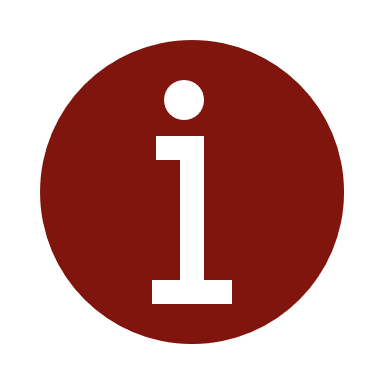
IPA 2024  
 Abschlussarbeit

OVWEB – Workflow Unterbruch

20. März 2024



|  |  |
| --- | --- |
| Kanditat / Autor:  Firma:  Berufsfachschule:  Abteilung:  Hauptexperte:  Nebenexperte:  Berufsbildner:  Verantwortliche Fachkraft:  Fachrichtung:  Projektvorgehensmodell: | Samuel Hajnik  Generali AG  TBZ  IT Apprentices  Marc Siegmund  Janis Marty  Roger Sorg  Roman Babenko  Applikationsentwicklung  IPERKA |

Inhalt

[Vorwort 4](#_Toc160616991)

[Teil 1 - Administration 5](#_Toc160616992)

[1. Projektorganisation 5](#_Toc160616993)

[2. Organisation der Arbeitsergebnisse 6](#_Toc160616994)

[3. Datensicherung 6](#_Toc160616995)

[3.1 Code 6](#_Toc160616996)

[3.2 Dokumentation 6](#_Toc160616997)

[4. Deklaration der benützten Firmenstandards 7](#_Toc160616998)

[5. Zeitplan 8](#_Toc160616999)

[6. Arbeitsjournal 9](#_Toc160617000)

[6.1 Dienstag, 05.03.2024 9](#_Toc160617001)

[6.2 Mittwoch, 06.03.2024 10](#_Toc160617002)

[6.3 Donnerstag, 07.03.2024 11](#_Toc160617003)

[6.4 Freitag, 08.03.2024 12](#_Toc160617004)

[6.5 Dienstag, 12.03.2024 13](#_Toc160617005)

[6.6 Mittwoch, 13.03.2024 14](#_Toc160617006)

[6.7 Donnerstag, 14.03.2024 15](#_Toc160617007)

[6.8 Freitag, 15.03.2024 16](#_Toc160617008)

[6.9 Dienstag, 19.03.2024 17](#_Toc160617009)

[6.10 Mittwoch, 20.03.2024 18](#_Toc160617010)

[Teil 2 – Projekt 20](#_Toc160617011)

[7. Kurzfassung 20](#_Toc160617012)

[7.1 Ausgangslage 20](#_Toc160617013)

[7.2 Umsetzung 20](#_Toc160617014)

[7.3 Ergebnis 20](#_Toc160617015)

[8. Projektbeschreibung 21](#_Toc160617016)

[8.1 IPERKA 21](#_Toc160617017)

[8.1.1 Informieren 21](#_Toc160617018)

[8.1.2 Planen 21](#_Toc160617019)

[8.1.3 Entscheiden 21](#_Toc160617020)

[8.1.4 Realisieren 22](#_Toc160617021)

[8.1.5 Kontrollieren 22](#_Toc160617022)

[8.1.6 Auswerten 22](#_Toc160617023)

[9. Einleitung 23](#_Toc160617024)

[10. Informieren 23](#_Toc160617025)

[10.1 Ziele der Aufgabenstellung 23](#_Toc160617026)

[10.2 Vorgaben 23](#_Toc160617027)

[10.3 Ist-Analyse 23](#_Toc160617028)

[10.3.1 Frontend 24](#_Toc160617029)

[10.3.2 Backend 25](#_Toc160617030)

[10.3.3 Überblick 26](#_Toc160617031)

[10.3.4 Klassendiagramme 27](#_Toc160617032)

[10.4 User Stories 28](#_Toc160617033)

[10.4.1 Rollen 28](#_Toc160617034)

[10.4.2 Funktionale Anforderungen 28](#_Toc160617035)

[10.4.3 Nicht funktionale Anforderungen: 31](#_Toc160617036)

[11. Planen 34](#_Toc160617037)

[11.1 Realisierungskonzept 34](#_Toc160617038)

[11.2 Testkonzept 34](#_Toc160617039)

