben werden, wie eine NodeJS/Express JavaScript Webapplikation als Docker Container in der Cloud verfügbar gemacht werden kann

## Abgrenzungskriterien

* Es reicht, wenn die Webapplikation einen einzigen REST Endpunkt hat, da der Fokus auf der Machbarkeit und nicht auf der eigentlichen Funktion liegt
* Das Bereitstellen des Containers mit der Webapplikation in der Cloud kann wahlweise manuell oder per Skript erfolgen
* Der Aufbau der MySQL Tabellen und Daten ist nicht wichtig und daher frei wählbar. Diese müssen nur von der Webapplikation verwendet werden können
* Das Ansteuern der REST Schnittstellen kann irgendwie erfolgen (z.B. Postman). Es muss nur deren korrekte Funktion gezeigt werden können.

## Schnittstellen und Umfeld

Sämtliche Infrastruktur für die Webapplikation ist auf der Azure-Cloud.  
Die im Docker laufende Applikation ist ebenfalls in der Cloud angesiedelt und von aussen per REST ansteuerbar. Alle verwendeten Datenbanken sind hinter der Azure-Firewall versteckt und sind nur von einem Webservice, der die einzige Schnittstelle zur Aussenwelt darstellt, ansteuerbar.

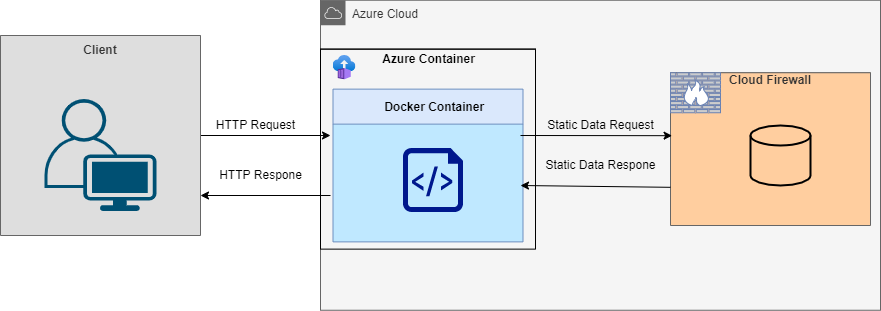


Abbildung 16: Schnittstellen und Umfeld

## Rahmenbedingungen

Die Rahmenbedingungen beschreiben, unter welchen Konditionen das Projekt abgewickelt wird. Dabei wird unterschieden zwischen technischen und organisatorischen Rahmenbedingungen.

### Technische Rahmenbedingungen

* Die Webapplikation wird in JavaScript geschrieben
* Das Datenbankverwaltungssystem ist MySQL
* Die Webapplikation läuft in einem Dockercontainer
* Auf die Dokumentation und andere projektrelevante Dokumente kann von anderen Geräten als dem Arbeitsnotebook des Kandidaten zugegriffen werden.
* Die Infrastruktur für die Webapplikation wird von Azure bereitgestellt

### Organisatorische Rahmenbedingungen

* Die zur Verfügung stehende Zeit während der Vor-IPA beträgt 40 Stunden, wobei jeweils an 6 Halbtagen und 2 vollen Arbeitstagen an der Vor-IPA gearbeitet wird.
* Das Resultat wird durch Eigenleistung des Kandidaten erzielt, Hilfestellungen müssen korrekt angegeben werden.
* Jede Phase muss von einem Experten freigegeben werden
* Die Projektmethode ist Hermes 5.1
* Die Dokumentation muss bis spätestens 26.11.2021 17.00 abgegeben werden.

# Konzept

## Erläuterung der verwendeten Mittel und Methoden

|  |  |
| --- | --- |
| **Bezeichnung** | **Beschreibung** |
| Azure | Azure (Microsoft Azure) ist eine Cloud-Computing-Plattform. Die Konzepte Infrastructure as a Service, Platform as a Service und Software as a Service können mit Azure realisiert werden. |
| Docker | Docker basiert auf dem Konzept «Platform as a Service» wobei Software in sogenannte Container isoliert werden können. |
| MySQL Datenbank | MySQL ist ein serverbasiertes relationales Datenbankmanagementsystem. |
| JavaScript | JavaScript ist eine Programmiersprache, welche mit Hilfe von Frameworks unter anderem auch als Backend-Programmiersprache verwendet wird. |

Tabelle 12: Mittel und Methoden Erläuterung

## Systemarchitektur

Abbildung 17: Systemarchitektur Kontextsicht

### Konzept Client

Der Client befindet sich ausserhalb der Azure-Infrastruktur. Er kann mithilfe von http Requests die Webappliaktion ansteuern. Da die Webapplikation nur aus dem Backend besteht, wird für das Ansteuern Insomnia oder Postman verwendet.

Detaillierte Anforderungen

|  |  |
| --- | --- |
| **ID** | **Beschreibung** |
| 1 | Daten müssen von aussen per REST (POST) gesendet werden |
| 2 | Daten müssen von aussen per REST (GET) geholt werden. |

Tabelle 13: Detaillierte Anforderungen

Voraussetzungen an den Client(benutzer)

|  |  |
| --- | --- |
| **ID** | **Beschreibung** |
| 1 | Insomnia oder Postman installiert |
| 2 | Kennt Endpunkte |
| 3 | Kennt Requestsyntax |
| 4 | Hat Verbindung zum Internet |

Tabelle 14: Vourraussetzungen an den Client

Requestsyntax

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Endpunkt** | **Methode** | **Datenformat** | **Body Inhalt** |
| 1 | / | GET | Kein Body | - |
| 2 | /api/messages/ | POST | JSON | {"messsage": "sample message"} |
| 3 | /api/messages/(:ID) | GET | Kein Body | - |

Tabelle 15: Requestsyntax

### Konzept Azure Cloud

Das gesamte Backend samt DB ist auf der Azure Cloud angesiedelt. Um Die Webapplikation auszuführen, werden verschiedene Dienste und Ressourcen verwendet. Für die VorIPA wurde eine Ressourcengruppe erstellt, worin alle Systembestandteile abgelegt sind.

