



IPA

Eingehende Message-Queue-Nachrichten im Web-UI

IPA von Joel Vontobel

Ergon Informatik AG
6. November 2024

Inhalt

I	Umfeld und Ablauf	2
1	Aufgabenstellung	3
1.1	Ausgangslage	3
1.2	Detaillierte Aufgabenstellung	3
1.3	Mittel und Methoden	4
1.4	Vorkenntnisse	4
1.5	Vorarbeiten	4
1.6	Neue Lerninhalte	5
1.7	Arbeiten in den letzten 6 Monaten	5
2	Projektaufbauorganisation	6
3	Benützte Firmenstandards	7
4	Arbeitsumgebung	8
4.1	Arbeitsplatz	8
4.1.1	Office Arbeitsplatz	8
4.2	Verwendete Tools	9
5	Versionierung und Sicherung der Arbeitsergebnisse	10
5.1	Verwendung von Git zu Versionierung	10
5.2	Quellcode	10
5.3	Probe-IPA Dokumentation	11
6	Projektmanagementmethode	13
6.1	IPERKA	13
6.2	Alternative Methode	14
7	Arbeitsprotokoll	15
II	Projekt	16
	Glossar	17
	Abbildungsverzeichnis	17
	Quellenverzeichnis	18

Teil I

Umfeld und Ablauf

1 Aufgabenstellung

In diesem Kapitel ist die Aufgabenstellung der Probe-IPA aufgeführt. Die Inhalte wurden zu einem grossen Teil von der originalen Aufgabenstellung übernommen und angepasst.

1.1 Ausgangslage

Die Firma Ergon Informatik AG entwickelt seit einigen Jahren eine Transaktions-Autorisierungs-Lösung für einige Banken in der Schweiz. Dieses Projekt heisst CardX und wird durch ein 12 zwölköpfiges Team umgesetzt. In diesem Projekt ist der Lernende seit Januar 2024 tätig und kennt sich deshalb schon ein bisschen aus.

Das Projekt kommuniziert mit verschiedenen bankspezifischen IT-Systemen wie zum Beispiel dem Kernbankensystem oder Service-Büros über Message-Queues. Diese Message-Queues werden in der Datenbank-Tabelle `MQ_TABLE` (eingehende Nachrichten) und `MQ_OUT` (ausgehende Nachrichten) zwischengespeichert. Falls man diese Message-Queues anschauen oder bearbeiten möchte, muss man dies in der Datenbank machen. Weil das ziemlich umständlich ist, besteht die Aufgabe des Lernenden jetzt daraus diese Message-Queues in einem Web-GUI darzustellen und sinnvolle Interaktionen mit diesen Daten anzubieten.

1.2 Detaillierte Aufgabenstellung

Minimalanforderungen Das Ziel dieser Aufgabe ist es, den Inhalt der Tabelle `MQ_TABLE` in diesem Web-GUI sichtbar zu machen und dem Nutzer sinnvolle Interaktionen mit diesen Daten anzubieten.

Es soll im Web-GUI eine neue Seite erstellt werden mit dem Inhalt einer Tabelle, welche die Message-Queues abbilden soll. Die Tabelle soll eine Hand voll Spalten besitzen, sodass sie übersichtlicher ist. Die Seite soll stimmig in das UI eingebaut werden. Im Backend sollen die neuen Methoden mithilfe von Unit-Tests abgedeckt werden.

Zusätzlich kann der Lernende noch zwischen zwei Erweiterungen entscheiden, welche er implementieren möchte.

Erweiterung: Pagination Bei der ersten Erweiterung ist das Ziel ein Paginator zur Tabelle hinzuzufügen. Eine Pagination ist wenn man eine Liste oder Tabelle auf ein paar Einträge limitiert und anschliessend weitere Einträge anzeigen kann mit einer Pfeiltaste. Dies hilft, das Laden der Seite zu verkürzen, da die Einträge, die nicht angezeigt werden, erst geladen werden, wenn sie auch wirklich gebraucht werden.

Erweiterung: Filter Die zweite Erweiterung ist ein Filtersystem. Die Seite soll nach dem Status, Inhalt der Nachricht und dem Datum gefiltert werden können. Die Filter-Werte sollen ausserdem in der URL Abgebildet werden, um das Teilen von gefilterten Ergebnissen zu vereinfachen oder den Filter als Lesezeichen ablegen zu können.

1.3 Mittel und Methoden

Technologien

- SQL
- Java
- TypeScript
- HTML
- Angular

Tools

- IntelliJ (IDE)
- Docker
- Bitbucket
- Confluence
- Jira
- Postman

1.4 Vorkenntnisse

Der Lernende hat bereits viele Arbeiten im Projekt CardX gemacht. Unter anderem im Backend und an CardX-spezifischen Tools. Die Codebasis hat der Lernende in den letzten 9 Monaten gut kennengelernt und findet sich gut zurecht. Der Lernende hat auch bereits die Tabelle TASK von der Datenbank in das Web-GUI gebracht, was eine ähnliche Aufgabe war wie die jetzige Aufgabenstellung.

Durch die früheren Projekte, wie ein Fussballtippspiel und eine Anmelde-Plattform für Bewerbende, konnte er bereits viel Erfahrung mit Java und Angular sammeln.

1.5 Vorarbeiten

Durch das bereits existierende Projekt und die vielen Arbeiten, die der Lernende bereits gemacht hat, musste er keine Vorarbeiten leisten.

1.6 Neue Lerninhalte

- Pagination:

Mit Pagination hat der Lernende sich noch nie auseinandergesetzt. Er hat es schon oft auf anderen Seiten gesehen aber noch nie selbst implementiert.

- Filtersystem:

Im Projekt WM-Tippspiel gab es ein Filtersystem, aber dieses hat der Lernende nicht selbst implementiert und ist so ein neuer Lerninhalt.

1.7 Arbeiten in den letzten 6 Monaten

In den letzten 6 Monaten hat der Lernende sich, wie oben schon genannt, mit dem Projekt CardX auseinandergesetzt und ist ein aktives Team Mitglied. Er hat viele verschiedene Arbeiten umgesetzt. Einige davon hier:

- CheckDB Task:

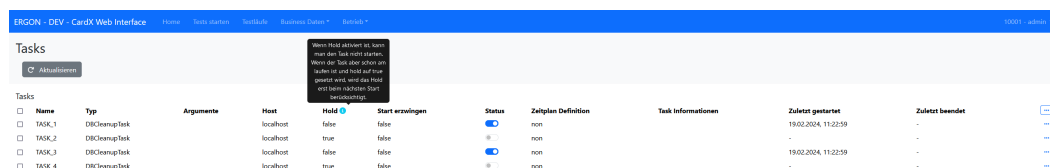
In dieser Aufgabe geht es darum, eine Aufgabe zu erstellen, welche periodisch oder manuell ausgeführt werden kann. Diese Aufgabe überprüft die Datenbankdefinition, ob immer noch alles fehlerfrei ist. Es werden Sequenzen, Indexe, Felder und mehr in eine Datei geschrieben, gespeichert und anschliessend mit der vorherigen Datei auf Veränderungen verglichen. Bei einer Veränderung schlägt die Aufgabe fehl und die Differenz wird in der Fehlermeldung angezeigt.

- ServiceLogMove und Zip:

Mit der Aufgabe ServiceLogMerge werden Protokolle des Systems in Dateien geschrieben und gespeichert. Ein Teil dieser Dateien wird jetzt mithilfe der Aufgabe in bestimmte Verzeichnisse verschoben und komprimiert. Die Dateien werden nach Bank sortiert und anschliessend in ihn vorgesehene Verzeichnis verschoben. Auch diese Aufgabe wird periodisch jeden Tag ausgeführt, um die Dateiablage übersichtlich zu halten.

- Tasks im Frontend anzeigen und bearbeiten:

Diese Aufgabe zeigt alle Tasks im Frontend an und man kann sie dort auch bearbeiten. Das Ziel von dieser Aufgabe war gleich wie die Mindestanforderungen von der Aufgabenstellung.



Name	Type	Argumente	Host	Held	Start erzeugen	Status	Zeitplan Definition	Task Informationen	Zuletzt gestartet	Zuletzt beendet
TASK_1	DBCleanupTask		localhost	false	false	on	non		19.02.2024, 11:22:59	-
TASK_2	DBCleanupTask		localhost	true	false	on	non		-	-
TASK_3	DBCleanupTask		localhost	false	false	on	non		19.02.2024, 11:22:59	-
TASK_4	DBCleanupTask		localhost	true	false	on	non		-	-

Abbildung 1.1: Anzeigen und bearbeiten der Tasks

2 Projektaufbauorganisation

In der folgenden Tabelle sind die an der Probe-IPA beteiligten Personen und ihre jeweiligen Aufgaben aufgeführt.

Person	Rolle	Aufgabe/Verantwortung
Joel Vontobel	Kandidat (K)	Umsetzen der Facharbeit
Loris Diana und Dominic Monzón	Verantwortliche Fachkraft (VF)	Facharbeit begleiten, technische Fragen beantworten, Bewertung der Facharbeit
Bernd Lienberger	Hauptexperte (HEX)	IPA bezogene Fragen beantworten, Entscheiden bei auftretenden Problemen, Besuchstermine festlegen, Fachgespräch leiten, Bewertung der Facharbeit
	Nebenexperte (NEX)	Notizen erstellen zu Präsentation und zum Fachgespräch, Bewertung der Facharbeit

3 Benützte Firmenstandards

Es werden keine spezifischen Firmenstandards verwendet.

4 Arbeitsumgebung

In diesem Kapitel ist beschrieben, wie die Arbeitsumgebung des Lernenden während der Probe-IPA aussah.

4.1 Arbeitsplatz

4.1.1 Office Arbeitsplatz

Die Probe-IPA wird am gewohnten Arbeitsplatz im Fünferbüro des Lernenden durchgeführt. Als Arbeitsgerät wird ein Notebook verwendet, welches mithilfe einer Dockingstation das Gerät mit zwei Monitoren und dem Firmennetzwerk verbindet. Der Stuhl und Tisch sind höhenverstellbar, und der Lernende kann dadurch in verschiedenen Sitzpositionen oder stehend arbeiten.

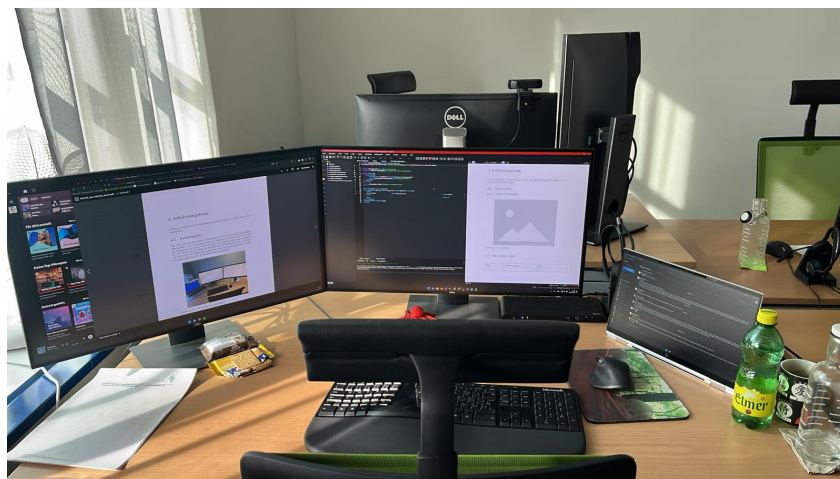


Abbildung 4.1: Arbeitsplatz des Lernenden

4.2 Verwendete Tools

Die folgende Tabelle zeigt auf, welche Tools für die Umsetzung der Probe-IPA eingesetzt wurden.

Tool	Einsatzzweck	Link
IntelliJ	Entwicklungsumgebung für die Programmierung	https://www.jetbrains.com/de-de/idea/
Docker	Ausführen der Programme	https://www.docker.com/
Git	Versionierung vom Quellcode	https://git-scm.com/
Postman	Ausführen von HTTP-Requests (Testen vom Bankend)	https://www.postman.com/
Bitbucket	Speicherung der Quellcodes	https://bitbucket.org/product/
Confluence	Probe-IPA Kriterien	https://www.atlassian.com/de/software/confluence
Jira	Aufgabenstellung	https://www.atlassian.com/de/software/jira
Draw.io	Erstellen von Diagrammen und Abbildungen	https://www.drawio.com/
TexStudio	Dokumentationstool	https://www.texstudio.org/
LaTeX	Ein Dokumentenvorbereitungssystem	https://www.latex-project.org/
Mattermost	Text basiertes Kommunikationsmittel	https://mattermost.com/
Microsoft Teams	Video basiertes Kommunikationsmittel	https://www.microsoft.com/de-ch/microsoft-teams/group-chat-software
Microsoft Excel	Erstellung und Bearbeitung des Zeitplans	https://www.microsoft.com/de-ch/microsoft-365/excel?market=ch