[11.3 Aktivitätsdiagramme 34](#_Toc160617040)

[11.4 Sequenzdiagramm 37](#_Toc160617041)

[11.5 Klassendiagramme 38](#_Toc160617042)

[11.6 Datenmodell 38](#_Toc160617043)

[11.7 Design 38](#_Toc160617044)

[11.7.1 Variante 1 39](#_Toc160617045)

[11.7.2 Variante 2 40](#_Toc160617046)

[12. Entscheiden 41](#_Toc160617047)

[13. Realisieren 41](#_Toc160617048)

[13.1 Projektumgebung 41](#_Toc160617049)

[13.2 Datenbank 41](#_Toc160617050)

[13.3 Backend Unit Tests 41](#_Toc160617051)

[13.4 Frontend Unit Tests 42](#_Toc160617052)

[14. Kontrollieren 42](#_Toc160617053)

[14.1 Testkonzept 42](#_Toc160617054)

[14.2 Testprotokoll Samuel Hajnik 42](#_Toc160617055)

[15. Auswerten 43](#_Toc160617056)

[15.1 Vergleich Ist / Soll 43](#_Toc160617057)

[15.2 Fazit zur IPA 43](#_Toc160617058)

[15.3 Reflexion 43](#_Toc160617059)

[Danksagung 44](#_Toc160617060)

[Glossar 45](#_Toc160617061)

[Abbildungsverzeichnis 46](#_Toc160617062)

[Quellenverzeichnis 47](#_Toc160617063)

Vorwort

Teil 1 - Administration

1. Projektorganisation

**Lehrbetrieb und Durchführungsort: Nebenexperte:**Generali AG Marty Janis  
Soodmattenstrasse 4 janis.marty@ubs.com  
8134 Adliswil 079 227 53 58  
[058 472 75 00](https://www.google.com/search?client=firefox-b-d&q=generali+adliswil+&lqi=ChFnZW5lcmFsaSBhZGxpc3dpbCIDiAEBSKmIiN3lgICACFofEAAYABgBIhFnZW5lcmFsaSBhZGxpc3dpbCoECAIQAJIBEGluc3VyYW5jZV9hZ2VuY3mqAVcKDS9nLzExYjY2anJoc2QQASoMIghnZW5lcmFsaSgFMh8QASIb7rjLivujVg1VYAHWgXdZ_l1vC9h1_he9K8OnMhUQAiIRZ2VuZXJhbGkgYWRsaXN3aWw)  
life.ch@generali.com

**Kandidat:**Hajnik Samuel  
Im Heumoos 2  
8906 Bonstetten  
077 968 38 56  
samuel.hajnik@generali.com

**Berufsbildner / Lehrfirma:**Sorg Roger  
Generali AG  
Soodmattenstrasse 4  
8134 Adliswil  
058 472 40 42  
roger.sorg@generali.com

**Verantwortliche Fachkraft:**Babenko Roman  
Generali AG  
Soodmattenstrasse 4  
8134 Adliswil  
076 410 87 29  
roman.babenko@loewenzahn.dev

**Hauptexperte:**Siegmund Marc  
[marc.siegmund+pkorg@gmail.com](mailto:marc.siegmund+pkorg@gmail.com)  
078 656 35 16

1. Organisation der Arbeitsergebnisse

Dieses Kapitel beschreibt, wie während der IPA sicherstellt wird, dass alle relevanten Daten gesichert sind und nötigenfalls wiederhergestellt werden können.

1. Datensicherung

Die Datensicherung ist essenziell für den Fall, dass Probleme auftreten sollen, somit ist sie ein wichtiger Bestandteil der IPA. Für die tägliche Sicherung der Arbeit wurde auf verschiedene Speicherungssysteme gesetzt.

* 1. Code

Für den Code wurde ein neuer Github-Branch in der Team-Repository erstellt.

