

Seminararbeit

Fahrradkurier Radius - Fahrtenbücher

Projektseminar „Informatik und Optimierung für Nachhaltigkeit“

Flavian Raithel

Martin Kammleiter

Nico Beckers

Semester: Sommersemester 2023

Abgabedatum: 30. Juli 2023

Betreuer: Prof. Dr. Marie Schmidt



Julius-Maximilians-Universität Würzburg
Lehrstuhl für Informatik I
Algorithmen und Komplexität

Inhaltsverzeichnis

1	Abstract	3
2	Einleitung	4
2.1	Kontext	4
2.2	Projektziel(e)	4
3	Projektarbeit	5
3.1	Koordination in der Gruppe	5
3.2	Kontakt und Interaktion mit den Projektstellern	5
3.3	Einordnung	5
3.4	Problemstellung	6
3.5	Zeitlicher Ablauf	6
3.6	Vorgehen	9
3.7	Probleme im Verlauf des Projektes - und Lösungsansätze	18
3.8	Beiträge der Gruppenmitglieder	21
4	Projektergebnisse und Diskussion	22
4.1	Deliverables/Resultate	22
4.1.1	Übersicht	22
4.1.2	Bedienungsanleitung	22
4.1.3	Formular Sonderfahrten	24
4.1.4	Skript Sonderfahrten	24
4.1.5	Skript Testdatengenerierung	25
4.1.6	Skript Filterfunktion	26
4.2	Diskussion	28
4.3	Wie könnte es weitergehen?	32
5	Anhang	32

1 Abstract

Dieser Bericht behandelt das Projekt Fahrtenbücher, welches in Kooperation mit dem Fahrradkurierunternehmen Radius im Rahmen des Projektseminars an der Universität Würzburg im Sommersemester 2023 stattfand. Dabei bieten Unternehmen, mit Nachhaltigkeitsbezug, Studierenden ein Projekt im Bereich Informatik. In diesem Fall die Erstellung eines digitalen Fahrtenbuchs. Radius ist ein kleines Unternehmen aus Würzburg ohne IT-Abteilung oder fundierten technischen Kenntnissen. Ihr Anliegen ist ein System, dass die Ein- und Ausgabe von Fahrten erleichtert. Bislang wurden Fahrten analog dokumentiert oder manuell in einer Google Sheets Tabelle eingetragen. Dabei kam es häufig zu Eingabefehlern, gerade bei der Benutzung von mobilen Endgeräten. Im Laufe des Projektes gestalteten wir, zusammen mit Radius, ein Google Forms Formular, dass die Fahrten eines Arbeitstags in einem geführten Prozess abfragt und diese automatisch in eine Google Sheets Tabelle abspeichert. Dort lässt sich eine visuell aufbereitete, prägnante Monatsansicht aus den Daten erstellen, die mittels Filter- und Statistikfunktionen schnell ausgewertet werden kann. Dies geschieht mittels einem von uns geschriebenen, in die Tabelle integrierten Makro. Das Formular ist dabei im Browser ausfüllbar und für mobilen Eingabe optimiert. Testungen im Betrieb ergaben eine erfolgreiche Nutzung. Bei Übergabe des Projekts gab Radius an das System auch zukünftig verwenden zu wollen.

2 Einleitung

Das Projektseminar „Fahrradkurier Radius - Fahrtenbücher“ verfolgt das Ziel, dass aktuelle Fahrtenbuch der Fahrradkuriere von Radius weniger fehleranfällig und pflegeleichter zu machen. Radius ist ein Fahrradkurier, welcher den Transport und die Zustellung von kleineren und auch größeren Paketen mit dem Fahrrad in der Stadt Würzburg für verschiedenste Kunden durchführt. Die Fahrradkuriere arbeiten sehr umweltfreundlich, da Pakete sonst mit dem Auto/Bus/LKW ausgeliefert werden müssten. Außerdem arbeiten sie nach den Prinzipien der Gemeinwohl-Ökonomie und sind damit nicht nur aus Umweltaspekten, sondern auch aus sozialen Aspekten ein nachhaltiges Unternehmen. Die soziale Nachhaltigkeit zeigt sich beispielsweise an der flachen Hierarchie, dem großen Mitspracherecht bei wichtigen Entscheidungen für alle Mitarbeiter und einem geringen Lohnunterschied.

2.1 Kontext

Bei den Fahrradkurieren von Radius wird noch recht viel Arbeit „händisch“ und nicht automatisiert erledigt. Dies zeigt sich schon bei der Entgegennahme von spontanen Auslieferungsaufträgen oder bei der Routenplanung. Außerdem werden spontane Fahrten noch handschriftlich auf einem Fahrtenbuch-Block festgehalten. Die Eintragung der gefahrenen Fahrten pro Schicht erfolgt durch die Kuriere in eine Google-Sheets-Tabelle. Diese Tabelle wird vom Büro für die Rechnungserstellung und zur Überprüfung von Fahrten genutzt. In der Tabelle werden allerdings in einzelnen Zellen Angaben über recht viele verschiedene Fahrten gemacht, was unübersichtlich und unpraktisch ist. Außerdem kann die Freitext-Eingabe leicht zu Fehlern führen und die Einträge von Kollegen könnten versehentlich überschrieben werden. Zusätzlich ist diese Art der Fahrtenbuchführung von einem Smartphone aus sehr unpraktisch.

2.2 Projektziel(e)

Das Projektziel besteht darin, die Pflege des Fahrtenbuches einfacher, schneller und weniger fehleranfällig zu machen. Des Weiteren ist das Ziel ein Fahrtenbuch an die Kuriere von Radius zu übergeben, welches robust und so einfach wie möglich gehalten ist, da nach Projektende nicht mehr am Fahrtenbuch gearbeitet wird. Es sollen aber alle nötigen Infos übergeben werden und die durchgeführten Schritte so gut wie möglich erklärt werden, sodass bei Problemen in der Zukunft eine Lösung durch Radius selber gefunden werden kann. Daher ist auch eine Bedienungsanleitung und ausführliche Einarbeitung der Kuriere ein weiteres Projektziel.

3 Projektarbeit

3.1 Koordination in der Gruppe

Die Koordination innerhalb der Gruppe erfolgte sehr dynamisch und durch häufige Absprachen. Dazu gehörten meist Treffen in den Universitätsgebäuden, aber bei Bedarf auch online, in welchen der aktuelle Fortschritt und das weitere Vorgehen besprochen wurden. Die Aufgabenaufteilung war flexibel und wurde stets überdacht und neu festgelegt. Die Gruppe hat so zusammengearbeitet, dass neue Erkenntnisse und Fortschritte schnell mit allen Gruppenmitgliedern geteilt wurden, sodass man sich bei Problemen gegenseitig unterstützen konnte. Von den etablierten Entwicklungsmodellen spiegelt das Agile Entwicklungsmodell unsere Arbeit innerhalb der Gruppe am besten wieder, wobei durch die kleine Gruppe kein sonst typischer Scrum-Master oder Projektleiter benötigt wurde. Entscheidungen konnten durch eine Diskussion und gemeinsames Abwägen von Vor- und Nachteilen immer gut getroffen werden, auch wenn man bei einzelnen Entscheidungen Kompromisse eingehen musste.

3.2 Kontakt und Interaktion mit den Projektstellern

Mit den Projektstellern von Radius wurde sich regelmäßig getroffen. Zu Beginn fanden diese Treffen in den Universitätsgebäuden statt. Später wurden die Treffen im Büro von Radius durchgeführt. Die Treffen im Büro von Radius hatten den Vorteil, dass wir das Unternehmen besser kennenlernen konnten und uns so besser in deren Arbeitsalltag denken konnten. Die Atmosphäre und Stimmung bei den Treffen war stets offen und harmonisch. Der Austausch über Ziele, Änderungswünsche und Probleme lief ohne Probleme. Die Treffen vor Ort haben je nach Projektabschnitt und Besprechungsbedarf alle 1 - 2 Wochen stattgefunden und haben meist zwischen 1,5 und 2 Stunden gedauert. Für kleinere Anliegen und um die Treffen vor Ort zu organisieren wurde E-Mail benutzt. Des Weiteren wurden auch die Zwischenpräsentationen genutzt, um sich über den aktuellen Stand des Projekts auszutauschen und ein nächstes Treffen zu vereinbaren. Es gab zu keinem Zeitpunkt größere Schwierigkeiten einen passenden Termin zu finden.

3.3 Einordnung

Als nützliche Vorerfahrung diente das Softwarepraktikum, das alle Teilnehmer vor oder während des Projekts im Rahmen der Universität absolvierten. Dabei geht es um ein Softwareprojekt, das in einer Gruppe, größtenteils eigenständig bearbeitet wird. Eine essentielle Basis, für den Verlauf des Projektes, bot Google und deren kostenloses Portfolio an online Diensten. Vor allem Google Sheets und Forms. Ohne diese Dienste wäre das Projekt so nicht realisierbar gewesen.

3.4 Problemstellung

Die Problemstellung besteht im Wesentlichen darin, ein System zu liefern, das einfach zu handhaben ist. Explizit die Wartung und Instandhaltung. Radius ist ein kleines Unternehmen ohne IT-Abteilung, geschweige denn dediziertem Know-How in diesem Bereich. Es ist nicht davon auszugehen, dass mit technisch anspruchsvolleren Problemen umgegangen werden kann. Das System sollte dabei zwei Interessengruppen bedienen, zum einen die administrative Seite, also die Personen bei Radius, die sich um die Verwaltung kümmern, zum anderen Kuriere, die vorwiegend nur Fahrten eintragen. Konkret muss das System eine unkomplizierte, Fehler minimierende Eingabe für die Kuriere und eine aussagekräftige, prägnante Ausgabe für die Administratoren bieten. Wobei die Handhabung stets ohne tiefer gehendes Vorwissen problemlos möglich sein sollte. Ausgeschlossen war die weitere Datenverarbeitung, das bedeutet explizit keine automatische Rechnungserstellung oder Datenauswertung. Das System beschränkt sich auf die reine Ein- und Ausgabe der Fahrten.

3.5 Zeitlicher Ablauf

Der Ablauf des Projekts als Zeitstrahl. Dargestellt sind alle Treffen im Zusammenhang mit dem Seminar. Die Farben haben die folgende Bedeutung:

- Blaue Felder sind Treffen mit allen Teilnehmenden des Seminars.
- Organe Felder sind Treffen mit Team Radius.
- Grüne Felder sind Treffen in der Gruppe oder zu der Projektsprechstunde.
- In den grauen Zeiträumen fand eine individuelle Bearbeitung statt.