5 Versionierung und Sicherung der Arbeitsergebnisse

In diesem Kapitel wird beschrieben, wie der Lernende sicherstellt, dass die erarbeiteten Ergebnisse während der Probe-IPA sicher gespeichert und jederzeit wieder aufrufbar sind. Die Versionierung soll es ermöglichen, frühere erstellte Versionen der Daten jederzeit wiederherstellen zu können. Die Massnahmen werden hier vom Lernenden aufgeführt.

5.1 Verwendung von Git zu Versionierung

Für die Versionierung von der Probe-IPA wird Git verwendet. Git ist ein Versionierungstool und wird genutzt, um Quellcode zu versionieren und zu beschriften. In der Schule so wie auch in der Firma wurde Git bereits in diversen Projekten verwendet, um den Quellcode übersichtlich zu versionieren und in der Cloud zu sichern. Mit Git kann man sogenannte «Commits» machen, um einen kleinen Teil der Änderungen zu speichern und zu beschriften. Diese Änderungen kann man jederzeit wieder rückgängig machen oder aufrufen, um eine bestimmte Version genauer zu analysieren. Durch diese Commits ist der Quellcode für eine andere Person verständlicher zu lesen.

5.2 Quellcode

Der Quellcode der eingehenden Message-Queue-Nachrichten im Web-GUI wird mit Git verwaltet und ist in einem Repository auf Bitbucket gespeichert. In Abbildung 5.1 ist die Git Commit History des Quellcodes ersichtlich.



Abbildung 5.1: Git Commit History des Quellcodes

5.3 Probe-IPA Dokumentation

Die Probe-IPA-Dokumentation wird mithilfe von \LaTeX geschrieben, wodurch eine Versionierung mit Git auch möglich wird. Die \LaTeX - und alle anderen benötigten Dateien werden auf ein privates Repository in der Cloud geladen. In Abbildung 5.2 ist die Git Commit History der Probe-IPA-Dokumentation ersichtlich.



Abbildung 5.2: Git Commit History der Dokumentation

6 Projektmanagementmethode

In diesem Kapitel ist die Projektmanagement-Methode «IPERKA» beschrieben, die während der Probe-IPA verwendet wird. Es werden die Gründe für diese Projektmanagement-Methode aufgeführt und eine alternative Methode mit Gründen, warum sie nicht verwendet wurde.

6.1 IPERKA

Als Projektmanagement-Methode während der Probe-IPA wird IPERKA verwendet. IPERKA ist eine Vorgehensmethode, die sich gut für Projekte mit überschaubarem Umfang und klar definierte Ziele eignet. Die Methode wurde bereits im ersten Lehrjahr in der Schule behandelt und ist durch das bereits bekannt. Der Name «IPERKA» setzt sich aus den Anfangsbuchstaben der sechs verschiedenen Schritten zusammen, nach denen vorgegangen wird:

I nformieren: Den Auftrag verstehen, eine Vorstellung der Lösung erhalten, fehlende Informationen einholen, ordnen und bewerten

P lanen: Nötige Arbeitsschritte definieren, einen Zeitplan erstellen, Methoden und Arbeitsmittel definieren

E ntcheiden: Verschiedene Lösungsvarianten vergleichen, ausschlaggebende Kriterien definieren, eine Lösungsvariante auswählen

R ealisieren: Arbeit umsetzen, Arbeitsschritte und Ergebnisse dokumentieren, auftretende Probleme behandeln

K ontrollieren: Arbeit testen, Resultate mit den Anforderungen vergleichen, Soll-Ist-Vergleich des Zeitplans, Dokumentation nochmals durchlesen

A uswerten: Rückblick auf das Vorgehen, Arbeitsschritte beurteilen, Selbsteinschätzung vornehmen, mögliche Optimierungen für weitere Projekte definieren