Verwendete Dienste und Ressourcen

Nachfolgende Tabelle dient als Übersicht aller verwendeter Dienste und Ressourcen. Ausserdem wird jeder Dienst bzw. jede Ressource kurz beschrieben.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Beschreibung** | **Azure Icon** |
| 1 | Ressourcengruppen | Eine Ressourcengruppe ist ein Container, der zugehörige Ressourcen für eine Azure-Lösung enthält. |  |
| 2 | Azure Database for MySQL | Hier können SQL-Datenbanken erstellt, verwaltet und betrieben werden. |  |
| 3 | Containerregistrierung | Hier können Container registriert werden |  |
| 4 | Containerinstanzen | Hier können Containerinstanzen erstellt, verwaltet und betrieben werden. |  |
| 5 | Firewall Manager | Hier kann die Firewall verwaltet werden und es können unter anderem Virtuelle Netzwerke erstellt werden. |  |

Tabelle 16: Verwendete Azure Dienste und Ressourcen

Ressourcengruppen

In nachfolgender Tabelle ist beschrieben, wie die Ressourcengruppe konfiguriert ist.

|  |  |
| --- | --- |
| **Bezeichnung** | **Beschreibung** |
| Icon |  |
| Name | VIPA |
| Region | Schweiz |

Tabelle 17: Ressourcengruppen

SQL-Datenbank

Hier wird beschrieben in welchem Umfeld die Datenbank erstellt werden soll.

MySQL Server

In nachfolgender Tabelle ist beschrieben, wie der MySQL-Server konfiguriert ist.

|  |  |
| --- | --- |
| **Bezeichnung** | **Beschreibung** |
| Icon |  |
| Ressourcengruppe | VIPA |
| Servername | VIPASRV |
| Standort | Schweiz |
| Authentifizierungsmethode | SQL-Authentifizierung |
| Serveradministratoranmeldung | scmi |
| Kennwort | *geheim* |

Tabelle 18: MySQL Server

Datenbank

In nachfolgender Tabelle ist beschrieben, wie die Datenbank konfiguriert ist.

|  |  |
| --- | --- |
| **Bezeichnung** | **Beschreibung** |
| Icon |  |
| Ressourcengruppe | VIPA |
| Datenbankname | vIPAWebAppDB |
| Server | VIPASRV |
| Datenquelle | Keine |
| Zugriff | Andere Azure-Dienste und Ressourcen / Privater Endpunkt |

Tabelle 19: Datenbank

Firewall Manger

In nachfolgender Tabelle ist beschrieben, wie die Datenbank konfiguriert ist

|  |  |
| --- | --- |
| **Bezeichnung** | **Beschreibung** |
| Icon |  |
| Ressourcengruppe | VIPA |
| Name | VIPAFW |
| Name des Virtuellen Netzwerks | VIPAVNW |
| Subnetzadressraum | 10.10.10.0/26 |
| Zugriff gewähren von | Ressourcengruppe VorIPA |

Tabelle 20: Firewall Manager

Containerregistrierung

|  |  |
| --- | --- |
| **Bezeichnung** | **Beschreibung** |
| Icon |  |
| Ressourcengruppe | VIPA |
| Registrierungsname | dockerWebapp |
| Region | Schweiz |
| DNS-Namensbezeichnung | vipawebapp |