18.4	Einführungsveranstaltung
19.4	
20.4	
21.4	
22.4	
23.4	
24.4	
25.4	Erstes Treffen in der Gruppe/mit Radius; Erstellung Projektcanvas
26.4	
27.4	Treffen (Gruppe, Präsenz): Diskussion über Lösungsansätze; Arbeitsteilung für Recherche
28.4	
29.4	
30.4	April
01.5	Mai
02.5	Erste Projektsprechstunde und anschließendes Treffen (Gruppe, Präsenz): Recherche Ergebnisse zusammengetragen; Google Sheets und Forms gewählt
03.5	
04.5	Erstes Treffen im Betrieb Radius: Gemeinsame Erarbeitung der Problemstellung und Vorstellungen einer Lösung; Tiefgreifende geteilte Recherche über Google Sheets und Forms, um Machbarkeit der erarbeiteten Vorstellungen zu prüfen
05.5	
06.5	
07.5	
08.5	
09.5	
10.5	
11.5	
12.5	
13.5	Treffen (Gruppe, Online): Zusammentragen der Recherche Ergebnisse; Erste Formularprototypen testen; Vorbereitung der ersten Zwischenpräsentation
14.5	Treffen (Gruppe, Online): Probevortrag
15.5	Erste Zwischenpräsentation: Im Anschluss Rücksprache mit Team Radius; Übergabe erster Formularprototypen
16.5	
17.5	
18.5	
19.5	
20.5	
21.5	
22.5	
23.5	Treffen (Gruppe, Präsenz): Überarbeitung des Prototyps anhand von Feedback von Radius
24.5	
25.5	Treffen im Betrieb Radius: Vorstellung überarbeiteter Prototyp; Absprache über Darstellung der Tabelle in Google Sheets; Beginn Erstellung Makro für Sheets
26.5	
27.5	
28.5	
29.5	Treffen (Gruppe, Online): Gemeinsame Arbeit am Code; Aufteilung von Codeabschnitten/Recherche
30.5	Treffen (Gruppe, Online): Gemeinsame Arbeit am Code
31.5	Mai
01.6	Juni
02.6	
03.6	

04.6	Treffen (Gruppe, Online): Gemeinsame Arbeit am Code; Durchführung mit synthetisch erstellten Testdaten
05.6	
06.6	
07.6	
08.6	Treffen (Gruppe, Online): Vorbereitung zweite Zwischenpräsentation
09.6	
10.6	
11.6	Treffen (Gruppe, Online): Probevortrag
12.6	Zweite Zwischenpräsentation
13.6	Zweite Projektsprechstunde und anschließendes Treffen (Gruppe, Präsenz): Reflexion über bisherigen Verlauf; Rücksprache über restlichen Verlauf und Prüfungsleistungen
14.6	
15.6	Treffen im Betrieb Radius: Einrichtung des Projekts auf Firmenrechner von Radius; Beginn kleine Testphase (nur Admins tragen Fahrten ein)
16.6	
17.6	
18.6	
19.6	
20.6	
21.6	
22.6	Treffen im Betrieb Radius: Einarbeitung von Feedback der bisherigen Testphase; Einholen von weiterem Feedback
23.6	
24.6	
25.6	
26.6	
27.6	
28.6	
29.6	Treffen (Gruppe, Präsenz): Aufteilung Projektbericht, Fehlerbehebung und Verbesserung, Einarbeiten von Feedback
30.6	Juni
01.7	Juli
02.7	
03.7	
04.7	
05.7	Treffen (Gruppe, Online): Arbeit am Projektbericht und Bedienungsanleitung, Einfügen neuer Filter Möglichkeiten
06.7	Treffen im Betrieb Radius: Besprechung kleine Testphase; Sammeln von Feedback; Beginn große Testphase (Kuriere tragen Fahrten ein)
07.7	
08.7	
09.7	Treffen (Gruppe, Online): Probevortrag
10.7	Abschlusspräsentation
11.7	
12.7	
13.7	
14.7	
15.7	
16.7	
17.7	
18.7	Finales Treffen bei Radius: Übergabe von Projekt und Bedienungsanleitung

3.6 Vorgehen

Zu Beginn des Projektes fokussierten wir uns darauf herauszufinden was genau benötigt wird. Dazu haben wir uns mehrfach mit Team Radius getroffen und sukzessive eine genaue Zielsetzung für das Projekt erarbeitet. Diese wurde im weiteren Verlauf, je nach aktuellem Stand angepasst und aktualisiert. Am Anfang ging es in erster Linie darum die Eingabe für die Fahrer und Fahrerinnen zu erleichtern. Nach den Schilderungen von Radius kam es an dieser Stelle hauptsächlich zu Fehlern. Deshalb haben wir uns gemeinsam darauf verständigt, eine Art geführten Prozess zum Eintragen der Fahrten zu nutzen. Im besten Fall sollte unsere Wahl direkt, das Speichern und Weiterverarbeiten der Eingetragenen Daten ermöglichen. Die Recherche dazu ergab eine webbasierte Lösung als gute Möglichkeit. So können die Fahrten unabhängig von installierter Software (ein Browser mit Internetzugang vorausgesetzt) eingetragen werden. Vor allem aus einer praktischen Perspektive, im Kontext eines Fahrradkuriers, ist die Möglichkeit Fahrten auch über mobile Geräte eintragen zu können, wichtig. Das ist mit der webbasierten Lösung problemlos möglich. Für die Umsetzung ergaben sich drei Varianten:

- Variante 1: Eine eigene Website kreieren mit selbst gehosteten oder gemieteten Server im Backend. Die Fahrtenbuch-Daten in einer eigenen Datenbank verwalten. Auch wenn diese Variante uns den größten Freiheitsgrad gewährt hätte, haben wir uns schnell dagegen entschieden. Ein solches Vorgehen wäre überproportional aufwändig und für ein kleines Unternehmen, wie Radius nicht verhältnismäßig. Gerade in Hinblick auf zukünftige Wartung und Kosten.
- Variante 2: Bisher verwendet Radius die Google Drive Plattform als Online-Speicher für ihre Fahrten. Deshalb lag es nahe weiterhin Google Dienste als bekanntes Umfeld zu nutzen. Zumal Google eine Vielzahl an online verfügbaren Diensten bietet. Bei der Recherche sind wir auf Google AppSheet gestoßen, eine No-Code Entwicklungsplattform, die Google Dienste integriert. Allerdings haben wir uns auch gegen diese Variante entschieden, da eine Applikation für mehrere Benutzer mit eventuellen Kosten verbunden wäre. Obwohl AppSheet auf No-Code basiert, wären in Zukunft Fehler nur mit einer gewissen Einarbeitung und auch technischen Verständnis zu beheben, was für Radius suboptimal wäre.
- Variante 3: Wir haben uns intensiv mit Google Sheets befasst und sind auf die Möglichkeit gestoßen Tabellen mit Google Forms zu verknüpfen. Google Forms ist ein Google Dienst, der die eigene Erstellung von Fragebögen ermöglicht. Zum einen haben wir so einen geführten Prozess für die Fahrteneintragung, zum anderen werden die Daten direkt in das bereits bekannte Sheets gespeichert. Bei ersten Tests zeigte sich, dass die Erstellung und Bearbeitung von Formularen intuitiv und einfach gestaltet ist. Was für das spätere Arbeiten und Warten von Radius, ohne Betreuung, von großem Vorteil ist. Insgesamt schien diese Variante, für Radius mit verhältnismäßig wenig Aufwand verbunden zu sein.

Mit der Festlegung auf Variante 3, haben wir uns der vertieften Recherche von Google Sheets und Forms gewidmet. Der Zusammenhang zwischen den beiden Diensten, besteht

The image shows two parts of a user interface. On the left is a Google Form titled 'Fahrtenbuch'. It has a header with a login link and a red error message: '* Gibt eine erforderliche Frage an'. Below are input fields for 'Fahrernummer' (with a dropdown showing '11'), 'Fahrscheinnummer' (with a text input '12p345' and a red error message 'Ungültige Fahrscheinnummer'), and 'Datum und Uhrzeit' (with a date picker for '15.05.2023' and a time picker for '12:15'). At the bottom, there is a 'Weiter' button, a progress bar, and a 'Seite 1 von 5' indicator. On the right is a table titled 'Fahrten am Dienstag'. The table has columns: 'Auswahl: *', 'Welche der Fahrten haben stattgefunden?', 'Gefahren', 'Nicht Gefahren', 'Abgesagt', 'Cargo', and 'Mit Rückfahrt'. The rows list various activities with checkboxes in each column.

Auswahl: *	Welche der Fahrten haben stattgefunden?	Gefahren	Nicht Gefahren	Abgesagt	Cargo	Mit Rückfahrt
Geburtshaus		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Dom 9:45		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dom 11:45		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Dom 13:30		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dom 16:45		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FHWS		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DGB		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ökokisten		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
KKH		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Heine		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FH		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
KÄ		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Abb. 1: Auszug aus dem Prototypen zur ersten Zwischenpräsentation.

im Wesentlichen darin, dass für Fragen im Formular jeweils Spalten in der Sheets Tabelle angelegt werden. Wird ein Formular ausgefüllt und abgeschickt wird in der Tabelle eine neue Zeile, mit zu den Spalten, beziehungsweise Fragen, korrespondierenden Antworten, gefüllt. Daraus ergab sich eine Zweiteilung des Projekts, einmal die Sinnvolle Darstellung, der vom Formular gelieferten Daten in Sheets, sowie die Erstellung eines Formulars, welches die nötigen Details für die Fahrten abfragt. Es war allerdings schnell klar, dass die Darstellung in Sheets, ohne die Fragen im Formular und damit die Struktur der Daten in der Tabelle zu kennen schwierig machbar ist. Deshalb fokussierten wir uns auf die baldige Erstellung eines Formulars (siehe Abbildung 1), dass von Radius getestet werden konnte und somit erste Daten und Feedback generierte. Diesen Prozess haben wir mit Radius durchgeführt und wiederholt bis das Formular eine Form erreicht hat mit dem beide Seiten zufrieden waren. Das Formular hat nun die folgende Struktur:

- **Abschnitt 1** (Abbildung 2a): Es wird die Fahrernummer und ein optionales Datum abgefragt, für den Fall, dass eine Fahrt nachgetragen werden muss.
- **Abschnitt 2** (Abbildung 2a): Auswahl des Wochentages, damit es nicht notwendig ist durch alle Fahrten zu gehen und direkt eine Eingrenzung auf die relevanten Fahrten erfolgt. In der Regel werden die Fahrten jeden Tag verbucht und deshalb sind nur die Fahrten eines Wochentages von Interesse. Außerdem wird so natürlich auch Zeit gespart.

