**IPA 2024  
ABSCHLUSSARBEIT**

**OVWEB**

**Validation in insurance offer creation**

**27.03.2024**

**Individuelle Praxisarbeit Kanton Zürich**

**Andrei Mititelu**

**Generali AG**

**TBZ**

**IT Apprentices**

**Aschvin Umachandran**

**Sukhnaaz Singh**

**Roger Sorg**

**Roman Babenko**

**Applikationsentwicklung**

**IPERKA**

**Kandidat / Author:**

**Firma:**

**Berufsfachschule:**

**Abteilung:**

**Hauptexperte:**

**Nebenexperte:**

**Berfusbildner:**

**Verantwortliche Fachkraft:**

**Fachrichtung:**

**Projektvorgehensmodell:**

A red circle with a white letter i in it

Description automatically generated

Teil 1 – Administrativer Teil 3

1. Projektaufbauorganisation 4

2. Organisation der Arbeitsergebnisse 5

2.1 Codesicherung 5

2.2 Dokumentationssicherung 5

3. Standards 6

4. Zeitplan 9

5. Arbeitsjournal 10

5.1 Tag 1: Mittwoch, 03.04.2024 10

5.2 Tag 2: Donnerstag, 04.04.2024 11

6. Kurzfassung 12

6.1 Ausgangssituation 12

6.2 Projektvorgehen 12

6.3 Ergebnis 12

7. Projektvorgehen 13

8. Informieren 14

8.1 Ziele der Aufgabenstellung 14

8.2 Vorgaben und Anforderungen 14

8.3 Ist-Zustand 14

8.3.1 Projekt-Überblick 14

8.3.2 Backend 14

8.3.3 Frontend 14

8.3.4 ANGEBOTsoftware Rechenkern 14

8.3.5 Systemgrenzen 14

8.4 User Stories und Akzeptanzkriterien 14

8.4.1 Funktionale Anforderungen 14

8.4.2 Nichtfunktionale Anforderungen 14

9. Planen 14

9.1 Soll-Zustand 14

9.2 Mockups 15

9.3 Testkonzept 15

10. Entscheiden 15

11. Realisieren 15

12. Kontrollieren 15

13. Auswerten 15

13.1 Vergleich Ist/Soll 15

13.2 Persönliches Fazit 15

13.2.1 Was lief gut? 15

13.2.2 Was lief schief? 15

13.2.3 Was habe ich aus der Erfahrung gelernt? 15

13.3 Schlussreflexion 15

14. Glossar 15

15. Quellenvezeichnis 15

15.1 Einsatz von generativen Tools 15

16. Danksagungen 16

1. Teil 1 – Administrativer Teil

## Projektaufbauorganisation

**Lehrbetrieb und Durchführungsort**Generali AG   
Soodmattenstrasse 4   
8134, Adliswil   
[058 472 75 00](https://www.google.com/search?client=firefox-b-d&q=generali+adliswil+&lqi=ChFnZW5lcmFsaSBhZGxpc3dpbCIDiAEBSKmIiN3lgICACFofEAAYABgBIhFnZW5lcmFsaSBhZGxpc3dpbCoECAIQAJIBEGluc3VyYW5jZV9hZ2VuY3mqAVcKDS9nLzExYjY2anJoc2QQASoMIghnZW5lcmFsaSgFMh8QASIb7rjLivujVg1VYAHWgXdZ_l1vC9h1_he9K8OnMhUQAiIRZ2VuZXJhbGkgYWRsaXN3aWw)  
[life.ch@generali.com](mailto:life.ch@generali.com)  
  
  
**Hauptexperte**Aschvin Umachandran   
aschvin00@gmail.com   
078 826 61 57  
  
  
**Verantwortliche Fachkraft:**Roman Babenko  
[Generali](mailto:marc.siegmund+pkorg@gmail.com) AG  
Soodmattenstrasse 4  
8134, Adliswil  
076 410 87 29  
roman.babenko@generali.com  
  