Tabelle 21: Containerregistrierung

Containerinstanzen

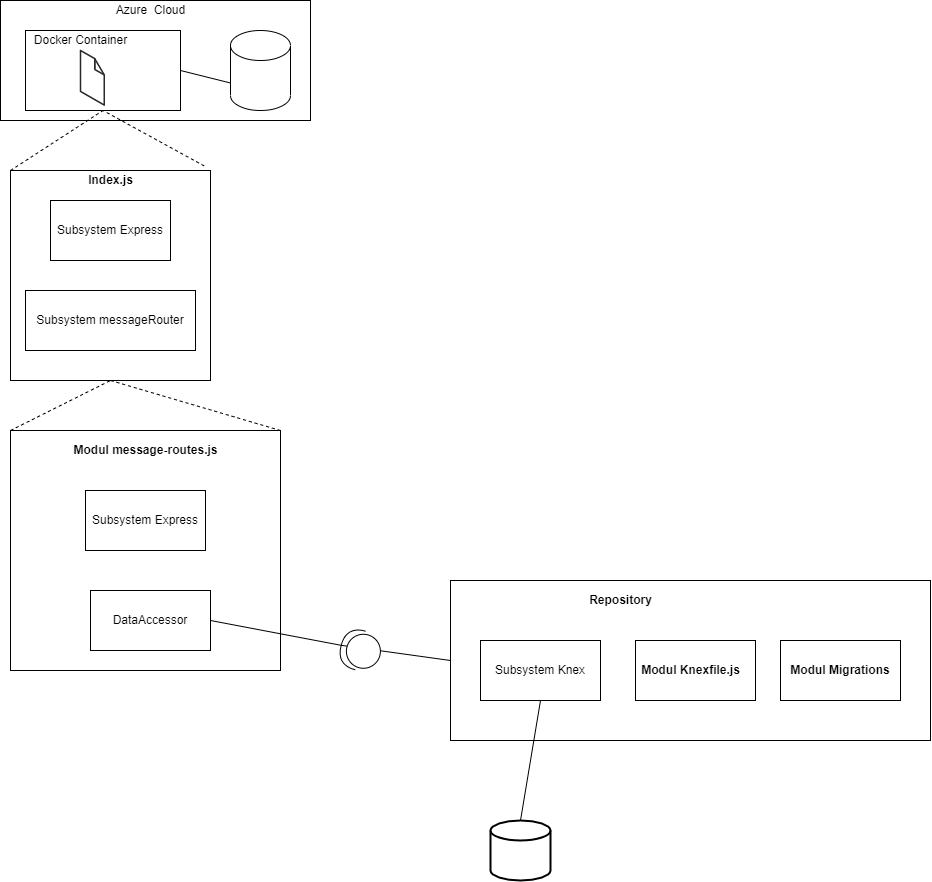
|  |  |
| --- | --- |
| **Bezeichnung** | **Beschreibung** |
| Icon |  |
| Ressourcengruppe | VIPA |
| Containername | dockerwebapp |
| Region | Schweiz |
| Imagequelle | Docker |
| Imagetyp | Öffentlich |
| Netzwerktyp | Öffentlich |
| DNS-Namensbezeichnung | vipawebapp |

Tabelle 22: Containerinstanzen

### Softwarearchitekture

Die Umgebung der Webapplikation auf Azure ist nun beschrieben, als nächstes wird nun die Webapplikation an sich genauer beschrieben.

Abbildung 18: Softwarearchitektur Bausteinsicht (Whitebox)



Module

|  |  |
| --- | --- |
| **Modul** | **Beschreibung** |
| MessageRoutes | Hier sind die verschiedenen Endpunkte der Route MessagesRoutes definiert. |
| Knexfile | Hier werden die Verbindungsdaten zur Datenbank deklariert. |
| Migrations | Hier wird die Tabellensyntax definiert. |

Enpunkte

Aus Gründen der Modularität wird die API mithilfe eines Routers unterteilt. In fall dieser Webapp existiert momentan lediglich eine Route «/api/messages/».

|  |  |  |
| --- | --- | --- |
|  | | |
| **http-request-methode** | **Endpunkt** | **Beschreib** |
| GET | / | Eine Nachricht wird ausgegeben im Browser. |

Tabelle 23: Root Endpunkt

|  |  |  |
| --- | --- | --- |
| **/api/messages** | | |
| **http-request-methode** | **Endpunkt** | **Beschreib** |
| POST | / | Im Request-Body wird eine Message mitgegeben und in der Datenbank abgespeichert. |
| GET | /(:uuid) | Gibt alle in der Datenbank gespeicherten Messages aus. Wird eine UUID mitgegeben, so wird nur die Message mit dieser uuid ausgegeben. |

Tabelle 24: Messages Endpunkt

Dies ist lediglich ein Grobkonzept. Änderungen während der Realisierung sind möglich

Dependencies

|  |  |
| --- | --- |
| **Dependency** | **Beschreibung** |
| express | JS-Backend Framework, welche modulare Softwarearchitektur ermöglicht |
| knex | SQL-Abfrage-Generator für Node.js |
| mysql | Node.js Treiber für MySQL |
| Jest | Framework zum Testen von JavaScript-Code |

### Zugriffsberechtigungen

Im folgenden Abschnitt werden die Zugriffberechtigungen für die einzelnen geschützten Elemente kurz beschrieben.

Azure

|  |  |
| --- | --- |
| **Bezeichnung** | **Wert** |
| Vollständiger Name | Michael Schmid |
| E-Mail | In19scmi@tfbern.ch |
| Passwort | - |
| Host Value | Portal.azure/michaelvorIPA |

Tabelle 25: Azure Zugriff

MySQL

|  |  |
| --- | --- |
| **Bezeichnung** | **Wert** |
| Benutzername | michael |
| Passwort | - |
| Host Value | Portal.azure/michaelvorIPA |

Tabelle 26: MySQL Zugriff

### Deploymentkonzept

Abbildung 19: Deployment in Azure Ablaufübersicht

Quelle: https://azuredays.com/2020/11/06/alternatives-to-azure-kubernetes-aks-azure-container-instances/

Deploymentschritte

**Einleitung**

Als erstes wurde die Webapplikation lokal erstellt und getestet. Sobald die Webapplikation lokal einwandfrei läuft, soll diese auf Azure deployed werden. Dazu wird als erstes ein Dockerimage erstellt. Das Dockerimage wird schliesslich auf Azure gepushed und dort kann es als dann als neue Containerinstanz ausgewählt werden.

**Voraussetzungen**

* Webapplikation lokal fertiggestellt
* Docker Image von Webapplikation erstellt
* Azure nach «Azure-Konzept» umgesetzt.

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Beschreibung** | **Befehl** |
| 1 | Auf Azure Ressourcengruppe anmelden | Docker login nodewebapi.azuecr.io |
| 2 | Dockerimage auf Cloudcontainerregistry pushen | Docker push nodewebapi.azuecr.io/node-webapp |
| 3 | Neue Container Instanz erstellen |  |
| 4 | Image Source: Azure Container Registry |  |
| 5 | Gewünschtes Image auswählen |  |

Tabelle 27: Deploymentschritte

## Anwendungsfälle

Die Webapplikation hat lediglich 3 Anwendungsfälle. In untenstehender Grafik sind die Anwendungsfälle dargestellt.