- **Abschnitt 3** (Abbildung 2c): Ausgewählter Wochentag mit den stattfindenden Fahrten. Die Fahrten sind chronologisch geordnet, so kann die Eintragung analog zum Tagesverlauf durchgeführt werden. Das hilft beim Finden einzelner Fahrten und reduziert Fehler bei der Eingabe. In erster Linie bei Fehlern der Art „Fahrt vergessen einzutragen“ oder „Fahrt übersehen“. Die Fragen selbst sind wie folgt aufgebaut: Fragentitel, Beschreibung, spezifische Auswahlkästchen. Die Beschreibung ermöglicht eine genauere Fragestellung oder die Hervorhebung bestimmter Besonderheiten. Zum Beispiel: „Inklusive Zwischenstopp Hauptstraße 25b!“ oder „Wichtig! Lastenrad nur ab 30kg!“. Die Auswahlkästchen können je nach Fahrt festgelegt werden. Dieser Ansatz ermöglicht eine flexible Fragestellung, was für die teilweise sehr heterogene Datenabfrage von Vorteil ist.
- **Abschnitt 4** (Abbildung 2d): Gewährt die Möglichkeit Anmerkungen zu verfassen. Diese werden hauptsächlich für außerplanmäßige Fahrten genutzt. Während der Entwicklung des Formulars gab es zunächst ein dediziertes Sonderfahrten Formular, welches sich allerdings nach den ersten Tests als unpraktisch erwiesen hat (mehr dazu in 4.1.3 Formular Sonderfahrten). Deshalb werden die Sonderfahrten jetzt am Ende des Formulars als Anmerkung vermerkt.

Wie oben erwähnt wird bei der Speicherung des Formulars eine neue Zeile in der Tabelle befüllt. Allerdings führt der beschriebene Aufbau des Formulars dazu, dass für jeden Wochentag, für jede Frage eine separate Spalte existiert. Das ergibt eine wie in Abbildung 3 dargestellte Rohdatentabelle. Obwohl es natürlich gleiche Fahrten gibt, die schlichtweg an verschiedenen Tagen stattfinden. Das resultiert in einer Anzahl der Fragen große Spaltenanzahl, zum Zeitpunkt der Erstellung dieses Berichtes sind es beispielsweise fast 90 Spalten. Diese Zahl schwankt je nachdem wie viele Fahrten Radius fährt. Auch die Zahl der Zeilen, also Einträge von Fahrten, wächst täglich um ungefähr 5-10. Das Gesamtbild der Tabelle ist dementsprechend unübersichtlich und wenig hilfreich für die weitere Arbeit mit den Daten.

Deshalb haben wir uns nun auf eine bessere Darstellung fokussiert. Ziel war es eine möglichst, mit den nativen Funktionen von Sheets, umsetzbare Lösung zu finden. Ein Ansatz war über die, zum Beispiel aus Excel bekannten, Formeln zu gehen. Diese bieten, neben arithmetischen Operationen, beispielsweise auch die Möglichkeit Inhalte von Zellen zu kopieren und an andere Stelle einzufügen. Anweisungen müssen stets beginnend mit = in einer Zelle geschrieben werden. Auch wenn es möglich ist Formeln selbst zu definieren, zu benennen und wieder aufzurufen, ist es äußerst schwierig und unübersichtlich komplexere Programme zu schreiben. Dafür ist diese Funktion einfach nicht gedacht. Mit Hinblick auf die spätere Verwendung durch Radius ist dieser Ansatz nicht sinnvoll. Des Weiteren sind wir auf Pivot-Tabellen gestoßen. Diese ermöglichen es Ansichten zur Datenauswertung zu erstellen. Allerdings müssen händisch die gewünschten Spalten gewählt werden und es können nicht zwei gleichnamige Spalten (gleiche Fahrten, jedoch an verschiedenen Tagen) zusammengefasst werden. Dieses Problem lässt sich damit nicht lösen. Außerdem ist es schwierig eine sinnvolle Struktur für die Pivot-Tabelle zu finden. Zum Beispiel ha-

Fahrtenbuch

fl.raithel@gmail.com [Konto wechseln](#)

Nicht freigegeben

* Gibt eine erforderliche Frage an

Fahrernummer *

Bitte Fahrernummer eingeben.

23

Datum (optional)

An welchem Datum war die Fahrt?

Nur notwendig, wenn Fahrt nicht heute stattfand

Datum

tt.mm.jjjj

[Weiter](#) [Alle Eingaben löschen](#)

Geben Sie niemals Passwörter über Google Formulare weiter.
Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

(a) Abfrage Fahrernummer und optionales Datum

Fahrtenbuch

fl.raithel@gmail.com [Konto wechseln](#)

Nicht freigegeben

* Gibt eine erforderliche Frage an

Terminierte Fahrten

Wochentag *

An welchem Wochentag fand die Fahrt statt?

Montag

Dienstag

Mittwoch

Donnerstag

Freitag

Samstag

[Zurück](#) [Weiter](#) [Alle Eingaben löschen](#)

Geben Sie niemals Passwörter über Google Formulare weiter.
Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

(b) Auswahl des Wochentages

Fahrtenbuch

fl.raithel@gmail.com [Konto wechseln](#)

Nicht freigegeben

Fahrten am **Dienstag**

Frühschicht

Fahrten vor 12:00

Dorm 9:45

Gefahren

Abgesagt

Sonstiges: _____

Dorm 11:45

Gefahren

Abgesagt

Sonstiges: _____

(c) Fahrten eines Wochentags (Ausschnitt)

Fahrtenbuch

fl.raithel@gmail.com [Konto wechseln](#)

Nicht freigegeben

Ende

Anmerkungen

Gibt es noch Anmerkungen?

Meine Antwort

[Zurück](#) [Senden](#) [Alle Eingaben löschen](#)

Geben Sie niemals Passwörter über Google Formulare weiter.
Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google Formulare

(d) Anmerkungen für Sonderfahrten

Abb. 2: Abschnitte des aktuellen Formulars.

ben manche Fahrten, neben ihrem Zeitstempel, ein optionales Datum, was schon für eine einfache chronologische Gliederung problematisch ist.

Das hat uns zu Makros geführt. Makros sind, zum Beispiel aus Excel, bekannt als kleine Hilfsprogramme. Sheets bietet die Option AppScript zu verwenden, eine Google Entwicklungsumgebung, die sich in den meisten Google Diensten findet. Dort können Skripte geschrieben werden, die Dienst-spezifische Frameworks und Libraries nutzen können. Die zugrunde liegende Programmiersprache ist Java-Script, welche es ermöglicht auch kom-

Abb. 3: Rohdaten in Sheets. Hervorgehoben Fragen an einem Wochentag.

plexe Programme problemlos zu erstellen. Nach weiteren Recherchen zur Verwendung mit Sheets, begannen wir mit der Implementierung erster Skripte. Zu Beginn haben wir ein einfaches Programm geschrieben, welches die Fahrten nach dem Wochentag filtert und nur Fahrten übernimmt, die auch an diesem Wochentag stattfanden. Das reduziert die Spaltenanzahl auf die Fahrten eines Tages, bei ursprünglich 90 bleiben ca. 15-20 Spalten, was in einer Tabelle überschaubar darstellbar ist. Im Gespräch mit Radius wurde allerdings klar, dass es bessere Aufteilungen der Fahrten gibt. Radius nutzt die Dokumentation der Fahrten in erster Linie zur Rechnungsstellung. Eine Filterung nach Wochentag ist für die Rechnungsstellung suboptimal, da ein Kunde durchaus an verschiedenen Tagen Fahrten haben kann.

Wir haben uns auf eine Monatsansicht geeinigt, da die Rechnungsstellung monatlich erfolgt. Das Skript dafür, hat die folgende Funktionsweise:

- Gehe alle Zeitstempel (die Ausfüllzeitpunkte) durch und ersetze diese, falls vorhanden, durch optionales Datum.
- Filtere die Zeilen, deren Zeitstempel dem gewünschten Monat und Jahr entspricht.
- Konstruiere Datenstruktur, welche die Indices der Spalten erfasst und gleichnamige Spalten denselben Index zuweist.
- Sortiere Spalten die Fahrten beinhalten lexikographisch.
- Gruppier Zeilen nach Wochentag.
- Füge Zeilen, mittels Datenstruktur und Gruppierung in richtiger Reihenfolge und mit zusammengelegten Spalten in die Ausgabetablelle.
- Färbe Zeilen mit selbem Wochentag einheitlich.
- Ziehe Trennstriche zwischen den Wochentagen.

Die so entstehende Ansicht (Abbildung 4), soll einen sofortigen visuellen Überblick des Monats ermöglichen. Dazu sollen Wochentage klar voneinander unterscheidbar sein. Aber auch schnell wieder erkannt werden, deshalb bleibt die Farbe für einen Tag gleich. Idee ist es, bei wiederholter Arbeit mit dem System, Assoziationen zwischen den Farben und Wochentagen zu schaffen. Nach dem Schema: hellrot → Montag oder dunkelgrau → Sonntag, neue Woche. Da es an einem Wochentag meistens mehrere Fahrten gibt, wird hier zusätzlich zur Färbung visuell gruppiert. Zwischen Fahrten gleichen Wochentags

sind die Linien der Zellen schmal und hellgrau. Zwischen verscheidenden Wochentagen sind dicke schwarze Linie gezogen. Die Kopfzeile mit den Spaltentiteln hat weiße Schrift auf schwarzem Grund, damit ist diese einfach zu finden und klar vom Inhalt der Tabelle getrennt. Außerdem ist diese fixiert, was bewirkt, dass auch bei Scrollen in der Tabelle die Spaltentitel immer angezeigt werden.

Zeitstempel	Wochentag	Fahrernummer	Anmerkungen	Apotheke	Dom 09:45	Dom 11:45	Dom 13:30	Dom 16:45	TH	KKH	KÄ
01.07.2023	Samstag	191			Gefahren						
02.07.2023	Sonntag										
03.07.2023	Montag	32			Gefahren						Gefahren
03.07.2023	Montag	173			Gefahren	Cargo			Gefahren		
03.07.2023	Montag	157	final_kommentar_2	Gefahren				Gefahren			
03.07.2023	Montag	62						Gefahren		Gefahren	
04.07.2023	Dienstag	131	final_kommentar_3								
04.07.2023	Dienstag	101	final_kommentar_3		Gefahren						
04.07.2023	Dienstag	128		Gefahren							
04.07.2023	Dienstag	136					Gefahren				Abgesagt
04.07.2023	Dienstag	190	final_kommentar_3							+ Reture	
04.07.2023	Dienstag	74						Abgesagt	Gefahren		
04.07.2023	Dienstag	157		Gefahren							
05.07.2023	Mittwoch	87	final_kommentar_3								
05.07.2023	Mittwoch	141		Gefahren				Cargo			
05.07.2023	Mittwoch	89		Gefahren							
06.07.2023	Donnerstag	164					Abgesagt				
06.07.2023	Donnerstag	125		Gefahren					Cargo		
06.07.2023	Donnerstag	154	final_kommentar_3	Cargo	Gefahren						+ Reture
06.07.2023	Donnerstag	120	final_kommentar_2								
07.07.2023	Freitag	197	final_kommentar_1		Gefahren					+ Reture	
07.07.2023	Freitag	54	final_kommentar_1	Abgesagt							
07.07.2023	Freitag	182				Gefahren					

Abb. 4: Die vom Programm generierte Monatsansicht (Ausschnitt Testdaten).

