## IPERKA Buisness-Trip

## I-Informationen beschaffen

Wir haben uns zuerst mit der Aufgabenstellung vertraut gemacht und anschliessend in der Gruppe Unklarheiten bereinigt. Dabei haben wir insbesondere darauf geachtet, was genau zur Aufgabe gehört und was nicht. Anschliessend haben wir Anforderungen und Erwartungen an die Lösung erstellt, insbesondere im Hinblick darauf, welche Funktionen sie bieten muss.

Jan hat ein GitHub-Repository für unser Projekt erstellt und den anderen Teammitgliedern Zugriff gewährt. Wir haben das Repository lokal auf unsere Geräte geklont, und jeder hat einen eigenen Branch erstellt, um effizient parallel arbeiten zu können.

Zur Informationsbeschaffung haben wir die auf Teams zu Verfügung gestellten Dokumente verwendet und mögliche Ansätze diskutiert. Wir haben überlegt, welche ähnlichen Projekte wir bereits gemacht haben und ob wir daraus Erkenntnisse übernehmen können. Zudem haben wir uns mit GitHub, insbesondere Wiki, vertraut gemacht.

Die Verantwortlichkeiten innerhalb des Teams wurden klar verteilt, sodass jeder weiss, für welchen Teil er oder sie zuständig ist. Wir haben auch definiert, wie wir den Arbeitsfortschritt dokumentieren und untereinander austauschen wollen.

## P-Planen

Wir haben zuerst alle Ideen für Use-Cases zusammengetragen und die besten Ideen aufgeschrieben:

**Kassenbons:**

Während der Reise können Mitarbeiter ihre Rechnungen im Portal eintragen, wobei sie die Kategorie und Preis der Ausgabe angeben müssen. Anschliessend werden die Ausgaben zusammengerechnet und man erhält die Ausgaben der ganzen Reise auf einen Blick.

**Packliste:**

Dem Reisenden wird direkt eine Packliste mit AI generiert, damit er weiss, was er alles mitnehmen muss. Die AI erhält Daten über die Reise, wie z-B. Reisedauer, Jahreszeit, Kontinent, etc.

**Zeitplan:**

Die Website hat einen integrierten Zeitplan, Welcher den Teilnehmer anzeigt wo sie wann einen Termin haben.

**Shared Folder:**

In einem geteilten Ordner können Mitarbeiter alle Dokumente hochladen, damit alle Zugriff auf die Dateien haben.

**Teilnehmerverwaltung**

Alle Beteiligten einer Dienstreise können sich für eine Reise anmelden und müssen dabei ihre Personaldaten und Position(z.B. Teamlead, Mitarbeiter) angeben. Die Listen aller Mitarbeiter auf der Reise kann man als Excel-Datei exportieren.

**Restaurant- und Hotel-Empfehlungen**

Die Website gibt AI-gestützte Empfehlungen für Restaurants, Cafés oder Hotels in der Nähe des Reiseziels (eventuell mit Filteroptionen (z. B. Preis, Entfernung, Küche)).

**Währungsrechner**

Integrierter Währungsrechner, der aktuelle Wechselkurse nutzt, um Ausgaben in unterschiedlichen Ländern zu vereinheitlichen. Praktisch für Reisen ins Ausland, damit alle Beträge in einer Währung angezeigt werden können.

## E-Entscheiden

Zunächst haben wir einzeln überlegt, welche Use-Cases wir gerne umsetzen würden und haben uns anschliessend unsere Meinungen untereinander ausgetauscht. Gemeinsam konnten wir uns auf 4 Use-Cases einigen, welche wir definitiv umsetzen wollen. Die 4 Use-Cases sind: *Kassenbons, Packliste, Teilnehmerverwaltung und Währungsrechner*.

Jedes Teammitglied ist für die Erarbeitung eines eigenen Use-Case zuständig. Die finalen Entschiedungen wurden immer gemeinsam getroffen und wir sind alle Kompromisse eingegangen, sodas alle einverstanden waren. Die Begründung für die Auswahl der 4 Use-Cases liegt in der Relevanz für das Grund-Programm und die Funktionalität, während sie für uns in der gegebenen Zeit umsetzbar sind. Unsere Use-Cases entsprechen also den Kriterien ‘Essenzialität für das Grundprogramm’ und ‘Machbarkeit’.

Von der Idee mit den Restaurant- und Hotelempfehlungen konnten wir uns nicht ganz lösen. Wir haben uns aber dazu entschieden, diese erst bei genügend Zeit umzusetzen, da es nicht essenziell für das Grund-Programm ist. Dieser Teilentscheid steht im Einklang mit unserem Fokus auf die Essenzialität.