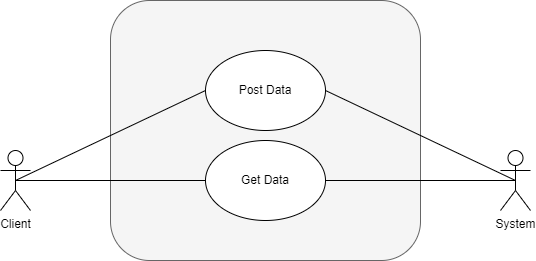


Abbildung 20: Anwendungsfälle

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Anwendungsfall** | **Client** | **System** |
| 1 | Post Data | Der Client sendet in JSON Format Daten an die WebApp | Das System versucht die Daten in der Datenbank abzuspeichern. |
| 2 | Get Data | Der Client fordert von der Webapp Daten | Das System sucht die geforderten Daten in der Datenbank und gibt sie dem Client zurück |

Tabelle 28: Anwendungsfälle

## Datenbankkonzept / DV-Konzept

|  |  |
| --- | --- |
| **Titel** | **Beschreibung** |
| Datenbankverwaltungssystem | MySQL |
| Standort der DB | Auf Azure hinter einer Firewall |
| Anzahl Datenbanken | 1 |
| Anzahl Tabellen | 1 |

Tabelle 29: Datenbank Spezifikationen

### ERM

Das ERM ist sehr simpel. Es existiert lediglich eine einzige Tabelle mit zwei Attributen, wobei das Attribut ID zur eindeutigen Identifizierung dient.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 21: ERM

## Testkonzept

## Testart

|  |  |  |
| --- | --- | --- |
| **ID** | **Testart** | **Beschreibung** |
| 1 | Integrationstests | Integrationstests werden verwendet um zu testen, ob die verschiedenen Elemente i Zusammenspiel funktionieren. |
| 2 | Unit Tests | Unit Tests werden verwendet um einzelne Komponenten eines Computerprogramms zu testen. Anders als bei Integrationstests, darf diese Unit keine Abhängigkeiten besitzen. |

Tabelle 30: Testart

### Testziele

|  |  |  |
| --- | --- | --- |
| **Testziel ID** | **Zielbeschreib** | **MK ID (Ableitung)** |
| 1 | Die Website ist von aussen aufrufbar. | 3 |
| 2 | Es können Daten postiert werden. | 6 |
| 3 | Es können Daten geholt werden. | 7 |
| 4 | Die Webapplikation ist mit der Datenbank verbunden. | 5 |

Tabelle 30: Testziele

### Testobjekte

In nachfolgender Tabelle sind meine Testobjekte aufgelistet. Im Falle dieses Projekts stellen die Testobjekte Module der Software dar.

|  |  |
| --- | --- |
| **Testobjekt ID** | **Testobjekt** |
| 1 | Express Webapp |
| 2 | MySQL Datenbank |

Tabelle 31: Testobjekte

### Testarten

|  |  |
| --- | --- |
| **Testart** | **Beschreibung** |
| Integrationstest | Mehrere Komponenten werden getestet |

Tabelle 32: Testarten

### Testabdeckung

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Testziel** | **Testobjekt ID** | **Testart ID** |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 1 | 2 |
| 3 | 3 | 1 | 2 |
| 4 | 4 | 1,2 | 1 |

Tabelle 33: Testabdeckung

### Testvoraussetzungen

Um die Tests durchführen zu können, muss die Realisation abgeschlossen worden sein.  
Des Weiteren braucht es eine Testperson, welche Vorkenntnisse mit der Funktionsweise des Produkts hat.

### Mängeklassifizierung

|  |  |  |
| --- | --- | --- |
| **Klassen ID** | **Fehlerklassen** | **Beschreibung** |
| M0 | Kein Mangel | Das erwartete sowie das Tatsächliche Resultat stimmen überein. |
| M1 | Leichter Mangel | Grundlegend funktioniert alles, kleinere Mängel ersichtlich |
| M2 | Schwerer Mangel | Die Anforderung wird nur teilweise erfüllt. |
| M3 | Kritischer Mangel | Die Anforderungen werden sehr schlecht bis gar nicht erfüllt. |

Tabelle 34: Mängelklassen

### Testumgebung

Die Webapplikation läuft in Azure. Getestet wird vom lokalen Host aus via Insomnia oder Postman.

### Testinfrastruktur

**Testsystem**

* Datenbank Webapp

**Testdaten**

* Individuelle Eingabe vom Benutzer als Message

**Testhilfsmittel**

* Jest.js
* Insomnia
* Postman

### Testfall Vorlage

|  |  |
| --- | --- |
| Testfall ID | ID zur eindeutigen Identifizierung des Testfalls |
| Testziel ID | Id des Testziels |
| Testobjekt ID | Was wird getest? |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Zu erfüllender Test 1 |
| Test Schritte | Was muss der Tester tun? |
| Erwartetes Resultat | Was ist das erwartete Resultat? |
| Tatsächliches Resultat | Was ist das tatsächliche Resultat? (falls möglich Screenshot einfügen) |
| Kommentar | Bemerkungen hier festhalten |
| Fehlerklasse | Fehlerklasse angeben |