Die Aufteilung der Arbeit in die genannten Schritte unterstützt dabei, die Aufgaben sinnvoll zu strukturieren und systematisch vorzugehen. Ausserdem hilft die bewusste Steuerung des Arbeitsprozesses, die persönlichen Kompetenzen weiterzuentwickeln (vgl. ICT Berufsbildung Bern 2024(Bern 2024)). Die klare Trennung der Schritte stellt sicher, dass der Umfang und die erwarteten Ergebnisse der Aufgabe sich im Verlauf der Arbeit nicht mehr stark ändern, da IPERKA grundsätzlich kein «Rückwärtsgehen» in den Phasen, wie dies aus iterativen Modellen bekannt ist, vorsieht. Ausserdem stellt sie sicher, dass die Realisierung nicht zu schnell in Angriff genommen wird.

6.2 Alternative Methode

Neben IPERKA wurde noch eine andere Vorgehensmethode angeschaut. Diese ist folgend, jeweils mit einer Begründung, wieso IPERKA der Methode vorgezogen wurde, kurz beschrieben.

Scrum ist eine bekannte agile Vorgehensmethode, die heutzutage in der Softwareentwicklung weit verbreitet ist. Die Methode legt den Fokus mehr auf Punkte wie laufende Software, gute Zusammenarbeit, oder flexibles Reagieren auf Veränderungen, anstatt einem strikten Plan zu folgen. Scrum sieht einen iterativen Prozess vor, der laufend optimiert werden soll, und definiert verschiedene Rollen, die jeweils ihre Aufgabe in diesem Scrum-Prozess haben. Ein «agiles» Vorgehen ist in der Softwareentwicklung grundsätzlich sinnvoll, da oft nicht von Beginn her klar ist, wie das Resultat schlussendlich aussehen soll. Da die Probe-IPA schlussendlich aber eine Prüfung ist, sind die Vorgaben und erwarteten Resultate relativ klar. Auch der Umfang der Probe-IPA und das Enddatum sind von Beginn an bekannt und verändern sich nicht während der Arbeit. Ausserdem wird die Aufgabe von nur einer Person bearbeitet, wofür die Abläufe von Scrum nicht optimal geeignet sind. Aus diesen Gründen wird IPERKA gegenüber Scrum vom Lernenden bevorzugt.

7 Arbeitsprotokoll

Datum	...
Bearbeitete Arbeitspakete	...
Arbeitszeit	...
Überzeit	...
Vergleich mit dem Zeitplan	...
Erfolge und Probleme	...
Tageasdfsreflexion...	
In Anspruch genommene Hilfe	...

Teil II

Projekt

Glossar

Backend Der Server-Teil eine Applikation, welcher meistens als Verbindung zwischen der Datenbank und dem Web-GUI genutzt wird. 3

CardX Eine Transaktions-Authorisierungs-Lösung von der Firma Ergon Informatik AG, die bei einigen Banken in der Schweiz im Einsatz ist. 3, 17

Ergon Informatik AG Die Firma, in der ich meine Lehre absolviere. 3, 17

UI Ein UI (User Interface) ist eine Benutzeroberfläche von einer Software oder einem Gerät um mit der Software zu Interagieren. 3

Web-GUI Eine Webseite, die genutzt wird um das Projekt CardX zu überwachen und anzupassen. 3, 17

Abbildungsverzeichnis

1	Logo der Ergon Informatik AG (Ergon Informatik AG 2024)	1
1.1	Anzeigen und bearbeiten der Tasks	5
4.1	Arbeitsplatz des Lernenden	8
5.1	Git Commit History des Quellcodes	11
5.2	Git Commit History der Dokumentation	12

Quellenverzeichnis

Bern, ICT Berufsbildung (2024). *Die 6-Schrittmethode IPERKA*. URL: https://www.ict-berufsbildung-bern.ch/resources/Iperka_OdA_200617.pdf (besucht am 06.11.2024).

Ergon Informatik AG (2024). *Logo der Ergon Informatik AG*. URL: <https://www.ergon.ch/dam/jcr:c42b10d3-a5c7-4ada-b8e5-ccc94f802da3/ergon-logo.png> (besucht am 06.11.2024).