Des Weiteren war es von Radius gewünscht, bestimmte Einträge farblich zu markieren. Mit dem Hintergrund schnell eine Übersicht bekommen zu können. Hier haben wir zunächst im Skript Zellen nach Wörtern wie „Gefahren“ oder „Abgesagt“, durchsucht und diese dann grün oder rot hinterlegt. Dieser Ansatz stellte sich jedoch, in der internen Testphase, als zu unflexibel heraus, da ein „hardcoding“ im Skript keine spätere Änderung durch Radius zulässt. So sollten Cargo und Retoure als weitere Hervorhebungen hinzukommen. Um das möglichst einfach zu lösen hat sich bedingte Formatierung als gute Möglichkeit ergeben. Bedingte Formatierung ist eine native Sheets Funktion, die es erlaubt Formatierungsregeln zu definieren. Diese Regeln färben Zellen in einer voreingestellten Farbe, falls deren Bedingung erfüllt ist. Die Erstellung ist intuitiv und unkompliziert. Nach Anpassung des Skriptes bleiben die Regeln, auch nach erneuter Ausführung, bestehen und gelten dann immer für das entsprechende Tabellenblatt. Wird das Skript in einem neu erstellten Tabellenblatt ausgeführt gelten zunächst Standard Regeln (Abbildung 5a), damit nicht alle Regeln neu erstellt werden müssen.

Die Spalten der Tabelle haben die folgende Ordnung:

Zeitstempel, Wochentag, Fahrernummer, Anmerkungen, Fahrten...

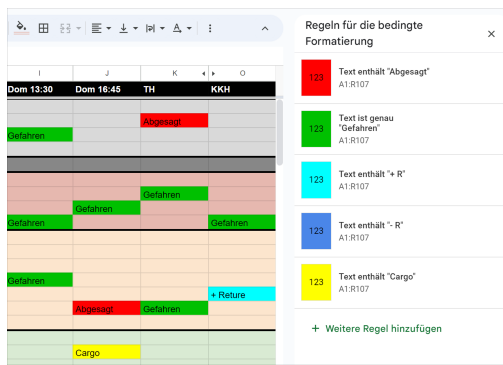
Die Spalten der Fahrten sind alphabetisch sortiert, das erleichtert die Suche nach bestimmten Namen und ermöglicht die Gruppierung von Fahrten. Zum Beispiel, falls Fahrten für den gleichen Kunden, aber zu unterschiedlichen Uhrzeiten stattfinden. Das ergäbe die Gruppierung:

... ,Kunde A 11:15, Kunde A 12:30, Kunde A 16:15, ...

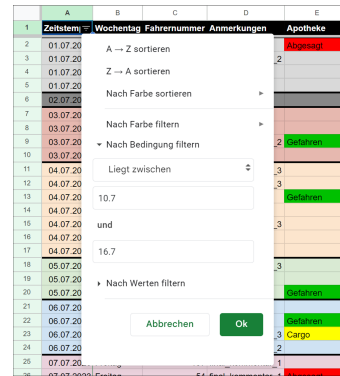
Um das Makro zu testen, füllten wir zunächst das Formular mehrfach aus und duplizierten dann die Zeilen in der Rohdatentabelle. Die so entstandenen Datensätze, waren jedoch nicht gut als reale Testdaten verwendbar. Die Testdaten sollten möglichst viele Kombinationen an möglichen Formulareingaben abdecken. Das wären bei 15 Fragen mit je 3 einfachen Auswahlmöglichkeiten an einem Wochentag schon $3^{15} \approx 14$ Millionen mögliche Kombinationen. Deshalb haben wir uns dazu entschieden ein Skript zur Testdatengenerierung zu erstellen, welches zufällige Formulareingaben liefert. Damit konnten wir das Makro auch mit größeren Datensätzen, im Bereich von 2000 - 3000 Einträgen testen. Diese Zahlen ergeben sich durch die Schätzung der gesamten Fahrten in einem Jahr. Danach empfehlen wir das Formular mit einer neuen Rohdatentabelle zu verknüpfen, um die Tabellengröße zu beschränken. Mehr über das Testdatenscript kann unter 4.1.5 Skript Testdatengenerierung gelesen werden.

Damit bietet unsere Implementierung eine visuell aufbereitete und geordnete Darstellung. Allerdings noch keine Möglichkeit, der gezielten Datenauswertung. Hier kommen die nativen Sheets-Funktionen ins Spiel. Mit Filter und Spaltenstatistiken können exakte Auswertungen vorgenommen werden. Die Filter-Funktion ermöglicht es Spalten auf eine Bedingung zu prüfen (siehe Abbildung 5b). Nur Zeilen, deren Werte in dieser Spalte die Bedingung erfüllen, werden angezeigt. So kann beispielsweise nach Fahrten eines bestimmten Fahrers oder FahrerIn gefiltert werden, indem auf die entsprechende FahrerNummer geprüft wird. Spaltenstatistiken geben einen Überblick über die Anzahl vorkommender Werte in einer Spalte. Bei den Kundenspalten kann so direkt herausgefunden werden wie häufig im Monat gefahren oder abgesagt wurde (siehe Abbildung 5c). Wenn sich in der Spalte mehrere numerische Werte befinden, werden statistische Maße, wie maximal Wert, Arithmetisches Mittel oder Summe, berechnet (siehe Abbildung 5d). So kann zum Beispiel direkt abgelesen werden wie viele Kisten in diesem Monat für Kunde X transportiert wurden. Mit der Kombination aus beiden Funktionen können auch komplexere Auswertungen durchgeführt werden. Allerdings kann mit der nativen Filter-Funktion nur nach Zeilen gefiltert werden. Es hat sich jedoch gezeigt, es in der Monatsansicht ebenfalls relevant ist nach Spalten zu filtern, das heißt nur bestimmte Fahrten anzuzeigen. Deshalb haben wir uns dazu entschieden eine solche Filter-Funktion, zusätzlich als Makro zu erstellen. Details zu dieser Funktion finden sich in 4.1.6 Skript Filterfunktion.

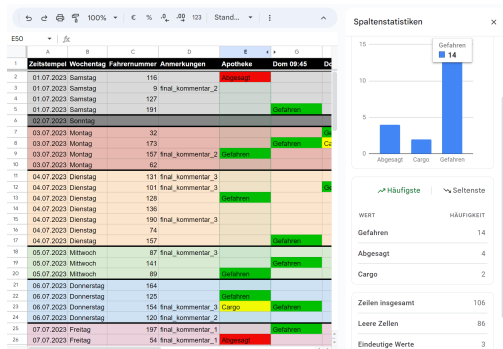
In den letzten Wochen des Projekts haben wir uns auf die Einarbeitung, Testung und Fehlerbehebung konzentriert. Dazu haben wir uns im häufig mit Radius, vor Ort im Betrieb, getroffen. Für die Einarbeitung haben gemeinsam, mit Radius das Formular gefüllt. Bisher waren in unseren Formularen nur Ausschnitte der Fahrten vorhanden. Deshalb war dieser Schritt besonders interessant, in Hinblick auf die Übersichtlichkeit an den jeweiligen Wochentagen. Zu unserer Freude war der Umfang in dem von uns erwarteten Rahmen. So sind es im aktuellen Formular durchschnittlich 15 Fahrten pro Wochentag. Mit der Unterteilung in das Schichtsystem, ist der Aufwand für das Scrollen überschaubar. Die eigentliche Erstellung der Fragen ist dank Google einfach und intuitiv. Wir hatten den Eindruck, dass Radius mit der Erstellung gut zurecht gekommen ist. Die Einführung in Sheets war anspruchsvoller, weshalb wir uns dafür auf ein eigenes



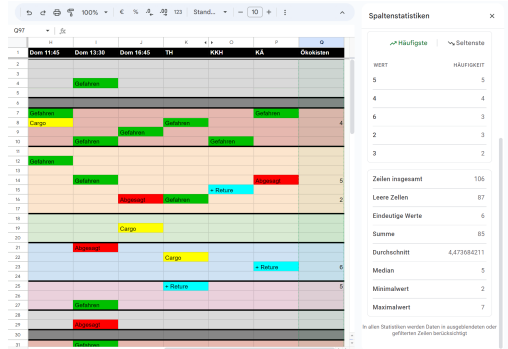
(a) Bedingte Formatierung



(b) Native Sheets Filterfunktion (nach Zeilen)



(c) Spaltenstatistik mit kategorischen Daten



(d) Spaltenstatistik mit numerischen Daten

Abb. 5: Native Sheets Funktionen für das Arbeiten mit der Tabelle.

Treffen verständigten. Da Radius im Umgang mit Sheets und dessen Funktionen bereits Erfahrung hat, waren die Funktionen für die Datenauswertung leicht zu vermitteln. Die Erklärung der Darstellung der Rohdaten und die Ursachen für diese, haben mehr Zeit in Anspruch genommen. Aber auch hier hatten wir den Eindruck, dass die Mitarbeiter von Radius uns folgen konnten und jetzt ein grundsätzliches Verständnis haben. Bei Monatsansicht und deren Erstellung haben wir Wert auf eine intuitive Ausführung gelegt. So findet sich nun im normalen Sheets Menü-Band der Eintrag „Fahrtenbücher“ mit dem Untermenü „Monatsansicht erstellen“ (Abbildung 6). Damit auch in Zukunft der volle Funktionsumfang des Systems nachvollzogen werden kann, haben wir uns entschieden zusätzlich eine Bedienungsanleitung zu erstellen. Dort sind die wichtigsten Funktionen erklärt, sowie einige Problemfälle und deren Behebung beschrieben (mehr dazu in 4.1.2 Bedienungsanleitung).

Um nun sicherzustellen, dass alles auf die vorgesehene Weise funktioniert, haben wir uns mit Radius auf eine Testphase geeinigt. Diese sollte sich in zwei Teile gliedern. Zunächst in eine Art interne Testung durch die Administratoren, also die Personen die bei Radius für Rechnungsstellung und Dokumentation zuständig sind. Im Anschluss sollte eine große Testphase folgen, bei denen die Kurier die Fahrten selbst eintragen, stattfinden.

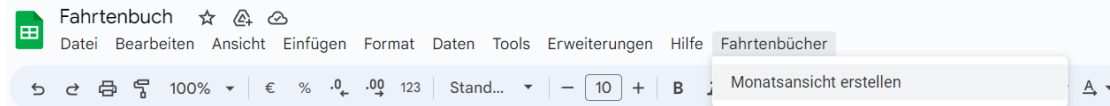


Abb. 6: Das Menü zum Erstellen der Monatsansicht.

So können wir, falls es zu schwerwiegenden Fehlern kommen sollte, diese gleich mit den Administratoren beheben. Damit sind diese für die große Testphase vorgegriffen. Wichtig an dieser Stelle ist anzumerken, dass unser System nur parallel genutzt wurde. Die Fahrten wurden nach wie vor in das bisherige System eingetragen. Dafür haben wir uns entschieden, um Datenverlusten vorzubeugen und die korrekte Rechnungsstellung nicht zu gefährden.