Tabelle 35: Testfall Vorlage

### Testfälle

|  |  |
| --- | --- |
| Testfall ID | 1 |
| Testart ID | 1 |
| Testziel ID | 1 |
| Testobjekt ID | 1 |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Die Website ist von aussen aufrufbar. |
| Test Schritte | 1. Insomnia oder Postman aufrufen 2. webapp.switzerlandnorth.azurecontainer.io:4444/ aufrufen 3. Get Abfrage schicken |
| Erwartetes Resultat | Der Webserver antwortet mit einer Nachricht / Auf den Webserver kann nicht zugegriffen werden. |
| Tatsächliches Resultat | Der Webserver antwortet mit einer Nachricht / Auf den Webserver kann nicht zugegriffen werden. |
| Kommentar |  |
| Fehlerklasse |  |

Tabelle 36: Testfall 1

|  |  |
| --- | --- |
| Testfall ID | 2 |
| Testart ID | 2 |
| Testziel ID | 2 |
| Testobjekt ID | 1 |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Es müssen Daten postiert werden können |
| Test Schritte | Dieser Testfall wird mit JEST realisiert. |
| Erwartetes Resultat | Jest gibt keinen Fehler zurück / Jest gibt Fehler zurück |
| Tatsächliches Resultat | Jest gibt keinen Fehler zurück / Jest gibt Fehler zurück |
| Kommentar |  |
| Fehlerklasse |  |

Tabelle 37: Testfall 2

|  |  |
| --- | --- |
| Testfall ID | 3 |
| Testart ID | 2 |
| Testziel ID | 3 |
| Testobjekt ID | 1 |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Es können Daten geholt werden |
| Test Schritte | Dieser Testfall wird mit JEST realisiert. |
| Erwartetes Resultat | InJest gibt keinen Fehler zurück / Jest gibt Fehler zurück |
| Tatsächliches Resultat | Jest gibt keinen Fehler zurück / Jest gibt Fehler zurück |
| Kommentar |  |
| Fehlerklasse |  |

Tabelle 37: Testfall 3

|  |  |
| --- | --- |
| Testfall ID | 4 |
| Testart ID | 1 |
| Testziel ID | 4 |
| Testobjekt ID | 1,2 |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Die Webapplikation ist mit der Datenbank verbunden. |
| Test Schritte | 1. Insomnia starten 2. webapp.switzerlandnorth.azurecontainer.io:4444/api/messages/ im Request-Header aufrufen 3. Nachricht im Body einfügen 4. Post Request 5. Auf gleiche Endpunkt Get Request |
| Erwartetes Resultat | Die vorher eingegebenen Daten werden wieder zurückgegeben / Es werden keine Daten zurückgegeben |
| Tatsächliches Resultat | Die vorher eingegebenen Daten werden wieder zurückgegeben / Es werden keine Daten zurückgegeben. |
| Kommentar |  |
| Fehlerklasse |  |

Tabelle 37: Testfall 4

# Realisierung

## Webapplikation lokal erstellen

Die Webapplikation wurde zuerst lokal auf dem Laptop des Kandidaten erstellt und abgelegt. Der Kandidat hat die Webapplikation nach der in der Konzeptphase beschriebenen Softwarearchitektur nachgebaut. Zum Testen der Datenbank wurde eine lokale MySQL Datenbank verwendet.

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 23: Datenbank lokal

Abbildung 22: Ordnerstruktur lokal

## Lokale Webapplikation testen/überarbeiten

Die Webapplikation ist nun so weit fertig. Bevor ein Docker Container daraus erstellt wird, wird die Webapplikation getestet und gegebenenfalls überarbeitet. Um die Webapplikation lokal zu testen, wurde Insomnia verwendet.

### Get-Request auf Rootendpunkt

Wie im untenstehenden Bild zu sehen ist, antwortet die Webapp mit einer Message auf den Get-Request.

Ein Bild, das Text, Monitor, Screenshot, Bildschirm enthält.

Automatisch generierte Beschreibung

Abbildung 23: Get-Request auf Rootendpunkt

### Postmessage auf messageendpunkt

Es wird nun eine Post-Request ausgeführt und der Server antwortet auch hier mit dem Status 200, was darauf hinweist, dass die DB die Daten erfolgreich eingespeichert hat.

Ein Bild, das Text, Monitor, Screenshot, schwarz enthält.

Automatisch generierte Beschreibung

Abbildung 24: Post-Request auf Messageendpunkt

### Daten in Datenbanktabelle überpüfen

Nun wird MySQL-Workbench verwendet, um zu überprüfen, ob die vorher gesendeten Daten in die Datenbank abgespeichert wurden. Wie in unterstehender Grafik ersichtlich ist, wurden die Daten erfolgreich in der Datenbank abgespeichert

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 25: Datenbanktabelle lokale Datenbank

### Get-Request auf Messageenpunkt

Nun gilt es alle vorher gespeicherten Daten wieder auszugeben. Wie in untenstehender Grafik ersichtlich ist, wird auch die vorher erstellte russische Message wieder zurückgegeben.

Ein Bild, das Text, Screenshot, Monitor enthält.

Automatisch generierte Beschreibung

Abbildung 26: Get-Request auf messageenpunkt

## Container von Webapplikation erstellen

Als nächster Schritt wurde ein Docker Container erstellt. Dafür brauchte in ein docker File, wie auch ein dockerignore File. Die genauen Schritte sind in der Anleitung im Anhang der Dokumentation zu finden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 27: Dockerfile

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 28: Dockerignore

Ein Bild, das Text, Monitor, Screenshot, Bildschirm enthält.