  
**Auftraggeber:**Daniel Sager  
Generali AG  
Soodmattenstrasse 4  
8134, Adliswil  
daniel.sager@generali.com

**Kandidat**Andrei Mititelu   
Breitwiesstrasse 15   
8135, Langnau am Albis   
079 343 49 40  
[andrei@gimli.ro](mailto:andrei@gimli.ro)  
  
  
**Nebenexpertin**Sukhnaaz Singh  
sukhnaazsingh@hotmail.com  
076 411 42 35  
  
  
**Berufsbildner / Lehrfirma:**Sorg Roger  
[Generali](mailto:marc.siegmund+pkorg@gmail.com) AG  
Soodmattenstrasse 4  
8134, Adliswil  
058 472 40 42  
roger.sorg@generali.com

## Organisation der Arbeitsergebnisse

Um den Verlust der Dateien im Falle eines Ausfalles zu vermeiden, werden während der IPA verschiedene Versionierungs- und Speicherungssysteme genutzt.   
Mithilfe dieser Systeme kann ebenfalls auf ältere Versionen der Dokumente und Dateien zugegriffen werden.

### Codesicherung

Für den Code wird einen eigenständigen Branch in der Monorepo des Projekts erstellt. Änderungen am Code oder an der Projektdokumentation werden auf diesem Branch gepusht und somit gespeichert.

**Link zum Branch:** https://github.com/IT-Apprentices/insurance-offer-management/tree/OVB-170-Offerten-Vergleich-IPA

Folgend steht die Historie des Branches :

### Dokumentationssicherung

Für die Sicherung aller Dateien für die Dokumentation wird ein separates GitHub Repository erstellt. Auf diesem wird täglich ein Mal gepusht. Ebenfalls werden die Word-Dokumente automatisch auf OneDrive gespeichert.

**Link zum Repository:** https://github.com/Rad14nt/IPA\_2024\_Docs

Folgend steht die Historie des Repositorys :

## Standards

Im Ramen des Projektes OVWEB und dieser IPA werden folgende Technologien eingesetzt:

|  |  |
| --- | --- |
| **Frontend** | |
| Name | Einsatz |
| React 18 | Frontend UI Framework |
| NextJS 14 | React-based Framework |
| Typescript | Frontendsprache |
| Jest | Testing Framework |
| Storybook | Development und Testing Framework |
| Shadcn/ui | Komponenten Library |
| Prettier und ESLint | Code Formattierungs- und Analysetool |
| NextAuth | Authentifizierung |

|  |  |
| --- | --- |
| **Backend** | |
| Name | Einsatz |
| Java | Backendsprache |
| Springboot | Java-Framework |
| Lombok | Annotationen |
| JUnit | Testing |
| Hibernate | ORM-Framework |

|  |  |
| --- | --- |
| **Daten Layer** | |
| Name | Einsatz |
| PostgreSQL | Datenbank |
| Flyway | Datenbank-Versionierungstool |

|  |  |
| --- | --- |
| **Daten Layer** | |
| Name | Einsatz |
| PostgreSQL | Datenbank |
| Flyway | Datenbank-Versionierungstool |

|  |  |
| --- | --- |
| **Identity Provider** | |
| Name | Einsatz |
| OpenID Connect | Identitätsprotokoll |
| Keycloak | Identitäts- und Zugriffsmanagement |

|  |  |
| --- | --- |
| **Entwicklungsumgebung** | |
| Name | Einsatz |
| JetBrains IntelliJ IDEA | Entwicklungsumgebung |
| Docker | Containerisierungsplatform |
| Windows 11 | Operating System |

|  |  |
| --- | --- |
| **UI/UX Design** | |
| Name | Einsatz |
| Figma | Design Tool |
| Generali-Styleguide | Styleguide |

|  |  |
| --- | --- |
| **Dokumentation und Administration** | |
| Name | Einsatz |
| Microsoft Office (Word und Excel) | Dokumentation |
| Draw.io und IntelliJ IDEA | Diagramme bzw. Skizzen |
| OneDrive | Automatische Speicherung |
| Github | Versionierung der Dokumentation und Backups |
| ChatGPT | Textgenerierung und Informationsbeschaffung |
| Duden.de | Rechtschreibprüfung |

|  |  |
| --- | --- |
| **Projektmanagement und Development Tools** | |
| Name | Einsatz |
| Jira | Projektmanagement-Tool |
| Github | Versionskontroll- und Kollaborationsplattform |

## Zeitplan

**A screenshot of a computer

Description automatically generated**

## Arbeitsjournal

### Tag 1: Mittwoch, 03.04.2024

|  |  |
| --- | --- |
| **Tätigkeiten** | - Aufgabenstellung und Anforderung studieren  - Zeitplan erstellen  - Ist-Zustand Analyse  - User Stories und Akzeptanzkriterien definieren  - Dokumentation – Teil 1, Administrativer Teil  - Dokumentation – Informieren dokumentieren  - Dokumentation – Arbeitsjournal schreiben |
| **Probleme** | SendGrid stoppte das Backend vom starten, da mir die Authentifizierungsdaten im .env File fehlten. |
| **Hilfestellung** | Mein Arbeitskollege nach einem Passwort für SendGrid gefragt. |
| **Reflexion** | Ich konnte heute gut mit der IPA starten und kam auch sehr zügig voran, vorallem war die Vorlage besonders hilfreich. Den Zeitplan musste ich jedoch ganz neu gestalten, dies hat ein wenig mehr Zeit in anspruch genommen als erwartet, aber ich konnte diese Zeit schnell beim ausfüllen des Administrativen Teils meines Dokumentes, wo ich ein wenig schneller war als erwartet.  Weil ich mir für den Zeitplan jedoch Zeit genommen habe, erlaube dieser mir sehr strukturiert in meiner IPA einzugehen. Etwas Stress gab es trotzdem, da ich noch am Anfang der IPA bin ist alles noch ein wenig unsicher.  Ebenfalls hatte ich heute Probleme beim Aufsetzen meines Projektes, ich brauchte diesen aufzustarten, um Screenshots für den IST-Zustand zu machen, hatte aber Probleme mit einer neuen Implementation von SendGrid, ein E-Mail Service, welcher von jemand anders implementiert wurde. Es fehlten die Login Daten im environment variables File, und diese waren nicht in der Dokumentation zu finden, daher musste ich meinen Kollegen Samuel fragen, mir diese von seinem Projekt zu kopieren.  Allgemein bin ich aber mit dem heutigen Fortschritt froh und freue mich, dass meine IPA soweit nach Plan läuft. |
| **Nächste Schritte** | Informierphase abschliessen und fertig dokumentieren.  Planeungsphase beginnen |

### Tag 2: Donnerstag, 04.04.2024

|  |  |
| --- | --- |
| **Tätigkeiten** | - Aufgabenstellung und Anforderung studieren  - Zeitplan erstellen  - Ist-Zustand Analyse  - User Stories und Akzeptanzkriterien definieren  - Dokumentation – Teil 1, Administrativer Teil  - Dokumentation – Informieren dokumentieren  - Dokumentation – Arbeitsjournal schreiben |
| **Erreichte Ziele** |  |
| **Probleme** |  |
| **Hilfestellung** |  |
| **Ausserplanmässige**  **Arbeiten** |  |
| **Überstunden** |  |
| **Reflexion** |  |
| **Nächste Schritte** |  |

Teil 2 – Indivudueller Praktischer Teil

## Kurzfassung

### Ausgangssituation

In der Applikation OVWEB kann man schon eine Todesfallversicherung-Offerte erstellen. Jedoch ist ein vergleich zwischen verschiedene Versicherungssummen noch nicht möglich. Ziel dieser IPA ist, für dem Kunden einen benutzerfreundlichen Vergleich zu präsentieren. Dabei wird für die Berechnung der Offerten der Rechenkern des ANGEBOTsoftwares benutzt.

### Projektvorgehen

Die Realisierung des Projekts orientiert sich an der IPERKA-Methode, welche ein systematisches Vorgehen bei der Projektarbeit gewährleistet. Mehr zu dieser Methodik finden Sie unter Kapitel 7.

Nach eine detaillierte Informations- und Planungsphase, während der der IST-Zustand dokumentiert und der SOLL-Zustand definiert, ist es möglich mit einem deutlichen Verständnis der Ziele in die Umsetzungsphase zu starten.

In der Realisierungsphase wurden Änderungen am Controller und am Service vorgenommen. Um diese Änderungen zu benutzen werden auch das Model, DTOs und die Datenbank.

Im Frontend Teil des Projektes wird die Detail- und die Erstellungsseiten angepasst. Im Form bei der Erstellung braucht es neue Felder und es muss eine Validation implementiert werden. Die Detail Seite muss so angepasst werden, dass sie übersichtlich höchstens drei Offerten gleichzeitig anzeigen kann.

### Ergebnis

## Projektvorgehen

Die IPERKA-Methode ist ein systematischer Ansatz zur Problemlösung und Projektplanung.   
Sie gliedert den Prozess in sechs Phasen:   
Informieren, Planen, Entscheiden, Realisieren, Kontrollieren und Auswerten.   
Diese Methode hilft dabei, komplexe Aufgaben strukturiert anzugehen, Ziele klar zu definieren und die Umsetzung effektiv zu gestalten.

**Informieren**:

In der ersten Phase geht es darum, alle relevanten Informationen zu sammeln und zu bewerten. Dies beinhaltet das Verständnis der Aufgabe, das Identifizieren von notwendigen Ressourcen und das Analysieren des jetzigen Zustandes sowie der Systemumgebung.

**Planen**:

Nachdem alle Informationen gesammelt wurden, wird in dieser Phase ein detaillierter Plan erstellt. Dabei werden Ziele definiert (Soll-Zustand), erforderliche Schritte zur Zielerreichung ausgearbeitet, Diagramme erstellt, Mockups angefertigt und ein Testkonzept festgelegt.

**Entscheiden**:

Basierend auf den gesammelten Informationen und dem erarbeiteten Plan wird eine Entscheidung über die optimale Implementierung getroffen. Die Auswahl erfolgt anhand eines Kriterienkatalogs.

**Realisieren**:

Die Realisierungsphase umfasst die aktive Umsetzung des Plans. In dieser Phase werden die geplanten Schritte ausgeführt und kontinuierlich dokumentiert, um den Fortschritt nachvollziehbar zu machen.

**Kontrollieren**:

Nach der Umsetzung ist es wichtig, den Fortschritt zu überprüfen. Dies geschieht anhand der zuvor definierten Tests, um sicherzustellen, dass alles wie geplant funktioniert und alle Ziele erreicht wurden.

**Auswerten**:

Nach Abschluss der Arbeit wird der gesamte Prozess ausgewertet. Diese Reflexion beinhaltet die Bewertung des erreichten Ergebnisses im Vergleich zu den gesetzten Zielen. Ebenfall werden gesammelten Erfahrungen analyisiert, um sowohl die Stärken aus auch die Verbesserungspotenzialle des Projektes zu bewerten.

## Informieren

### Ziele der Aufgabenstellung

Das Ziel dieser Aufgabe ist es, dem Berater einen Prämienvergleich mit maximal drei Todesfallversicherungsofferten darzustellen. Auch sollte dem Berater eine Fehlermeldung gezeigt werden, falls die eingegebene Daten für die Berechnung der Offerten die Business-Rules verletzen.

### Vorgaben und Anforderungen

Der Berater sollte in der Lage sein, bis zu drei verschiedene Versicherungssummen einzugeben. Auf der Detailseite sollte es dem Berater möglich sein, alle drei Offerten gleichzeitig einzusehen, um diese präsentieren und vergleichen zu können. Die für die Offerte eingegebenen Daten müssen mittels des Rechenkerns der bestehenden Angebotssoftware validiert werden. Eventuell auftretende Fehler sollten im Frontend angezeigt werden, damit der Berater diese einsehen kann.

### Ist-Zustand

Im Moment kann man in der OVWEB Applikation eine Todesfallversicherung mittels Formular erstellen. Diese wird dann im Backend der OVWEB Applikation gespeichert. Im Frotnend wird nach der Erstellung auf der Detailseite mit der Offerten-ID weitergeleitet, wo noch mal ein Call auf die Datenbank gemacht wird, um die Daten mittels Rechenkern der alten ANGEBOTSoftware zu berechnen. Diese Antwort wird dann zum Frontend auf der Detail-Seite weitergeleitet, wo diese angezeigt wird, egal ob ein Fehler oder eine Offerte zurückgegeben wird.

Neben dem gibt es auch eine Übersichtsseite für Offerten, wo alle bisher erstellte Offerten stehen. Wenn man auf diese klickt, wird man wieder auf die Detailseite der spezifischen Offerte weitergeleitet, mittels des gleichen Prozesses wie oben beschrieben.

#### Projekt-Überblick

Das folgende Diagramm beschreibt übersichtlich die OVWEB Applikation so wie die Abhängigkeiten.

A diagram of a software

Description automatically generated

Da die Erstellung einer Offerte kompliziert ist, wurde ein Sequenzdiagramm gezeichnet, um diesen Prozess vereinfacht darzustellen.

A screenshot of a computer

Description automatically generated

#### Frontend

Hier werden die IPA relevante Seiten dargestellt.

A white background with red and blue text

Description automatically generatedOffer Dashboard

A white paper with red and blue lines

Description automatically generatedOffer Creation

A screenshot of a computer

Description automatically generatedOffer Details – Erfolgreiche Berechnung der Offerte

A screenshot of a computer

Description automatically generatedOffer Details – Fehlgeschlagene Berechnung der Offerte

#### Backend

Diese Klassendiagramme dienen dem Überblick und dem besseren Verständnis der Systemarchitektur und der Beziehungen bezüglich Offerten zwischen den verschiedenen Klassen.

Für Offerten gibt es Controller und Services für die Verbindung zwischen dem Frontend und Backend der OVWEB Applikation, sowie auch einen Controller und einen Service für die Verbindung zwischen dem OVWEB Backend und das Rechenkern der ANGEBOTsoftware.

Im Rahmen dieser IPA wird aber nichts an der zweiten Verbdinung nichts geändert, deswegen wird diese nicht in Detail betrachtet.

Die Offerten werden nicht berechnet und inklussive der Kundendaten in der Datenbank gespeichert. Grund dafür ist, dass sich die Kunden- sowie Offertendaten nach der Erstellung der Offerte nicht geändert werden dürfen. Die Berechnung könnte sich jedoch ändern, zum Beispiel wenn in 2025 die Prämien steigen, deswegen wird die Offerte bei jedem Aufruf der Detail Seite im Frontend neu berechnet.

**Controller und Service**

A screenshot of a computer screen

Description automatically generated

**Model und DTOs**

A screenshot of a computer

Description automatically generated

#### ERM-Datenbankdiagramm

A screenshot of a computer

Description automatically generated

#### ANGEBOTsoftware Rechenkern

Das ANGEBOTsoftware ist eine bestehende Applikation und kann somit nicht geändert bzw. angepasst werden. Wir greifen auf dem ... Endpoint D2 der Applikation zu um die Offerten zu berechnen und diese gleichzeitig zu validieren.

#### Systemgrenzen

Im Rahmen dieser Arbeit liegt der Fokus auf die Erstellung, Berechnung, Validierung und Anzeige mehrer Todesfallversicherungsofferten. Daher gehört das ANGEBOTSoftware, spezifisch das Rechenkern davon, als Abhängigkeit zu dem Projekt.

Es ist wichtig zu erwähnen, dass SendGrid, welches für E-Mails implementiert ist, und Keycloak, nicht Teil der Systemgrenzen dieser spezifischen Aufgabe sind.

### User Stories und Akzeptanzkriterien

#### Funktionale Anforderungen

#### Nichtfunktionale Anforderungen

## Planen

### Soll-Zustand

Entscheidmatrix fuer Design

### Mockups

### Testkonzept

## Entscheiden

## Realisieren

## Kontrollieren

Testprotokoll

Testbericht

Beschreiben was bewusst nicht getestet wurde.

## Auswerten

### Vergleich Ist/Soll

### Persönliches Fazit

#### Was lief gut?

#### Was lief schief?

#### Was habe ich aus der Erfahrung gelernt?

### Schlussreflexion

## Glossar

## Quellenvezeichnis

### Einsatz von generativen Tools

## Danksagungen