Aus dem Feedback der internen Testung ergaben sich einige Optimierungen. Bei der Besprechung mit Radius wurde der Wunsch geäußert auch an einem Wochentag mehrere Fragen mit demselben Titel verwenden zu können. Die damalige Implementierung ist davon ausgegangen, dass Fragen mit demselben Titel nur an verschiedenen Wochentagen vorkommen. Deshalb wurden bei mehrfach ausgefüllten gleichnamigen Fragen für einen Wochentag, je nur der letzte Eintrag in der Monatsansicht angezeigt. Um dem Abhilfe zu schaffen, werden nun alle Antworten mit einem | getrennt in einer Zelle angezeigt. Bei der Bearbeitung dieses Problems, ist uns aufgefallen, dass Radius mehrfach-Fragen häufig zur Abfrage von Anzahlen verwendet. Zum Beispiel „Wie viele Kisten wurden transportiert?“. Deshalb haben wir uns für eine Fallunterscheidung entschieden. Kommen Texte mehrfach vor, werden sie mit | getrennt aufgelistet. Kommen Zahlen mehrfach vor werden diese addiert und die Summe in die Zelle geschrieben. Eine weitere Optimierung hat sich aus dem Nachtragen von Fahrten ergeben. Bislang war es so, dass falls bei dem ersten Ausfüllen des Formulars etwas vergessen wurde, die Einträge eines erneut ausgefüllten Formulars mit derselben Fahrer Nummer, als eigene Zeile in der Monatsansicht angezeigt wurde. Hier war der Wunsch nur eine Zeile pro Wochentag pro Fahrer zu haben. Nun werden diese Einträge in einer Zeile angezeigt, wobei in den Spalten alle Einträge der Fahrten mit der gleichen Fahrer Nummer stehen. Falls Fragen mehrfach ausgefüllt wurden, wird analog zu oben mit | getrennt oder bei Zahlen aufsummiert.

Die große Testphase konnten wir leider aus zeitlichen Gründen nur eineinhalb Wochen begleiten. Ein früherer Beginn war aus unserer Sicht, sowie von Seiten Radius nicht sinnvoll. Da, wie bereits erwähnt, eine gut funktionierende Version an die Kuriere zum Testen übergeben werden sollte. Die interne Testung/kleine Testphase ergab bis zu diesem Zeitpunkt noch viel neues Feedback und Verbesserungsmöglichkeiten. Außerdem hatten wir den Eindruck das Radius immer vertrauter mit dem System wurde und selbst mit dem Formular „herum probierte“. Darüber hinaus haben die Kuriere meistens nur mit dem Formular zu tun und nicht mit der Tabelle, beziehungsweise der Monatsansicht. Deshalb freute es uns das Radius sich viel mit dem Formular auseinandersetzte. So kann in Zukunft Radius selbst Feedback von Kurieren einarbeiten. Ein Problem, das sich in der

großen Testphase ergab, war es bereits eingetragene Fahrten einzusehen. Kuriere wollten gerne nachsehen welche Fahrten sie bereits eingetragen hatten beziehungsweise eine falsche Eingabe nach dem Senden des Formulars korrigieren. Hier konnten wir leider nur eine Teillösung bieten. Es gibt die Option Formulare nachträglich zu bearbeiten. Ist diese in den Formulareinstellungen aktiv, erscheint am Ende, nach dem Senden, des Formulars ein Knopf, der es ermöglicht die Antworten zu bearbeiten. Das ist jedoch nur solange möglich, wie die Seite mit dem Formular geöffnet ist. Sobald die Seite geschlossen oder neu geladen wird kann lediglich ein neues Formular ausgefüllt werden. Weiteres Feedback der Kuriere konnte von Radius selbst in das Formular übernommen werden. Im Allgemeinen schien es keine gravierenden Probleme seitens der Kuriere zu geben und das Eintragen mit dem Formular zu funktionieren. Der tatsächliche Erfolg des Systems wird sich erst in der Zukunft zeigen. Wir, sowie Radius sind jedoch zuversichtlich, dass es seinen Zweck erfüllen wird.

3.7 Probleme im Verlauf des Projektes - und Lösungsansätze

Projektergebnis und Umsetzung waren nur sehr grob definiert:

Die größte Herausforderung, welche sich durch den gesamten Projektverlauf gezogen hat, war, dass das gewünschte Projektergebnis und die Umsetzung nur sehr grob definiert waren. Das Ziel war zwar immer klar, und zwar das Fahrtenbuch mit Blick auf Fahrteneintragung, Kontrolle der Fahrten und Rechnungserstellung weniger Fehleranfällig und weniger Zeitintensiv zu gestalten. Ein klassisches Pflichtenheft gab es zu Beginn aber nicht und musste erst erstellt werden, wobei sich herausgestellt hatte, dass bestimmte Ansätze und Überlegungen während dem Projektverlauf geändert werden mussten. Durch die verhältnismäßig geringe Zeit, die für die Durchführung zur Verfügung stand, musste auch der Ansatz gewählt werden, durch welchen die Art und Weise der Umsetzung während der Durchführung angepasst werden konnte.

Die Prozesse im Betrieb Radius waren zu Projektbeginn unklar:

Um eine möglichst gute Lösung für den Betrieb Radius finden zu können, mussten zunächst deren interne Prozesse möglichst gut verstanden werden. Die Art und Weise wie deren Fahrten bisher dokumentiert wurden sind schon deshalb relevant, da unsere angebotene Lösung nur dann verwendet werden wird, wenn sie von den Mitarbeitern im Büro genauso wie von den Fahrradkurieren Akzeptanz erhält. Dies benötigte viel Kommunikation mit den Projektstellern.

Wenig Zeit für Testphase:

Für die Testphase musste von Beginn ein geeigneter Zeitraum freigehalten werden, welcher den Zeitraum für die eigentliche Umsetzung des Projekts weiter verringerte. Vor allem mit Blick auf die Testphase war der gesamte Projektzeitraum sehr klein. Es ist zum jetzigen Zeitpunkt nur sehr schwer abzuschätzen, ob die Testphase lange genug gewählt wurde, bzw. wie viel mehr Zeit benötigt würde, damit diese lange genug gewesen wäre.

Zeitlich und finanziell geringer Rahmen:

Der zeitlich und finanziell geringe Rahmen führte dazu, dass wir uns für die Umsetzung mit Google Sheets und den Google Formularen entschieden hatten. Wie an anderen Stellen in diesem Projektbericht zu lesen, hatte dieser Lösungsansatz viele Vorteile. Allerdings gibt es auch einige Beschränkungen, für welche wir Kompromisse eingehen mussten oder einen recht Zeitintensiven Lösungsansatz wählen mussten, wie beispielsweise das Skript, um die unübersichtliche Rohdatentabelle aufzubereiten. Ein größerer finanzieller Rahmen hätte die Nutzung von „Baukastensystemen“ (z.B. AppSheet) erlaubt, welche bessere Möglichkeiten für eine Webanwendung mit Datenauswertung bieten ohne dabei Programmieren zu müssen. Ein größerer zeitlicher Rahmen hätte eine größere Testphase ermöglicht.

Laufender Betrieb bei Radius durfte nicht gestört werden:

Dies spielte eine geringere Rolle beim Vereinbaren von Treffen mit den Projektstellern. Bei der Testphase stellte dies allerdings ein größeres Problem dar. Daher wurden zunächst ausführliche Tests durch das Projektteam durchgeführt. Darauf folgte eine Bürointerne Testphase beim Betrieb Radius. Hier wurden die Fahrten parallel zum bestehenden System in unser neues System eingetragen und testweise verwendet. Der Test unter Realbedingungen konnte erst starten, sobald alle Probleme und Änderungswünsche behandelt wurden.

Fehlendes Know-How und Ressourcen bei Radius für selbstständige Weiterführung des Projekts:

Es ist als eines der größten Probleme hervorzuheben, dass die Fahrradkuriere von Radius nicht über das Know-How und die Ressourcen verfügen, um nach Projektende selbständig größere Änderungen am Programm durchzuführen, falls es sich herausstellen sollte, dass diese nötig wären. Dies musste bei jeder Entscheidung bedacht werden. Es wurde von Beginn an Fokus daraufgesetzt, dass ein vollständig funktionsfähiges und fehlerfreies Programm übergeben wird. Das fehlende Know-How führte auch zu der Entscheidung eine Bedienungsanleitung zu schreiben die zusätzlich viel Zeit in Anspruch genommen hat.

Durch Verwendung von Google Forms mussten Kompromisse eingegangen werden:

All die bisher genannten Punkte führten dazu, dass in sehr regelmäßig und kurzen Abständen Rücksprache mit den Projektstellern gehalten werden musste. Der Lösungsansatz mit Google Forms führte dabei dazu, dass Kompromisse eingegangen werden mussten. Die Auftragsteller hätten sich an mancher Stelle im Formular kleine Änderungen gewünscht, wobei die endgültige Lösung einen für alle zufriedenstellenden Kompromiss darstellt. Die meisten Wünsche konnten mit den Google Formularen sehr gut umgesetzt werden. Die Wünsche, denen wir nicht vollkommen gerecht werden konnten möchten wir hier kurz betrachten und erklären, warum das Ergebnis trotzdem den besten Möglichkeiten entspricht:

- **Eingabe Datum:** Ein Date-picking-Kalender wurde gewünscht. Es stellte sich allerdings heraus, dass das Eingabeformat für Daten durch Google Forms grundsätzlich einen Date-picking-Kalender bereitstellt, dieser aber beispielsweise mit dem Firefox-Webbrowser nicht verwendet werden kann. Alle anderen Eingabeformen für das Datum wären fehleranfälliger und aufwändiger gewesen
- **Abfrage Wochentag:** Die Abfrage des Wochentags wirkt redundant, da bereits das Datum bekannt ist und bietet auch Potential für eine widersprüchliche Eingabe von Datum und Wochentag. Eine Falsche Eingabe wird in der Übersicht durch das Skript farblich markiert. Die Abfrage des Wochentags ist allerdings sinnvoll, da dadurch im weiteren Verlauf des Formulars nur Fahrten angezeigt werden, welche auch wirklich an diesem Tag stattfinden und nicht alle Fahrten, welche an irgendeinem Tag in der Woche stattfinden könnten. Dies erleichtert die Eingabe im Formular für die Fahrradkuriere sehr, bedeutet aber auch etwas mehr Arbeit, beim Einpflegen von neuen Fahrten in das Formular. Eine automatische Abfrage des Wochentags über das Datum direkt im Formular ist nicht ohne weiteres möglich.
- **Sehr hohe Spaltenzahl in der verknüpften Tabelle:** Der komplexere Aufbau des Formulars mit vielen verschiedenen Abschnitten führt zu einer sehr hohen Spaltenzahl in der verknüpften Tabelle. Dies ist durch die Funktionsweise der Formulare so gegeben und war für uns nicht einfach zu ändern. Die Lösung war das von uns geschriebene Skript, welches die Spaltenzahl auf das nötigste reduziert und zusätzlich noch die Möglichkeit von anderen customized Darstellungen bietet.