Automatisch generierte Beschreibung

Abbildung 29: docker images

## Datenbank in Cloud konfigurieren

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 30: Konfiguration DB

Verbindunginformationen holen

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 31: Verbindungsinformatioinen

Im Knexfile Verbindung definieren.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 32: Knexfile

Ein Bild, das Text enthält.

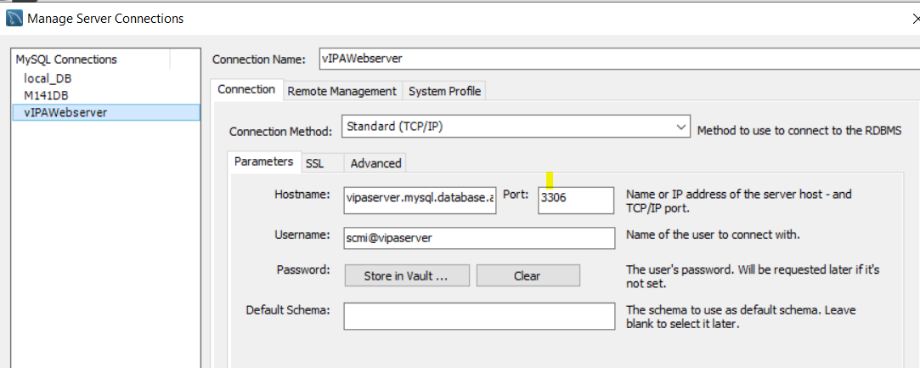
Automatisch generierte Beschreibung   
Es wurde in MySQL Workbench eine neue Verbindung hinzugefügt und überprüft, ob die Webapplikation die Migrations erstellt hat. Wie in unterstehender Grafik ersichtlich, war die Verbindung erfolgreich und die Tabellen wurden erstellt.

Abbildung 34: Migrationstabellen

Abbildung 34: Workbench Verbindung

## Testen/überarbeiten des Systems

Zuerst werden alle Testfälle durchgeführt. Falls notwendig (Fehler bei den Testfällen), wird das System überarbeitet.

### Testfälle

|  |  |
| --- | --- |
| Testfall ID | 1 |
| Testart ID | 1 |
| Testziel ID | 1 |
| Testobjekt ID | 1 |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Die Website ist von aussen aufrufbar. |
| Test Schritte | 1. Insomnia oder Postman aufrufen 2. webapp.switzerlandnorth.azurecontainer.io:4444/ aufrufen 3. Get Abfrage schicken |
| Erwartetes Resultat | Der Webserver antwortet mit einer Nachricht / Auf den Webserver kann nicht zugegriffen werden. |
| Tatsächliches Resultat | Der Webserver antwortet mit einer Nachricht. |
| Kommentar |  |
| Fehlerklasse | M0 |

Tabelle 36: Testfall 1

|  |  |
| --- | --- |
| Testfall ID | 2 |
| Testart ID | 2 |
| Testziel ID | 2 |
| Testobjekt ID | 1 |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Es müssen Daten postiert werden können |
| Test Schritte | Dieser Testfall wird mit JEST realisiert. |
| Erwartetes Resultat | Jest gibt keinen Fehler zurück / Jest gibt Fehler zurück |
| Tatsächliches Resultat | Jest gibt keinen Fehler zurück / Jest gibt Fehler zurück |
| Kommentar |  |
| Fehlerklasse |  |

Tabelle 37: Testfall 2

|  |  |
| --- | --- |
| Testfall ID | 3 |
| Testart ID | 2 |
| Testziel ID | 3 |
| Testobjekt ID | 1 |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Es können Daten geholt werden |
| Test Schritte | Dieser Testfall wird mit JEST realisiert. |
| Erwartetes Resultat | InJest gibt keinen Fehler zurück / Jest gibt Fehler zurück |
| Tatsächliches Resultat | Jest gibt keinen Fehler zurück / Jest gibt Fehler zurück |
| Kommentar |  |
| Fehlerklasse |  |

Tabelle 37: Testfall 3

|  |  |
| --- | --- |
| Testfall ID | 4 |
| Testart ID | 1 |
| Testziel ID | 4 |
| Testobjekt ID | 1,2 |
| Voraussetzungen | Testprodukt muss vorhanden sein und Vorkenntnisse müssen vorhanden sein. |
| Beschreibung | Die Webapplikation ist mit der Datenbank verbunden. |
| Test Schritte | 1. Insomnia starten 2. webapp.switzerlandnorth.azurecontainer.io:4444/api/messages/ im Request-Header aufrufen 3. Nachricht im Body einfügen 4. Post Request 5. Auf gleiche Endpunkt Get Request |
| Erwartetes Resultat | Die vorher eingegebenen Daten werden wieder zurückgegeben / Es werden keine Daten zurückgegeben |
| Tatsächliches Resultat | Die vorher eingegebenen Daten werden wieder zurückgegeben. |
| Kommentar |  |
| Fehlerklasse | M0 |

Tabelle 37: Testfall 4

## Ein Bild, das Text enthält. Automatisch generierte Beschreibung

# Abbildungsverzeichnis

[Abbildung 1: Arbeitsplatz 15](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814167)

[Abbildung 2: Dokumentablage 16](#_Toc88814168)

[Abbildung 3: OneDrive Cloud 17](#_Toc88814169)

[Abbildung 4: OneDrive Datenwiederherstellung 18](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814170)