## R-Realisieren

Nachdem die Aufgabenstellung klar definiert war, begannen wir motiviert mit der Programmierung unseres Projekts. Schon bald stiessen wir jedoch auf Schwierigkeiten mit der Datenbank. Ursprünglich hatten wir geplant, eine Maven-basierte Datenbanklösung zu verwenden. Leider merkten wir schnell, dass es schwieriger war als erwartet, da wir uns alle nicht sicher waren, wie diese funktioniert. Anfangs versuchten wir, das Problem mit Hilfe von Künstlicher Intelligenz zu lösen, indem wir unser Programm entsprechend umschreiben liessen. Doch durch die automatisch generierten Änderungen wurde der Code unübersichtlich und für uns auch schwer nachvollziehbar.

Nach einigen frustrierenden Versuchen und gemeinsamen Besprechungen trafen wir schliesslich die Entscheidung, das Projekt von Grund auf neu zu beginnen. Uns war bewusst, dass dieser Schritt zusätzlichen Zeitdruck mit sich bringen würde, doch wir waren überzeugt, dass ein klar strukturierter Neuanfang die sinnvollste Lösung war.

Für unseren zweiten Versuch entschieden wir uns für eine in Java integrierte SQL-Datenbank, die wir bereits in einem ÜK kennengelernt hatten. Dadurch hatten wir ein solides Grundverständnis und konnten sie problemlos und effizient in unser Projekt einbinden.

Die Aufgabenzuteilung innerhalb des Teams blieb unverändert, sodass jeder weiterhin für seinen Bereich verantwortlich war. Während der Entwicklung achteten wir darauf, regelmässig Zwischenstände in Git zu pushen. So konnten alle Teammitglieder jederzeit auf dem aktuellen Stand bleiben und nahtlos an den bereits bestehenden Teilen weiterarbeiten.

Im Verlauf der Entwicklung haben wir uns immer wieder gegenseitig unterstützt, wenn jemand bei seinem Use-Case nicht weiterkam. Wir haben uns regelmässig ausgetauscht, um offene Fragen zu klären und gemeinsam Lösungen zu finden. Auch kleinere Anpassungen an den Use-Cases wurden im Team besprochen, damit alles zusammenpasst.

Die Entwicklung erfolgte getrennt im Frontend und Backend. Wir wollten uns auf eine saubere Trennung zwischen Frontend und Backend achten und deshalb entschieden wir uns für die schrittweise Entwicklung. Für das Backend haben wir Java mit Spring Boot verwendet und eine SQL-Datenbank eingebunden. Im Frontend haben wir mit React gearbeitet. Die Schnittstellen zwischen Frontend und Backend haben wir mit REST-APIs umgesetzt.

Am Ende haben wir die verschiedenen Teile zusammengeführt und getestet, ob alles wie geplant funktioniert. Dabei haben wir auch darauf geachtet, dass die Daten korrekt zwischen Frontend und Backend übertragen werden.

## K-Kontrollieren

Nachdem die Hauptfunktionen fertiggestellt waren, haben wir die Anwendung gemeinsam getestet. Jeder hat die Use-Cases der anderen ausprobiert, um Fehler zu finden und Verbesserungsvorschläge zu machen. Wir haben dabei besonders darauf geachtet, dass die Funktionen wie das Eintragen von Kassenbons, die Packliste, die Teilnehmerverwaltung und der Währungsrechner reibungslos laufen.

Wir haben verschiedene Tests durchgeführt, zum Beispiel ob die Anmeldung funktioniert, ob die Packliste sinnvoll generiert wird und ob die Ausgaben korrekt berechnet werden. Auch die Benutzeroberfläche haben wir überprüft, damit sie übersichtlich und einfach zu bedienen ist.

Weiter mussten wird die Berechtigungen der User genau betrachten. Hier ist das Wichtigste, dass nur der Teamleader alle Kassenbons sieht. Die normalen Mitarbeiter sollten dies nur von sich selbst sehen. Hier gab es zu erst einige Probleme, welche jedoch mit der richtigen SQL-Query richtig angepasst werden konnten.

## A-Auswerten

Zum Schluss haben wir im Team besprochen, was gut gelaufen ist und wo wir noch Verbesserungspotenzial sehen. Wir waren zufrieden, dass wir trotz Startschwierigkeiten ein funktionierendes Programm auf die Beine gestellt haben. Besonders hilfreich waren die klare Aufgabenverteilung und die regelmässige Kommunikation im Team.

Wir haben festgestellt, dass es sich gelohnt hat, frühzeitig die wichtigsten Use-Cases festzulegen und die Arbeitsschritte zu planen. Die Nutzung von GitHub für die Versionskontrolle und Zusammenarbeit hat uns viel geholfen.

Für die Zukunft haben wir uns vorgenommen, noch mehr Zeit für das Testen einzuplanen und bei Problemen schneller gemeinsam nach Lösungen zu suchen. Ausserdem möchten wir beim nächsten Mal die Dokumentation noch während der Entwicklung laufend ergänzen, damit am Ende nichts vergessen geht. Insgesamt sind wir mit dem Ergebnis zufrieden und haben viel dazugelernt.