3.8 Beiträge der Gruppenmitglieder

Inhaltlich

Aufgabe	Bearbeitet von
Recherche	Nico, Martin und Flavian
Formulare kreieren	Nico, Martin und Flavian
Skript(terminierte Fahrten): Logik, Färbung, Daten Selektion	Flavian
Skript(terminierte Fahrten): Trennstriche, bedingte Formatierung	Nico
Skript(Sonderfahrten)	Flavian
Skript(Testdatengenerierung)	Martin
Skript(Spaltenfilter)	Martin
Bedienungsanleitung	Nico, Martin und Flavian
Koordination/Kommunikation mit Radius	Flavian, Nico
Protokoll bei Treffen mit Radius	Martin, Flavian
Administration Sheets/Forms	Flavian, Nico

Projektbericht

Abschnitt	Bearbeitet von
1 Abstract	Flavian
2 Einleitung	Nico
3.1 Koordination in der Gruppe	Nico
3.2 Kontakt und Interaktion mit den Projektstellern	Nico
3.3 Einordnung	Flavian
3.4 Problemstellung	Flavian
3.5 Zeitlicher Ablauf	Flavian
3.6 Vorgehen	Flavian
3.7 Probleme im Verlauf des Projektes - und Lösungsansätze	Nico
3.8 Beiträge der Gruppenmitglieder	Flavian, Nico und Martin
4.1.1 Übersicht	Nico
4.1.2 Bedienungsanleitung	Nico
4.1.3 Formular Sonderfahrten	Nico
4.1.4 Skript Sonderfahrten	Nico
4.1.5 Skript Testdatengenerierung	Martin
4.1.6 Skript Filterfunktion	Martin
4.2 Diskussion	Martin
4.3 Wie könnte es weitergehen?	Martin

4 Projektergebnisse und Diskussion

4.1 Deliverables/Resultate

4.1.1 Übersicht

Zum eigentlichen Projektergebnis gehört vor allem das im vorherigen Abschnitt vorgestellte Formular für die Eingabe der Fahrten und das Marko, welches die Monatsübersicht erstellt. Weitere Leistungen waren allerdings nötig, um eine Verwendung im Arbeitsalltag vom Betrieb Radius zu ermöglichen.

Übersicht der Deliverables:

- Formular für Fahrteneingabe (terminierte Fahrten)
- Skript für Monatsübersicht (terminierte Fahrten)
- Skript für Filterfunktion
- Bedienungsanleitung

Weitere Resultate, welche nicht ausgeliefert wurden:

- Formular für Sonderfahrten
- Skript für Sonderfahrten
- Skript für Testdatengenerierung

In den folgenden Unterkapiteln werden die Projektergebnisse beschrieben, welche an anderer Stelle noch nicht genauer erklärt wurden.

4.1.2 Bedienungsanleitung

Motivation

Da der Betrieb Radius keinen fortlaufenden Support von uns erhält und keine Mitarbeiter hat, welche sich um solch komplexe Implementierungen wie dieses Fahrtenbuch kümmern können, haben wir uns entschieden eine Bedienungsanleitung zu schreiben. Die Bedienungsanleitung soll einen Überblick über den gesamten Funktionsumfang des Fahrtenbuchs geben und als Hilfsmittel bei Problemen dienen. Es werden dort Schritte erklärt, die je nach Verwendung wöchentlich, monatlich oder auch nur jährlich durchgeführt werden müssen und in dieser Zeit in Vergessenheit geraten können. Große Aufmerksamkeit bei der Erstellung der Bedienungsanleitung lag außerdem auf der Erklärung von Problemen, welche bei falscher Handhabung auftreten könnten.

Aufbau

Zu Beginn der Bedienungsanleitung wird eine Beschreibung des Fahrtenbuchs gegeben. Anschließend werden die grundsätzlichen Funktionen beschrieben, welche für die korrekte Funktionsweise des Fahrtenbuchs wichtig sind. Die meisten dieser Funktionen sollten nach kurzer Einarbeitung in das Programm intuitiv und leicht verständlich sein. Trotzdem wurden diese in die Bedienungsanleitung aufgenommen, damit diese vollständig ist und auch für einfache Arbeiten mit dem Programm als Nachschlagwerk dient. Daraufhin folgt ein Abschnitt, welcher die Filterfunktionen und Statistiken erklärt. Diese sind sehr nützlich, um die Rechnungserstellung zu vereinfachen. Der Zugang zum Programmcode wird ebenfalls in einem Kapitel beschrieben, allerdings mit dem Hinweis darauf, dass Änderungen dazu führen können, dass das Programm nicht mehr richtig funktioniert. Ein wichtiges Kapitel stellt das letzte dar, da dieses die Probleme beschreibt, welche auftreten könnten. Dabei handelt es sich meist um Probleme, welche im Zusammenhang mit unserer Implementierung stehen. Beispielsweise müssen die Tabellenblätter eine bestimmte Namensstruktur aufweisen, damit das Programm weiß, was es machen soll. Eine Tabelle mit dem Namen „Fahrtenbuch Juli 23“ erzeugt die Übersicht für Juli 2023. Eine Tabelle mit dem Namen „Juli 2023 Fahrtenbuch“ weist, wie im Bedienhandbuch beschrieben, die falsche Namensstruktur auf und führt zu einem Fehler (mit Fehlermeldung). Dieser Fehler wäre durch eine selbstständige Internetrecherche nicht zu beheben. Um den Fehler ohne Bedienungsanleitung nachvollziehen zu können würde es eine Analyse des Codes benötigen.

Inhaltsverzeichnis

1. Beschreibung	3
2. Leeres/neues Fahrtenbuch erstellen und einrichten (Jahreswechsel)	4
a. Formular kopieren	4
3. Zugriffsrechte bearbeiten	5
4. Terminierte Fahrten	6
a. Formular öffnen und ausfüllen	6
b. Formular bearbeiten	8
i. Fahrten hinzufügen	8
ii. Fahrten bearbeiten.	9
iii. Fahrten entfernen,duplizieren,verschieben.	9
c. Erklärung zur Rohdaten-Tabelle	10
d. Monatsübersicht erstellen	10
e. Erklärung zur Monatsübersicht	11
5. Filterfunktionen	12
a. Zeilen filtern	12
b. Spalten filtern	13
6. Statistiken anzeigen	16
a. In der Tabellenspalte	16
b. Formular Statistik	17
7. Skript (Code)	17
8. Probleme	18
a. Abgesendete Formulardaten werden nicht in Tabelle angezeigt	18
b. Monatsübersicht kann nicht erstellt werden	18
c. Statistik ist falsch	18
d. Es sind unnötige Spalten in der Monatsübersicht	18
e. Spalten in Monatsübersicht werden nicht zusammengefasst	19

Abb. 7: Das Inhaltsverzeichnis der Bedienungsanleitung.

Resümee Bedienungsanleitung

Die Bedienungsanleitung zu schreiben, gehörte zu den trockeneren und weniger spannenden Tätigkeiten des Projekts und hat durch den großen Umfang auch viel Zeit in Anspruch genommen. Wir hoffen und denken trotzdem, dass die Bedienungsanleitung dem Betrieb Radius helfen wird Probleme selbstständig lösen zu können, auch wenn wir natürlich davon ausgehen, dass wir alle Funktionalitäten ausführlich genug erklärt haben, sodass keine Probleme auftreten sollten. Ein Problem, welches nicht selbstständig vom Betrieb Radius gelöst werden kann, könnte dazu führen, dass das Programm nicht weiter verwendet werden kann. Dies wäre sehr schade, für den großen Zeitaufwand und die Mühe, die in das gesamte Projekt gesteckt wurden. Daher glauben wir, dass es die Arbeit wert war.

4.1.3 Formular Sonderfahrten

Das Formular für Sonderfahrten wurde ähnlich entwickelt wie das Formular für terminierte Fahrten. Es unterscheidet sich allerdings an einigen Stellen und ist teilweise sogar komplexer. Das Formular war funktionsfähig, allerdings bestand im Büro Radius die Sorge, dass das etwas komplexere Formular zu Problemen bei der Eingabe durch die Fahrer führen könnte. Die größte Schwierigkeit bestand darin, dass die Anzahl der Sonderfahrten variiert und diese variable Anzahl nicht gut durch Google Forms dargestellt werden kann. Das Eingabeschema, welches nötig war, ist nicht so intuitiv wie die Eingabe der terminierten Fahrten, hätte aber die gleichzeitige Eingabe von bis zu fünf Fahrten ermöglicht. Das Büro vom Betrieb Radius hatte sich daher im Laufe des Projekts entschieden die Sonderfahrten wie bisher zu handhaben. In der folgenden Abbildung ist ein Ausschnitt des Formular Sonderfahrten zu sehen, auf dem man auch das Eingabeschema sieht. Alle Informationen, welche dort abgefragt werden, sind notwendig für das Fahrtenbuch. Dadurch hätte eine Eingabe ohne dieses Schema zu einem sehr langen und unübersichtlichen Formular geführt.

4.1.4 Skript Sonderfahrten

Auch dieses Skript, welches die unübersichtliche Darstellung der Sonderfahrten zu einer schönen Übersicht zusammenfasst, war bereits funktionsfähig. Es wurde sogar ohne weitere Anpassungen als sehr gut für die Rechnungserstellung befunden. Im Zusammenhang mit den bereits beschriebenen Problemen beim Formular Sonderfahrten kam es allerdings nicht zu einer Auslieferung dieses Skripts. Auf eine genauere Beschreibung kann hier verzichtet werden, da die Darstellung der Fahrten sehr der Darstellung der terminierten Fahrten ähnelt. Der Code des Skripts hatte allerdings teilweise große Unterschiede, welche damit zusammenhängen, dass das Formular anders aufgebaut war. Die Komplexität des Formulars spiegelte sich deshalb auch teilweise im Code wider.

Abb. 8: Eingabeschema mehrerer Fahrten im Formular Sonderfahrten.

4.1.5 Skript Testdatengenerierung

Es wurde ein umfangreiches Python-Skript geschrieben, um Testdaten zu generieren. Diese Testdaten erlaubten uns ausführliche Tests unserer Prototyp-Algorithmen ohne den Betrieb bei Radius dafür stören zu müssen. Die Testdaten, die auch sehr schnell erstellt werden, dienen als Eingabedaten für die Algorithmen zur Erstellung der Monatsansicht. Die automatisch generierten Daten haben uns außerdem viel händische Arbeit abgenommen, da wir anfangs Testdaten durch manuelles Ausfüllen der Google-Formulare erzeugten. Größere Mengen an Testdaten konnten zudem verwendet werden um das Laufzeitverhalten der Prototyp-Algorithmen als positiv einzuschätzen. Wir erstellten in einem Szenario Testdaten im Umfang von 3000 Zeilen für den Zeitraum eines Jahres. Der Prototyp - Algorithmus erstellte eine Monatsansicht für diese Datenmenge in zirka 5 Sekunden, ein Wert, der sehr akzeptabel ist. Neben der Laufzeiteinschätzung war es auch wichtig einen visuellen Eindruck zu bekommen, wie nun die Daten nach Anwendung des Algorithmus, in einer Monatsansicht aussehen würden. (der erste Eindruck war positiv auch seitens Team Radius). Um den Visuellen Eindruck der Monatsansicht aussagekräftig zu machen, war es klar, dass die Testdaten echten Datensätzen zumindest annähernd gleichen sollten. Nachdem Ressourcen und Zeit freigeschafft wurden, wurde ein umfangreiches Python Skript implementiert, das halbwegs realistische Datensätze ge-

bei der sich das Haupt - Dialogfenster öffnet (Abbildung 11a). In dieser Ansicht werden in einem grün/grauen Bereich alle vorhandenen Filternamen aufgelistet und können optional auch durch Anklicken ausgeführt werden. Jeder Filter kann zudem bearbeitet (Abbildung 11b) sowie gelöscht werden. Nur der Filter „Alle Spalten anzeigen“ ist unveränderbar. Er dient dazu die Monatsansicht in der originalen Form wiederherzustellen. Durch Auswählen des Buttons „Neuen Filter Erstellen“ öffnet sich ein weiteres Dialog - Fenster (Abbildung 11c) in dem neuen Filter erstellt werden können. Ein neuer Filter erfordert zunächst einen Filternamen, welcher im dynamischen Untermenü erscheinen wird, sowie in der Auflistung im grün/grauen Bereich des Haupt - Dialogfensters. Des Weiteren erfordert ein Filter einen Filterausdruck der im Skript und im Filtervorgang als regulärer Ausdruck interpretiert wird. Durch reguläre Ausdrücke ist ein mächtiges Werkzeug bereitgestellt, das komplexe Filter - Kriterien zulässt; worauf in der Bedienungsanleitung eingegangen wird. So sind Zeitintervallfilter denkbar, die Spalten mit Zeitangaben filtern wie etwa nach den Kategorien vormittags oder nachmittags. Diese Art hilfreiche Datenaufbereitung könnte sich bei Radius potenziell als nützlich herausstellen, weshalb hier die Entscheidung getroffen wurde, die mächtigen Regulären Ausdrücke zu nutzen. Mit dem Button “ Filter Hinzufügen” wird der Filter dauerhaft (persistent) gespeichert und ist auch einsatzbereit.

Zuletzt hat die Implementierung der Spaltenfilterfunktion auch die Möglichkeit, bestimmte Spalten in der Filteroperation zu ignorieren. So gibt es eine Einstellungsmöglichkeit, bestimmte Zeilen immer anzuzeigen, egal was das Filterkriterium ist (der Filterausdruck). Diese Anforderung wurde auf Anfrage von Radius umgesetzt, um etwa die Spalten „Wochentag“ und „Fahrernummer“ dauerhaft anzuzeigen. Um diese Standardspalten zu konfigurieren gibt es den Button “ Einstellungen” in dem Haupt - Dialogfenster. Durch Anklicken des Buttons wird ein weiteres Dialogfenster geöffnet (Abbildung 11d) bei dem Standardspalten geändert und gespeichert werden können.

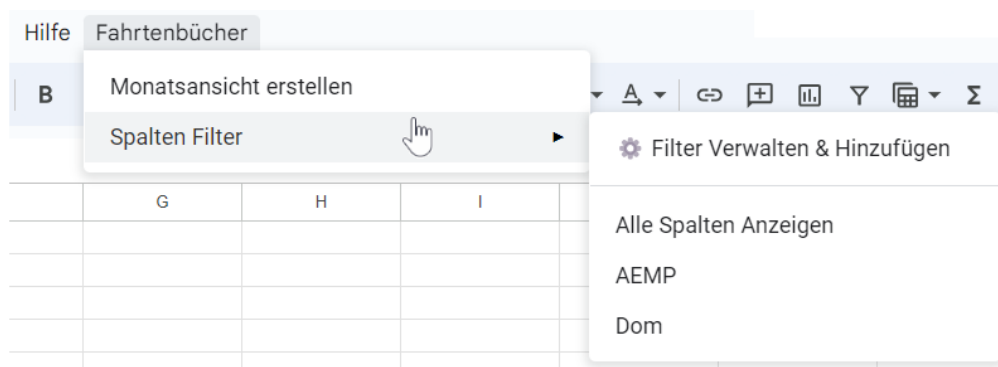


Abb. 10: Einbindung der Spaltenfilterfunktion in dem benutzerdefiniertem Menü.

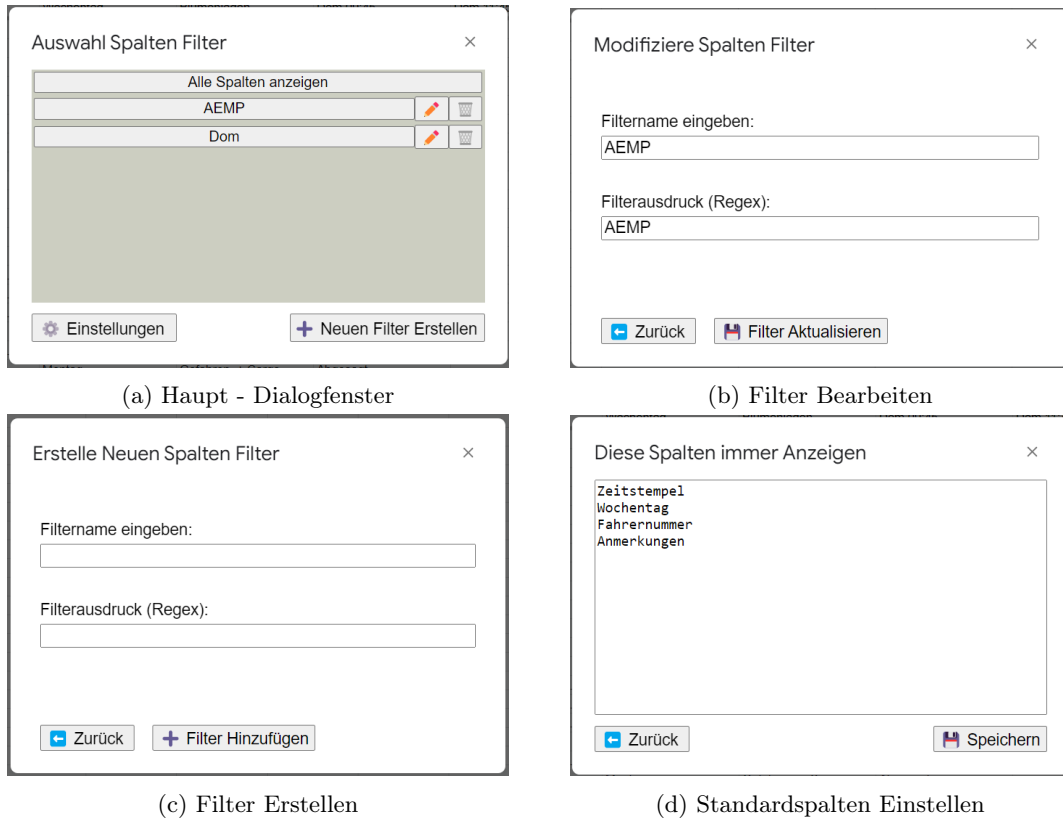


Abb. 11: Dialog Elemente der Spaltenfilter - Implementierung.

4.2 Diskussion

Die Hauptaufgabe des Projekts bestand darin, ein System zu konzeptionieren und zu implementieren, dass die Protokollierung von Auslieferungsfahrten bei Fahrradkurier Radius verbessert. Es ging auch darum, protokollierte Daten so bereitzustellen, das eine einfache Weiterverarbeitung von Informationen über Lieferungen (Kurierfahrten) möglich ist. Dadurch werden die Verwaltungsaufgaben vereinfacht. Ganz allgemein haben wir die Projektziele erreicht, indem wir ein Datenerfassungssystem über Google - Formular eingerichtet haben, sowie auch ein Makro für Monatsansichten und ein Tool zur Spaltenfilterung implementiert haben. Die Frage nach der Qualität unseres Systems ist differenzierter zu betrachten. So war die Anforderung ein verbessertes System bereitzustellen mit folgenden Unterkriterien verknüpft:

- Zeitersparnis
- Falsche Datenerfassung minimieren
- Einfacher/Einheitlicher/Übersichtlicher
- Vollständig einsatzbereit zum Projektabschluss

Zeitersparnis: Unser System strukturiert den Prozess zur Eintragung von Informationen zu Lieferungsfahrten. Dabei können Kurierere sowie Kurierinnen über ein Smartphone oder auch von beliebigen Rechnern aus ihre Daten eintragen. Diese Flexibilität ist an sich schon zeiter sparend, da die Protokollierung nicht wie zuvor am Firmen - Laptop geschehen muss. Andererseits kann es auch eine Frage der Präferenz sein, wie leicht und effektiv sich das neue System im Vergleich zum alten benutzen lässt. So ist es denkbar, dass einige Kurierere das alte System (manuelles Eintragen in einem Spreadsheet) bevorzugen, und wiederum andere das neuere als Verbesserung ansehen. Um das Kriterium Zeitersparnis seitens der Datenerfassung objektiv evaluieren zu können, wäre eine Gewöhnungsphase an das System notwendig oder wünschenswert. Diese Evaluierung konnte nicht mehr im Zeitplan des Seminars untergebracht werden. Andererseits wurde eine deutliche Zeitersparnis durch die vereinfachende Monatsansicht gewonnen, welche die monatliche Rechnungstellung bei Radius sehr erleichtert.

Falsche Datenerfassung minimieren: Die Minimierung von Fehlerquellen im Datenerfassungsprozess war eine wichtige Anforderung an das Fahrtenbuchsystem. Die Möglichkeit falsche Daten zu erfassen, sollte möglichst eingeschränkt werden, um eine getreue Protokollierung zu gewährleisten. Wieder stellte sich das Formular als adäquate Lösung heraus, insbesondere deshalb, weil die Dateneingabe sehr strukturiert stattfindet. Das Formular fungiert als Schnittstelle und verhindert den direkten Zugriff auf die Rohdatenspeicherung und somit auch katastrophale Fehlerquellen, wie versehentliches Löschen von Datensätzen oder Einfügen von Daten an falscher Stelle. Allerdings ist die Eingabe von falschen Informationen nicht ausgeschlossen, so könnte ein Kurier eine bestimmte Fahrt als „gefahren“ eintragen und ein anderer diese als „abgesagt“. An dieser Stelle hat unser System klar Verbesserungsbedarf. Hier wäre ein Lösungsansatz, das Formular mit einem Skript zu ergänzen (Google Apps Script), das Fehler - Situationen erkennt und falsche Eingaben blockiert. Diese Problematik war uns bewusst und konnte teils durch die bedingte Formatierung gelöst werden, allerdings nur durch farbliche Markierungen (Warnhinweise) in den Daten, und nicht als abwehrender Mechanismus. So werden zum Beispiel in der Monatsansicht Einträge/Zellen rot markiert, in denen Wochentag und Fahrtendatum in Konflikt stehen.

Einfacher/Einheitlicher/Übersichtlicher: Die Datenerfassung über das Google - Formular findet als geleiteter Prozess statt, was sehr für die Einfachheit spricht. Im Vergleich zum manuellen Einfügen von Daten ist die Datenerfassungsschnittstelle über ein Formular auch einheitlich; die Abfrage folgt einem Schema. Zudem ist der Kontrollfluss im Formular so gestaltet, dass nur wochentagsspezifische Abfragen angezeigt werden, ein Gestaltungselement, das die Erfassung übersichtlicher macht. Was nicht für die Übersichtlichkeit spricht, ist die Google - Interne Speicherung der Formulareinträge in einer Rohdatentabelle, die in der Breite (Spalten) sowie in der Länge (Zeilen) sehr groß werden kann. Da es keine technische Möglichkeit gab, diesen Aggregationsprozess der Daten zu steuern, und wir dennoch nicht auf das Google - Formular in unserem Gesamtsystem verzichten wollten, implementierten wir die Monatsansicht - Funktion. Trotzdem bleibt die

Rohdatentabelle bestehen, wobei sich das Editieren von Einträgen in Notfallsituationen aufwändiger als erwünscht herausstellt. Andererseits sind die Vorteile des Google - Formulars, dass die Verwaltungspersonen bei Team Radius die Fahrten - Abfragen unkompliziert bearbeiten können, was zum Beispiel bei neuen Kunden der Fall ist. Letztlich sind die generierten Monatsansichten übersichtlich und die zusätzliche Option der Spaltenfilterung ermöglicht eine noch spezifischere Ansicht der Daten. Die farbliche Markierung der Wochentage sowie die gewählte Farbauswahl für die bedingte Formatierung sind weitere Details, die die Daten höherwertig aufbereiten, und wurden in Absprache mit unseren Projektgebern gut getroffen.

Ein weiterer Kritikpunkt an unserem System ist, dass sich das nachträgliche Korrigieren von falschen Eingaben über die Formularechnittstelle für die Kuriere als schwierig herausstellte. Zwar kann in einer Browsersession ein bereits gesendeter Eintrag, revidiert werden (durch Öffnen eines Links am Ende des Formulars), wird jedoch der Browser oder Tab geschlossen, gibt es keine andere Möglichkeit mehr, als durch die Verwaltungspersonen die Dateneinträge nachträglich in der Tabelle zu bearbeiten. Hier wäre ein umfangreicheres System notwendig.

Ist die Funktionsweise des Fahrtenbuchsystems verstanden und auch die oben genannten Einschränkungen bekannt, dürften sich fehlerhafte Eingaben durch Erfahrung und eine Gewöhnungsphase stark reduzieren.

Vollständig einsatzbereit: Ein weiteres Projektziel war es, am Ende des Projektseminars ein vollständig einsatzbereites Fahrtenbuchsystem bereitzustellen. Auch diese Anforderung konnte erfüllt werden. Das erstellte System ist robust unter korrekter Verwendung, die Skripte wurden bestmöglich auf Korrektheit geprüft, und auch die ausführliche Dokumentation vervollständigt das Gesamtsystem. Allerdings ist der Einsatzbereich des neuen Fahrtenbuchs speziell für terminierte Fahrten, und umfasst nicht alle Lieferungsfahrten, die bei Fahrradkurier Radius stattfinden. So gibt es noch die Kategorien spontane Sonderfahrten und Großkundenfahrten. Die Protokollierung der Großkundenfahrten war für unser Arbeit nicht vorgesehen, und so befassten wir uns zuerst mit terminierten Fahrten und Sonderfahrten. Es wurde auch an einem Fahrtenbuch für spontane Sonderfahrten gearbeitet. Letztlich wurde dieses nicht übernommen, da das komplexere Abfrageschema zu Problemen bei der Eingabe führen konnte. Als Resultat war der Fokus des Projekts die Entwicklung des Fahrtenbuchsystems für terminierte Fahrten. Dieses System kann fehlerlos genutzt werden und wurde mit kurzer Einführung bei Team Radius vor Projektende abgegeben.

Verbesserungsbedarf: Rückblickend auf den Verlauf des Projekts, hätte die Anforderungsanalyse des Fahrtenbuchsystems strukturierter gestaltet werden können. Eine Möglichkeit wäre, eine umfassende Befragung von mehreren Kurieren und Kurierinnen, um ein noch breiteres Spektrum an Ideen und Vorstellungen zu sammeln. Es wäre auch sinnvoll gewesen, Anwendungsfälle oder sogenannte Use Cases, die in Bezug auf die Datenerfassung und die Datenweiterverarbeitung auftreten zu detaillieren, wie etwa durch Use - Case Diagramme.

Prinzipiell wäre ein Tool zur automatisierten Rechnungsstellung als Ergänzung des Fahrtenbuchsystems technisch realisierbar und von sehr hohem Nutzen für die Verwaltungsaufgaben. Im Verlauf des Projekts wurde diese Idee öfters diskutiert mit der Vorstellung, dass per Knopfdruck Rechnungen im PDF - Format erstellt werden könnten. Zeitlich konnte dieses ambitionierte Ziel nicht in Angriff genommen werden; es war zudem auch kein konkretes Projektziel. Auch wären die Qualitätsanforderungen an dieses Tool sehr hoch, und nicht optimal vereinbar mit der Datenspeicherungsart in unserem erstellten System; passender wäre ein Datenbanksystem aus dem kundenspezifische Daten sowie Preisangaben gezielt extrahiert und kombiniert werden können.

In Bezug auf den vorgesehenen Zeitumfang des Projektseminars, den vorhandenen Ressourcen und den Fähigkeiten der Projektmitglieder wurde eine passende Lösungsmöglichkeit diskutiert, exploriert und umgesetzt. Die Rahmenbedingungen wie etwa die Vorgabe, das Projektziel, ohne einmalige oder monatlichen Kosten umzusetzen, führten zu einer Lösung, die zwar noch Optimierungspotenzial hat, aber dafür gut an die Struktur des Unternehmens angepasst ist. Das System wurde zum Beispiel in der bereits gut vertrauten Umgebung von Google Sheets integriert. Rein technisch gesehen wäre ein komplizierteres Fahrtenbuchsystem vorstellbar gewesen, welches das angenehme Arbeitsklima und eventuell auch Aspekte der sozialen Nachhaltigkeit verändert hätte. So wäre ein Programm denkbar in dem Administratoren Routen für bestimmte Mitarbeiter vorbestimmen, Profile und Statistiken über einzelne Kuriere sammeln, etc. was wiederum nicht gut mit der flachen Hierarchie des Unternehmens vereinbar wäre. Auch die Vorgabe an das zu entwickelnde System, keine Pflege durch Fachexperten zu benötigen, schränkte den technischen Lösungsraum etwas ein, führte aber zu einer Lösung, die das Unternehmen ohne größeren Aufwand oder Kosten in Zukunft nutzen kann. Die endgültige Bewertung des Projekts wird sich in der konkreten Umsetzung des Fahrtenbuchsystems zeigen, wenn Team Radius zum Beispiel eine Zeitersparnis in der Verwaltung bei gleichzeitig größerer Genauigkeit in der Datenerfassung erzielt. Nach der Vorläufigen Rückmeldung der Projektsteller rechnen wir damit das dieses Ziel erreicht werden kann:

„Das Fahrtenbuch wurde genauso umgesetzt, wie wir es uns vorgestellt haben und es ist nun optimal nutzbar. Wir starten im Juli mit der Testphase mit allen Kurieren und wollen weiterhin das neu erstellte Fahrtenbuch dauerhaft einbinden. Zusätzlich sind wir in der Lage selbstständig mit den Formularen zu arbeiten, sie zu aktualisieren und auszuwerten“ – Team Radius

Wir hoffen zudem, dass die dauerhafte Nutzung des neuen verbesserten Fahrtenbuches das umweltfreundliche Unternehmen in seinen Verwaltungsaufgaben stärkt, und dadurch indirekt einen kleinen Beitrag zur Erhaltung der Umwelt leistet.

4.3 Wie könnte es weitergehen?

Das Fahrtenbuchsystem, das in dieser Projektarbeit erstellt wurde, kann an einigen Stellen durch weiteren aufwand ergänzt beziehungsweise verbessert werden. Ein weiteres Skript (Google App Script) könnte wie bereits erwähnt, die Problematik der falschen eingaben in dem Formular lösen.

Das Problem, dass Kuriere und Kurierinnen ihre Eingaben nicht einsehen können, ließe sich durch eine Websitendarstellung der Monatsansicht beheben. Dabei geht es um die Option „Im Web veröffentlichen“ die Sheets bietet. Allerdings müsste dann die Monatsansicht bei jedem Öffnen der Website eine Aktualisierung durchführen, indem das Makro ausgeführt wird. Sonst wird nur die Monatsansicht auf dem Stand des letzten Ausführens angezeigt. Hier könnten sogenannte „Trigger“ verwendet werden. Diese bieten die Möglichkeit Callbacks zu definieren. Ob und wie ein solcher Callback für die Websitendarstellung funktioniert müsste recherchiert werden.

Eine Beobachtung, die im Zusammenhang mit dem Projektseminar gemacht wurde, ist dass die Verwendung von Google - Formularen für weitere Zwecke nützlich sein kann. Weitere Verwaltungsaufgaben oder auch Umfragen an die Mitarbeitenden bei Radius könnten leicht umgesetzt werden. Da die Systematik und Funktionsweise der Formulare den Projektstellenden durch dieses Projekt bekannt ist, wäre eine weitere Verwendung einfach möglich.

Ein weiteres Projektseminar in Zusammenarbeit mit Fahrradkurier Radius wäre vorstellbar, um etwa die automatisierte Rechnungsstellung in Angriff zu nehmen. Weitere Interessante Themen besonders aus der Sicht der Informatik wären die optimierte Routenplanung sowie die automatisierte Erfassung von Fahrtendaten über GPS - Tracking. Sollte das Unternehmen Fahrradkurier Radius, in Zukunft deutlich expandieren, wäre die Anschaffung von individuellen Softwarelösungen für die Verwaltungsaufgaben relevant.

5 Anhang

- Skripte
 - Controller.gs
 - SummarySonderfahrten.gs
 - SummaryTerminierteFahrten.gs
 - dialogue_logic.gs
 - dialogue_addition.html
 - dialogue_selection.html
 - dialogue_settings.html
 - Testdatengenerierung.py
- Bedienungsanleitung.pdf