[Abbildung 5: Datenwiederherstellung mit Git 18](#_Toc88814171)

[Abbildung 6: Version als Zip Ordner 19](#_Toc88814172)

[Abbildung 7: Git Übersicht Commits 20](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814173)

[Abbildung 8: Git Hard Reset 21](#_Toc88814174)

[Abbildung 9: Git Hard Reset nacher 21](#_Toc88814175)

[Abbildung 10: Hermes Phasen 22](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814176)

[Abbildung 11: Hermes Module 24](#_Toc88814177)

[Abbildung 12: Projektorganisation 25](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814178)

[Abbildung 13: Smartziele 45](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814179)

[Abbildung 14: Persönliche Vorgehensziele 45](#_Toc88814180)

[Abbildung 15: Projektziele 46](#_Toc88814181)

[Abbildung 16: Schnittstellen und Umfeld 48](#_Toc88814182)

[Abbildung 17: Systemarchitektur Kontextsicht 50](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814183)

[Abbildung 18: Softwarearchitektur Bausteinsicht (Whitebox) 56](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814184)

[Abbildung 19: Deployment in Azure Ablaufübersicht 59](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814185)

[Abbildung 20: Anwendungsfälle 60](#_Toc88814186)

[Abbildung 21: ERM 61](#_Toc88814187)

[Abbildung 22: Ordnerstruktur lokal 67](#_Toc88814188)

[Abbildung 23: Get-Request auf Rootendpunkt 68](#_Toc88814189)

[Abbildung 24: Post-Request auf Messageendpunkt 69](#_Toc88814190)

[Abbildung 25: Datenbanktabelle lokale Datenbank 70](#_Toc88814191)

[Abbildung 26: Get-Request auf messageenpunkt 71](#_Toc88814192)

[Abbildung 27: Dockerfile 72](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814193)

[Abbildung 28: Dockerignore 72](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814194)

[Abbildung 29: docker images 72](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814195)

[Abbildung 30: Konfiguration DB 73](#_Toc88814196)

[Abbildung 31: Verbindungsinformatioinen 73](#_Toc88814197)

[Abbildung 32: Knexfile 74](#_Toc88814198)

[Abbildung 34: Workbench Verbindung 74](https://d.docs.live.net/2f9adfba9eb6e95b/Dokumente/TF_Bern/Informatik/VorIPA/Dokumentation/Version-0.4/Vor-IPA_Schmid_Michael.docx#_Toc88814199)

# Tabellenverzeichnis

[Tabelle 1: Vorkenntnisse 10](#_Toc88813884)

[Tabelle 2: Arbeitsnotebook 16](#_Toc88813885)

[Tabelle 3: Projektrollen 26](#_Toc88813886)

[Tabelle 4: Schadensausmass 27](#_Toc88813887)

[Tabelle 5: Eintrittswahrscheinlichkei 27](#_Toc88813888)

[Tabelle 6: Risikograph 28](#_Toc88813889)

[Tabelle 7: Risikomatrix 29](#_Toc88813890)

[Tabelle 8: Tagesjournal 15.11.21 32](#_Toc88813891)

[Tabelle 9: Tagesjournal 16.11.21 33](#_Toc88813892)

[Tabelle 10: Tagesjournal 17.11.21 35](#_Toc88813893)

[Tabelle 11: Tagesjournal 19.11.21 37](#_Toc88813894)

[Tabelle 11: Tagesjournal 19.11.21 39](#_Toc88813895)

[Tabelle 11: Tagesjournal 19.11.21 40](#_Toc88813896)

[Tabelle 12: Mittel und Methoden Erläuterung 50](#_Toc88813897)

[Tabelle 13: Detaillierte Anforderungen 51](#_Toc88813898)

[Tabelle 14: Vourraussetzungen an den Client 51](#_Toc88813899)

[Tabelle 15: Requestsyntax 51](#_Toc88813900)

[Tabelle 16: Verwendete Azure Dienste und Ressourcen 53](#_Toc88813901)

[Tabelle 17: Ressourcengruppen 53](#_Toc88813902)

[Tabelle 18: MySQL Server 54](#_Toc88813903)

[Tabelle 19: Datenbank 54](#_Toc88813904)

[Tabelle 20: Firewall Manager 54](#_Toc88813905)

[Tabelle 21: Containerregistrierung 55](#_Toc88813906)

[Tabelle 22: Containerinstanzen 55](#_Toc88813907)

[Tabelle 23: Root Endpunkt 57](#_Toc88813908)

[Tabelle 24: Messages Endpunkt 57](#_Toc88813909)

[Tabelle 25: Azure Zugriff 58](#_Toc88813910)

[Tabelle 26: MySQL Zugriff 58](#_Toc88813911)

[Tabelle 27: Deploymentschritte 60](#_Toc88813912)

[Tabelle 28: Anwendungsfälle 60](#_Toc88813913)

[Tabelle 29: Datenbank Spezifikationen 61](#_Toc88813914)

[Tabelle 30: Testart 62](#_Toc88813915)

[Tabelle 30: Testziele 62](#_Toc88813916)

[Tabelle 31: Testobjekte 62](#_Toc88813917)

[Tabelle 32: Testarten 63](#_Toc88813918)

[Tabelle 33: Testabdeckung 63](#_Toc88813919)

[Tabelle 34: Mängelklassen 63](#_Toc88813920)

[Tabelle 35: Testfall Vorlage 65](#_Toc88813921)

[Tabelle 36: Quellenverzeichnis 77](#_Toc88813922)