**Branch**: [Link]

Hier ist der Verlauf der Commits auf den Branch:

* 1. Dokumentation

Durch OneDrive wird der neuste Stand laufend gespeichert. Für zusätzliche Sicherheit wird eine separate Github Repository am Schluss von jedem Tag ein Backup erstellt.  
**Repository:** [Link]

Hier ist der Verlauf der Commits auf die Repository:

1. Deklaration der benützten Firmenstandards

Im Entwickler-Team werden folgende Technologien verwendet.

|  |  |
| --- | --- |
| Frontend | |
| **Name** | **Einsatz** |
| Next.js | Frontend-Technologie |
| shadcn | Komponenten Design Bibliothek |
| Typescript | Frontendsprache |
| Jest | Testing |
| Storybook | Frontend Dokumentation |
| **Backend** | |
| Springboot | Backend-Technologie |
| Java | Backendsprache |
| Lombok | Annotations für Automatisierung |
| JUnit | Testing |
| **Daten Layer** | |
| PostgreSQL | Datenbank |
| Flyway | Datenbankversionierungstool |
| **Identidy Provider** | |
| OpenID Connect | Identitätsprotokoll |
| Keycloak | Identitätsmanagement |

1. Zeitplan
2. Arbeitsjournal
   1. ****Dienstag, 05.03.2024****

|  |  |
| --- | --- |
| Tätigkeiten | * Zeitplan erstellen * Dokumentation starten * User Stories * Expertengespräch |
| **Erreichte Ziele** | * Zeitplan * User Stories |

|  |
| --- |
| **Probleme** |
| - |
| Hilfestellung |
| Roman hat mir nach dem Expertengespräch noch Tipps gegeben zu meinem Zeitplan. |
| Ausserplanmässige Arbeiten |
| - |
| Reflexion |
| ***Was lief Heute positiv:***  **Ich konnte Heute positiv und motiviert in meine IPA starten. Ich habe mir vorgenommen vieles möglichst schnell zu erledigen, damit ich der begrenzten Zeit vorauskomme. Ich konnte den Zeitplan in einer relative schnellen Zeit erledigen, was ich mir nicht dachte, da es doch eine anstrengende Aufgabe ist. Zugleich hat mir das Dokumentieren auch einfach gefallen Heute, ich konnte schon vieles runterschreiben, vor allem in der Phase Informieren, was mich nach vorne drängen wird in den nächsten Tagen. Beim Expertengespräch konnte ich sehr gute Einblicke holen, wie es laufen wird, was wichtig ist und auf was ich achten muss.**  ***Was lief nicht so gut:***  **Vor dem Expertengespräch habe ich mich kurzfristig entschieden meinen Zeitplan noch zu umstrukturieren, da es für mich ungenau war. Nach dem Gespräch holte ich mir noch Hilfe bei Roman, damit ich eine Idee bekomme, wie ich es verbessern kann. Er hat mir vorgeschlagen Formeln zu benutzen. Somit konnte ich meinen Zeitplan verbessern.**  ***Fazit:*** Heute war ein erfolgreicher Tag. Ich konnte super starten und fortlaufend arbeiten. Das Expertengespräch hat mich nur sicherer gemacht und ich bin mir sicher, dass es in den nächsten Tagen gut fortlaufen wird. |

* 1. Mittwoch, 06.03.2024

|  |  |
| --- | --- |
| Tätigkeiten | * Diagramme * Mockups * Planen anfangen |
| **Erreichte Ziele** | * Diagramme * Mockups * Planen anfangen |

|  |
| --- |
| **Probleme** |
| - |
| Hilfestellung |
| Roman gefragt, ob die Frist von N-Tagen auch in der Datenbank implementiert werden soll und ob mein Vorschlag für das neue Object richtig aussieht. |
| Ausserplanmässige Arbeiten |
| * Mehr Zeit in Planung reingesteckt |
| Reflexion |
| ***Was lief Heute positiv:***  **Der Tag verlief schnell, da ich ständig am arbeiten war und ich all meine Heute vorgesehenen Aufgaben erledigen konnte. Am Anfang hatte ich Stress, da ich mir nicht sicher war, ob ich genug Zeit eingeplant habe für die Diagramme und Mockups. Mit den Aktivitätsdiagrammen konnte ich rechtzeitig fertig werden, jedoch beim Sequenzdiagramm habe ich ein bisschen mehr Zeit gebraucht, da ich ein Programm dafür benutzt habe: PlantUML, mit dem ich mich noch nicht auskannte, aber es hat den Workflow verbessert. Es erlaubte mir das Diagramm code-mässig zu schreiben, was mich Zeit gekostet hat, da ich die Syntax lernen musste, aber schlussendlich sind mir die Resultate gelungen und ich denke es ist schneller gegangen, als wenn ich es von Hand designet hätte. Mit den Mockups hat es ich auch keine Mühe, es war einfach es zu gestalten mit Figma, zuerst habe ich die Hauptfarben aus dem Generali Styleguide als Variabeln deklariert und mit Frames gearbeitet, das hat meine Effizienz gesteigert.**  ***Was lief nicht so gut:***  **Die Datenmodellierung musste ich auf den Nachmittag verschieben, da die Vorarbeit noch nicht im Main branch gemerged war, aber es hat mich nicht gehindert. Nach dem Modellieren habe ich Roman noch nach seiner Meinung gefragt.**  ***Fazit:*** Wiederum ein guter Tag. Trotz einiger Stolpersteine konnte ich den Tag zielgesetzt durcharbeiten. Die “Rekonstruktion” des Zeitplans gestern hat mir sicherlich geholfen, da ich jetzt eine bessere Übersicht habe. Ich habe meine Ziele erreicht und bin somit bereit für den nächsten Tag. |

* 1. Donnerstag, 07.03.2024

|  |  |
| --- | --- |
| Tätigkeiten |  |
| **Erreichte Ziele** |  |

|  |
| --- |
| **Probleme** |
|  |
| Hilfestellung |
|  |
| Ausserplanmässige Arbeiten |
|  |
| Reflexion |
| ***Was lief Heute positiv:***  ***Was lief nicht so gut:***  ***Fazit:*** |

* 1. Freitag, 08.03.2024

|  |  |
| --- | --- |
| Tätigkeiten |  |
| **Erreichte Ziele** |  |

|  |
| --- |
| **Probleme** |
|  |
| Hilfestellung |
|  |
| Ausserplanmässige Arbeiten |
|  |
| Reflexion |
| ***Was lief Heute positiv:***  ***Was lief nicht so gut:***  ***Fazit:*** |

* 1. Dienstag, 12.03.2024

|  |  |
| --- | --- |
| Tätigkeiten |  |
| **Erreichte Ziele** |  |

|  |
| --- |
| **Probleme** |
|  |
| Hilfestellung |
|  |
| Ausserplanmässige Arbeiten |
|  |
| Reflexion |
| ***Was lief Heute positiv:***  ***Was lief nicht so gut:***  ***Fazit:*** |

* 1. Mittwoch, 13.03.2024

|  |  |
| --- | --- |
| Tätigkeiten |  |
| **Erreichte Ziele** |  |

|  |
| --- |
| **Probleme** |
|  |
| Hilfestellung |
|  |
| Ausserplanmässige Arbeiten |
|  |
| Reflexion |
| ***Was lief Heute positiv:***  ***Was lief nicht so gut:***  ***Fazit:*** |

* 1. Donnerstag, 14.03.2024

|  |  |
| --- | --- |
| Tätigkeiten |  |
| **Erreichte Ziele** |  |

|  |
| --- |
| **Probleme** |
|  |
| Hilfestellung |
|  |
| Ausserplanmässige Arbeiten |
|  |
| Reflexion |
| ***Was lief Heute positiv:***  ***Was lief nicht so gut:***  ***Fazit:*** |

* 1. Freitag, 15.03.2024

|  |  |
| --- | --- |
| Tätigkeiten |  |
| **Erreichte Ziele** |  |

|  |
| --- |
| **Probleme** |
|  |
| Hilfestellung |
|  |
| Ausserplanmässige Arbeiten |
|  |
| Reflexion |
| ***Was lief Heute positiv:***  ***Was lief nicht so gut:***  ***Fazit:*** |

* 1. Dienstag, 19.03.2024

|  |  |
| --- | --- |
| Tätigkeiten |  |
| **Erreichte Ziele** |  |

|  |
| --- |
| **Probleme** |
|  |
| Hilfestellung |
|  |
| Ausserplanmässige Arbeiten |
|  |
| Reflexion |
| ***Was lief Heute positiv:***  ***Was lief nicht so gut:***  ***Fazit:*** |

* 1. Mittwoch, 20.03.2024

|  |  |
| --- | --- |
| Tätigkeiten |  |
| **Erreichte Ziele** |  |

|  |
| --- |
| **Probleme** |
|  |
| Hilfestellung |
|  |
| Ausserplanmässige Arbeiten |
|  |
| Reflexion |
| ***Was lief Heute positiv:***  ***Was lief nicht so gut:***  ***Fazit:*** |

Teil 2 – Projekt

1. Kurzfassung
   1. Ausgangslage
   2. Umsetzung
   3. Ergebnis
2. Projektbeschreibung
   1. IPERKA

Diese Arbeit wird mit der IPERKA-Methode implementiert. Sie wird in sechs Schritte gegliedert: Informieren, Planen, Entscheiden, Kontrollieren und Auswerten. Die Bedeutung von Planung wird in dieser Methodik besonders hervorgehoben, um eine genaue Vorgehensweise zu fördern. Es ist die bevorzugte Projektmanagementmethode an der Berufsschule.

Das Projekt ist in die sechs Phasen unterteilt, um eine geordnete Struktur beizubehalten.

* + 1. Informieren

Die Ziele und Herausforderungen des Prozesses werden identifiziert. Es wird untersucht, welche Prozesse gelöst werden müssen, welche Ressourcen dafür gebraucht werden und Fragen werden geklärt. Dies ist wichtig für den Verstand der Aufgabe, sowie für die Erstellung des Zeitplans.

* + 1. Planen

Es wird ein detaillierter Plan erstellt, um einen strukturierten Projektablauf zu garantieren. Dies umfasst Festlegung von Aufgaben, Verantwortlichkeiten, sowie auch ein Realisierungskonzept und ein Testkonzept.

* + 1. Entscheiden

Der bestmögliche Lösungsweg wird ausgewählt in dem man die verschiedenen Ansätze zusammen vergleicht. Es ist wichtig sich zu überlegen, ob man die Herangehensweise auch implementieren kann und ob die Ideen überhaupt sinnvoll sind.

* + 1. Realisieren

Sobald alles klar definiert ist und alle Entscheidungen getroffen sind, wird mit der Realisierung begonnen. Die Arbeitsabläufe werden stetig protokolliert und die IST-Werte werden im Zeitplan eingetragen.

* + 1. Kontrollieren

Erst wenn die Realisierung vollständig implementiert wurde, werden die Resultate so getestet, wie es im Tesprotokoll steht. Fehlgeschlagene Tests werden entweder direkt korrigiert und behoben oder als Fehler vermerkt und dokumentiert.

* + 1. Auswerten

Als Abschluss gibt es eine Reflexion über die Arbeit und die Erfahrungen, die man gesammelt hat. Man geht durch all seine Schritte, welche man innerhalb des Projektes gemacht hat, und es wird untersucht was gut gelaufen ist und wo Verbesserungen möglich sind. Das ist auch praktisch für die Umsetzung zukünftiger Projekte.

1. Einleitung

In diesem Teil zeige ich die Umsetzung des Projektes mit der IPERKA-Methode auf. Die Kapitel sind in die Phasen unterteilt, um sich einfacher durch die Struktur des Dokumentes orientieren zu können.

1. Informieren
   1. Ziele der Aufgabenstellung

Das Ziel ist, dem Benutzer zu ermöglichen, die Kundenerstellung zu stoppen, um später fortzufahren. Wenn die Aufgabe die Frist von N-Tagen übersteigt, wird an den Benutzer eine E-Mail geschickt mit dem Link zu dieser Aufgabe, um zu fortfahren.

* 1. Vorgaben

Die Kundenerstellung soll bei einem beliebigen Punkt unterbrochen werden können. Die ausgefüllten Daten sollten in der Datenbank gespeichert werden ohne Validierung. Die Erstellung kann später weitergeführt werden. Sobald die Frist von N-Tagen bei einer Aufgabe abläuft, wird der Benutzer per E-Mail informiert.

* 1. Ist-Analyse

Die Offerten Verwaltung Web, kurz OVWEB, wird dazu dienen die alte Applikation in eine Cloud-basierte Webanwendung zu verwandeln. Das Programm ist gedacht für Kundenverwaltung und Erstellung von Offerten. Im jetzigen Stand der Applikation ist es möglich sich einzuloggen über den Keycloak-Server und Kunden erstellen via ein Formular. In der bestehenden Legacy Applikation mussten die Anwender die Erstellung von Kunden manchmal unterbrechen. Zum Beispiel mussten zusätzliche Abklärungen beim Kunden gemacht werden. Dieser war aber in diesem Moment nicht verfügbar. Bei einem teilerfassten Kunden gingen so die bisherigen Eingaben verloren, da die Validierung im halbfertigen Zustand der Erfassung eine Speicherung nicht zulässt. Diese Anforderung setze ich mir an, um den Unterbruch so einfach wie möglich zu gestalten.

* + 1. Frontend

Das Frontend wurde mit Next.js 14 umgesetzt und ist nach dem Generali-Styleguide gestaltet. Die Session wird mit NextAuth geregelt. Für Components wird die Shadcn Library benutzt. Für die Backend Kommunikation wird ein TypeScript-Client verwendet.

Ein Bild, das Text, Screenshot, Schrift, Logo enthält.

Automatisch generierte Beschreibung

Abbildung 1: Home-Page

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 2: Kundenformular Ist Zustand

* + 1. Backend

Das Backend wurde mit dem Java-Framework Springboot realisiert. Es besteht ein Model, das User, Customer, Task und Adresse beinhaltet. Als Query-Sprache wird PostgreSQL verwendet. Die Datenbank Anbindung läuft via einen lokalen Docker-Container und wird durch Flyway versioniert.

* + 1. Überblick

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte BeschreibungIn der folgenden Abbildung wird die Architektur der OVWEB-Applikation dargestellt.

Abbildung 3: OVWeb-Struktur

* + 1. Klassendiagramme

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte BeschreibungFür die Customer, User und Task Modelle gibt es Service und Controller für CRUD-Operationen (Create, Read, Update und Delete).

Abbildung 4:Klassendiagramm Models

Ein Bild, das Text, Screenshot, Multimedia-Software, Software enthält.

Automatisch generierte Beschreibung

Die Task Klasse besteht aus weiteren Subklassen.

Abbildung 5: Klassendiagramm Task mit Subklassen

Ein Bild, das Text, Screenshot, Multimedia-Software, Software enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Abbildung 6: Klassendiagramm Controller

Abbildung 7: Service Klassen

* + 1. Datenbankdiagramm

Die Datenbank wird durch Flyway verwaltet und enthält die gleichen Entitäten, wie die Models im Backend.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 8: Datenbankdiagramm

* 1. User Stories

Um ein tieferes Verständnis der Anforderungen zu gewinnen, werden User Stories erfasst. Die Anforderungen der Benutzer und der Implementierung der neuen Workflow-Funktionalität liegen dabei im Vordergrund. Ich habe jedoch auch nicht-funktionale Anforderungen als User Stories zu hinunterschreiben, um einen besseren Überblick zu behalten über die IPA relevanten Kriterien. Somit kann ich auch beim Testen gleich überprüfen, ob die Entwicklerthemen vollständig umgesetzt wurden, sowie am Ende den Ist- und Soll-Zustand vergleichen.

* + 1. Rollen

Es gibt Benutzer (Berater, Versicherungsvertreter), Entwickler und Administratoren.

* + 1. Funktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Frontend Button | 4 | 1 |
| Beschreibung |  |  |

Als ein **Benutzer** kann ich auf einen **Button klicken**, damit ich **den Fortschritt speichern kann**.

|  |
| --- |
| Acceptance Criteria |

**AC 1.1:** Auf der Customer Creation Page soll ein klar ersichtlicher «Später Fortfahren» Button ersichtlich sein.

**AC 1.2:** Die Daten werden in der Datenbank abgespeichert.

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Anzeige in Task Liste | 2 | 2 |
| Beschreibung |  |  |

Als ein **Benutzer** kann ich **nicht fertige Customer Creation Tasks sehen**, damit ich **einen Überblick habe über meine Tasks**.

|  |
| --- |
| Acceptance Criteria |

**AC 2.1:** Es sollte relevante Details wie Datum/Uhrzeit, Erfassungsobjekt und/oder den Namen des Kunden anzeigen.

**AC 2.2:** Jede unterbrochene Aufgabe in der Übersichtsliste sollte angeklickt werden können, um die Fortsetzung zu ermöglichen.

**AC 2.3:** Nachdem die konfigurierbare Frist von N-Tagen abgelaufen ist, wird die Aufgabe visuell vorgehoben.

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Formular Vorbefüllung | 4 | 3 |
| Beschreibung |  |  |

Als ein **Benutzer** kann ich auf den **«Fortsetzen» Button klicken**, damit ich **mit** **der Aufgabe weiterfahren kann**.

|  |
| --- |
| Acceptance Criteria |

**AC 3.1:** Auf der Customer Creation Page sollen die abgerufenen Daten ersichtlich und korrekt formatiert sein.

**AC 3.2:** Die Backend-Logik sollte unterbrochene Aufgaben aus der Datenbank abrufen

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Task Liste Sortierung | 2 | 4 |
| Beschreibung |  |  |

Als ein **Benutzer** kann ich auf den **auf das «Beschreibung» Feld klicken**, damit ich **die Tasks alphabetisch sortieren kann**.

|  |
| --- |
| Acceptance Criteria |

**AC 4.1:** Die Tasks werden von A-Z absteigend oder aufsteigend, wenn man doppelt klickt, angezeigt

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Task Liste Filter | 2 | 5 |
| Beschreibung |  |  |

Als ein **Benutzer** kann ich auf den **ein Begriff in das «Suchen» Feld eingeben**, damit ich **die Tasks filtrieren kann**.

|  |
| --- |
| Acceptance Criteria |

**AC 5.1:** Das Suchfeld erscheint auf der Task-Liste

**AC 5.2:** Es werden nur Tasks angezeigt, die die Sucheingabe beinhalten.

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| E-Mail versenden | 5 | 6 |
| Beschreibung |  |  |

Als ein **Benutzer** kann ich **per E-Mail über überfällige Aufgaben benachrichtigt werden**.

|  |
| --- |
| Acceptance Criteria |

**AC 6.1:** Nach dem die konfigurierbare Frist von N-Tagen abgelaufen ist, wird eine E-Mail an den Benutzer geschickt mit dem folgenden Inhalt: Ein Link zur Weiterführung des abgebrochenen Workflows.

**AC 6.2:** Der Link leitet auf die richtige Aufgabe weiter.

* + 1. Nicht funktionale Anforderungen:

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Usability | 1 | 7 |
| Beschreibung |  |  |

Als ein **Benutzer** möchte ich, dass **der Unterbruch benutzerfreundlich ist**, damit ich **ohne Hilfe die Aufgabe beenden kann**.

|  |
| --- |
| Acceptance Criteria |

**AC 7.1:** Die Anzeige ist in einer klar verständlichen Sprache, die Knöpfe sind klar beschriftet.

**AC 7.2:** Soll ein Fehler vorkommen, dann wird der Benutzer durch Hinweise darauf aufmerksam gemacht.

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Logging | 2 | 8 |
| Beschreibung |  |  |

Als ein **Administrator** möchte ich, dass **die neuen Funktionalitäten strukturiertes Logging implementieren**, damit **die Fehler einfacher zu beheben sind.**

|  |
| --- |
| Acceptance Criteria |

**AC 8.1:** Strukturiertes Logging wurde aktiv umgesetzt.

**AC 8.2:** Alle notwendigen Informationen wurden in den Logs enthalten, um die Aktionen nachvollziehen zu können.

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Tracing | 4 | 9 |
| Beschreibung |  |  |

Als ein **Entwickler** möchte ich, dass **jede Funktion mit einer Trace-Id versehen werden kann**, damit ich **die Fehlerbehebung vereinfacht werden kann**.

|  |
| --- |
| Acceptance Criteria |

**AC 9.1:** Bei jeder Frontend-Anfrage wird eine Trace-Id generiert, die dann an den Backend geschickt wird.

**AC 9.2:** Die Trace-Id wird konsquent durch alle Dienste hindurch weitergeleitet und in den Logs festgehalten.

**AC 9.3:** Es kann ein Diagramm der Service-Aufrufe erstellt werden, indem man die Trace-IDs als Grundlage verwendet.

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Unit Testing | 4 | 10 |
| Beschreibung |  |  |

Als ein **Entwickler** möchte ich, dass **die neuen Funktionalitäten durch automatisiertes Testing abgedeckt sind**, damit **ich die Wirksamkeit überprüfen kann**.

|  |
| --- |
| Acceptance Criteria |

**AC 10.1:** Für die neuen Funktionen wurden folgende Unit-Tests implementiert: Positiv- Negativszenarien, Null/Non-Null Tests für Datensätze, Empty/Not-Empty für Collections.

**AC 10.2:** Die Unit-Tests wurden im Backend, sowie im Frontend implementiert.

|  |  |  |
| --- | --- | --- |
| Titel der User Story | Schätzung in Stunden | User Story Nummer |
| Wiederverwendbarkeit | 2 | 11 |
| Beschreibung |  |  |

Als ein **Entwickler** möchte ich, dass **UI-Komponenten wiederverwendbar sind**, damit **ich sie in der Zukunft gebrauchen kann**.

|  |
| --- |
| Acceptance Criteria |

**AC 11.1:** Die Komponenten sind so aufgebaut, dass sie aus kleineren und einfacheren Komponenten bestehen.

1. Planen

Planen ist die zweite Phase des IPERKA-Vorgehenmodels. Es wird geplant und aufgezeigt, wie man von Ist-Zustand zum Soll-Zustand kommt.

* 1. Realisierungskonzept

Ein Realisierungskonzept dient dazu die Aufgabe bis in Detail zu planen. Der Zweck ist von der bisherigen Übersicht eine Strategie zu entwickeln, um einen Leitfaden zu haben für eine koordinierte Umsetzung. Das Konzept wurde in drei Unterthemen aufgeteilt, damit man eine erleichterte Übersicht der Aktivitäten zwischen den Schnittstellen hat.

* 1. Testkonzept
  2. Aktivitätsdiagramme

Um den Prozess zu veranschaulichen, wurden zwei Aktivitätsdiagramme erstellt. Einer für den Unterbruch und ein anderer für die Task-Liste. Es dient als Grundlage, um sicherzustellen, dass alle Projektbeteiligten vom Gleichen reden.

Ein Bild, das Text, Screenshot, Zahl, Diagramm enthält.

Automatisch generierte Beschreibung

Abbildung 9: Aktivitätsdiagramm Workflow Unterbruch

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 10: Aktivitätsdiagramm Pendenzenliste

* 1. Sequenzdiagramm

Das Sequenzdiagramm verhilft eine Übersicht zu geben, wie die Schnittstellen miteinander kommunizieren, vom User bis zur Datenbank. Dazu ergibt sich auch ein tieferes Verständnis, was für Methoden gebraucht werden.

Ein Bild, das Text, Diagramm, parallel, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 11: Sequenzdiagramm

* 1. Klassendiagramm

Für die Implementierung des Workflow Unterbruchs muss eine neue Subklasse «ResumeCustomerWorkflow» an Task hinzugefügt werden. Dabei müssen alle Felder aus der Customer Entity implementiert werden, damit diese auch abgespeichert werden können.

Ein Bild, das Schwarz, Dunkelheit enthält.

Automatisch generierte Beschreibung

* 1. Datenmodell

Das Datenmodel beinhaltet die gleichen Datensätze, wie das Klassendiagramm.

* 1. Design

Der Sinn der Mockups ist es ein besseres Verständnis wie auch einen Überblick zu verschaffen, wie alles am Ende aussehen soll und was alles zu implementieren ist. Dabei wird die User Experience besonders in Betracht bezogen. Die Mockups stellen die zwei, zu bearbeitenden, Seiten dar: Kundenerstellung und Pendenzenliste. Für die Umsetzung des User Interfaces für die Pendenzenliste, im Fall der Fristablauf nach N-Tagen, wurden zwei Varianten in Figma umgesetzt. Zusätzlich wurde eine Seite erstellt für den Fall, dass ein Error auftreten sollte.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 12: Kundenerstellung

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 13: Error Message

* + 1. Variante 1

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 14: Fristablauf Variante 1

* + 1. Variante 2

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abbildung 15: Fristablauf Variante 2

1. Entscheiden
2. Realisieren
   1. Projektumgebung
   2. Datenbank
   3. Backend Unit Tests
   4. Frontend Unit Tests
3. Kontrollieren
   1. Testkonzept
   2. Testprotokoll Samuel Hajnik
4. Auswerten
   1. Vergleich Ist / Soll

Checkliste User Stories

* 1. Fazit zur IPA
  2. Reflexion

Danksagung

Glossar

|  |  |
| --- | --- |
| Begriff | Erklärung |
|  |  |

Abbildungsverzeichnis

Quellenverzeichnis