[Tabelle 37: Glossar 78](#_Toc88813923)

# Quellenverzeichnis

|  |  |
| --- | --- |
| **Begriff** | **Quelle** |
| Dokumentation Vorlage | Von Felix Imobersteg – ist auf Moodle verfügbar. |
| Vorlage für Konzepte | Modul 431 und 143 – ist auf Moodle verfügbar. |
| Vorlage für Testkonzept | <https://www.hermes.admin.ch/de/projektmanagement/verstehen/ergebnisse/testkonzept.html>. |
| SQL Curdate Funktion | <https://ubiq.co/database-blog/how-to-get-current-date-and-time-in-mysql/> |
| PHP-Nachschlagewerk | Ist als Buch auf moodle unter dem Modul 151 verfügbar. |
| Testfälle | Betriebsinterne Dokumentvorlage |

Tabelle 36: Quellenverzeichnis

# Glossar

|  |  |
| --- | --- |
| **Begriff** | **Bedeuting** |
| LAMP-Server | Ein Server mit einem Linux-Betriebssystem, worauf Apache, MYSQL und PHP läuft. |
| TFBern | Technische Fachschule Bern |
| IPA | Individuelle Praktische Prüfung |

Tabelle 37: Glossar

# Anhänge

Relevante Informationen, welche nicht innerhalb der Phasen aufgeführt sind, werden hier abgelegt oder es wird beschrieben, wo diese Informationen zu finden sind.

## Anleitung

Javascript Web-Applikation als Docker Container in der Cloud verfügbar machen

**Ausgangslage**

**Sie haben eine fertige Node JS-Webapplikation lokal auf ihrem Gerät und wollen diese nun in der Cloud verfügbar machen.**

**Zielgruppe  
  
Lernende des dritten Lehrjahres Abteilung Informatik**

**Voraussetzungen**

* **Docker installiert**
* **Azure konfiguriert**
* **Kenntnisse mit Azure und Docker**

**Was ist nicht Bestandteil dieser Anleitung**

* **Auf Konfiguration von Docker und Azure Cloud wird nicht näher eingegangen**
* **Intergration der Datenbank ist nicht Bestandteil dieser Anleitung**
* **Port forwarding**

**Bitte beachten**

* ***Kursive Teile in den Befehlen* = Stehen für individuelle Teil, dies hängt davon ab, wie man seine Objekte benannt an. z.B. mein Server heisst vielleicht anders als Ihrer.**
* **Einstellungen, die nicht im Bild zu sehen sind, wurden immer auf der Standardeinstellung gelassen**
* **Es wurde kein Portforwarding erstellt. D.H der korrekte Port muss angegeben werden beim Aufrufen über den Browser co.**

**Anleitung**

|  |  |  |
| --- | --- | --- |
| ID | Beschreibung | **Grafik** |
| 1 | Azure 🡪 Icon Ressourcengruppen anwählen 🡪 auf Knop „Erstellen“ klicken | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 2 | Spezifikationen eingeben und erstellen | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 3 | Containerregistrierungen -> erstellen -> Spezifikationen eingeben und erstellen klicken | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 4 | Ins Verzeichnis der Webapp navigieren 🡪 File Namens „dockerfile“ erstellen 🡪 Inhalt vom Bild einfügen.  Achtung eventuell haben sie einen anderen Port definiert als ich. In diesem Fall ihren Port bei EXPOSE angeben. | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 5 | Im selben Verzeichnis File Namens „.dockerignore“ erstellen 🡪 Inhalt vom Bild einfügen. | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 6 | Sicherstellen, dass im File Pacakge.json unter scripts “start“: “node index.js“ definiert ist. | Ein Bild, das Text, Screenshot, Monitor, Bildschirm enthält.  Automatisch generierte Beschreibung |
| 6 | Im Terminal zum Ordner der Webapplikation navigieren | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 7 | Im Terminal den Befehl „docker build -t a*nmeldeserver/registrierungsname“* Die Daten dazu sind auf azure unter Einstellungen 🡪 Zugriffschlüssel zu finden (beim Befehl alles Lowercase). | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 8 | Überprüfen, ob das Image erstellt wurde mit dem Befehl „docker image ls“ |  |
| 9 | Im Anmeldeserver einloggen. |  |
| 10 | Nun muss im Terminal der Username und das Passwort eingegeben werden. Sie finden dies unter Zugriffsschlüssel nachdem sie Administratorbenutzer aktiviert haben. | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 11 | Das Docker Image mit dem Befehl „docker push *repository“* hochladen |  |
| 12 | Um zu kontrollieren ob das Pushen erfolgreich war, auf azure navigieren 🡪 Containerregistrierung 🡪 Dienste 🡪 Repositorys 🡪 auf Repository klicken. | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 13 | Azure 🡪 Containerinstanzen 🡪 Erstellen 🡪 Spezifikationen eingeben 🡪 erstellen   Hier muss als Imagequelle Azure Container Registry gewählt werden, weil wir dort unser Repository gepusht haben.  Als Port den vorher definierten Port eingeben . | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |
| 14 | Azure 🡪 Ressourcegruppen 🡪 vorher erstellte Containerinstanz wählen (in meinem Fall: dockertestwebappcontainer)  Hier kann der Webapp Container verwaltet werden | Ein Bild, das Text, Screenshot, drinnen enthält.  Automatisch generierte Beschreibung |
| 15 | Nun kann die Webapp im Browser angerufen werden 🡪 FQDN kopieren und Port hinzufügen. | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |