EIS

Entwicklung interaktiver Systeme

Sommersemester 2018

Prozessassessment MS3

Team:

Jan Omar Mehr

Armin Weinrebe

Mentor:

Robert Gabriel

Dozenten:

Prof. Dr. Gerhard Hartmann

Prof. Dr. Kristian Fischer

Inhaltsverzeichnis

1 Reflexion des Projektes

Mit der Erarbeitung des Projektplans, sollten die menschzentrierten und systemspezifischen Modelle untereinander koordiniert und passend abgestimmt werden, was im Allgemeinen während des Projekts durch gemeinsames Arbeiten an elementaren Modellen und Absprachen gut funktioniert hat. Aufgrund der umfangreichen Ausarbeitung der Anforderungen, ist eine sowohl tiefe, wie auch breite Modellierung der benötigten Klassen, Datenstrukturen und der ausgiebigen Kommunikation der Systemkomponenten die Folge. Das Hauptaugenmerk liegt dabei in der Erreichung der im zweiten Meilenstein neu definierten Ziele. Der gesamte Umfang des intendierten Systems wurde in Bezug auf den zeitlichen Rahmen hoch skaliert. Es sind Mängel in der Modellierung nach Feedback-Terminen aufgedeckt worden, die zu Iterationen und teilweise zu Neuerarbeitungen führten, sodass unvorhergesehene Neuplanungen im zweiten Meilenstein notwendig waren.

1.1 Herausforderungen in der Modellierung

Beim Festlegen und Verstehen des Nutzungskontexts, bestand zunächst das Problem, dass keine Probanden gefunden wurden, die bereit sind einen Termin für ein Interview wahrzunehmen. Im Nachhinein wurde in einem Audit besprochen, man hätte für das Projekt Personen aus einem ähnlichem Nutzungskontext befragen können. Stattdessen wurden jedoch Sekundärquellen recherchiert und daraufhin Benutzermodelle erarbeitet. Diese wurden wiederum auf der Grundlage von später durchgeführten Interviews iteriert. Die Gefahr an dieser Stelle war, dass eine Fülle an Informationen in die Arbeit mit einfloss, die für die Erreichung der Ziele nicht notwendig waren, sodass der Aufwand zu viel Zeit in Anspruch nahm. Gegenstand dessen war beispielsweise die Erarbeitung kultureller Unterschiede. Aus den Recherchen und den Interviews ergab sich letztendlich, dass das Hauptproblem in Missverständnissen durch die Sprachbarriere bestehen und sich weitere soziale Probleme auf Grundlage ungerechten Verhaltens beziehen. Vorteilhaft waren die Erkenntnisse, die z.B. aus Szenarien gewonnen werden konnten. Hier konnte der Problemraum stark erweitert werden und hat gezeigt, wie sehr das intendierte System erweiterbar ist. Im Umkehrschluss bedeutet es jedoch, dass neue Erfordernisse entstanden sind, sodass gezielt entschieden werden musste, welche Priorität hieraus bezogene neue Anforderungen im Projekt haben sollen. Des Weiteren konnte eine Vorstellung der Aufgaben für die Ausarbeitung von konkreten Arbeitsschritten genutzt werden. Die Anforderungen wurden sehr breit ausgearbeitet und häufiger iteriert, sodass sie Technologie- und Lösungsunabhängig dargestellt sind. Bei der Erarbeitung der Gestaltlösungen wurde vom Projektplan abgewichen. Hier war geplant nach den Dialoggestaltungen und einem Detailed User Interface Mockups und Clickdummies zu entwickeln. Jedoch ist die Erstellung einer geeigneten Navigation in dieser Planung nicht berücksichtigt worden, die zwingend notwendig ist und es wurde entschieden einen weiteren Entwicklungsschritt durchzuführen: Die Entwicklung von Wireframes, als klare visuelle Darstellung des Layouts. Diese Entscheidung schien in diesem Moment sinnvoll, ist aber in Anbetracht des

zeitlichen Rahmens von Nachteil gewesen. Die Evaluation konnte so lediglich an den Wireframes durchgeführt werden, in Form der zehn Heuristiken nach Jakob Nielsen. Der geplante Cognitive Walkthrough blieb aus.
Im Ersten Meilenstein wurde das intendierte System in Form einer Rest-Architektur geplant. Dies wurde im zweiten Meilenstein auf den Prüfstand gestellt und mit einer Architektur mit Websockets verglichen, um eine mögliche Fehlplanung auszuschließen. In der REST-konformen Systemarchitektur wurde im ersten Meilenstein die Node.js-Middleware Faye dargestellt. Auch in der Durchführung des ersten PoC, in dem der noch sehr einfach gehaltene Abrechnungsalgorithmus getestet wurde, ist Faye verwendet worden. Im zweiten Meilenstein beginnen erste Entwicklungen am Android Client und es wurde dabei deutlich, dass Faye in Verbindung zu diesem unmöglich ist, da dies ein Node.js-Modul ist und in einer Android-Umgebung nicht existiert. Hier musste also neu angesetzt werden und als alternative wurde der Push-Service FCM (Firebase Cloud Messaging) in die Systemarchitektur übernommen, wie in der Modellierungsbegründung (Kapitel 6.1) beschrieben ist. Das zugehörige Klassendiagramm stellt den gesamten, komplexen Umfang des intendierten Systems dar. Für eine übersichtlichere Modellierung hätte es sich angeboten es in mehrere Diagramme aufzuteilen und für den vertikalen Prototypen ein separates ER-Diagramm ergänzen können. Abschließend wurde der Abrechnungsalgorithmus erweitert, sodass er für den Automatisierungsprozess benötigte Bedingungen berücksichtigt. Dies wurde in Pseudocode dargestellt und als Vorlage für die Implementation genutzt. Hier wurde ein bewusstes Risiko eingegangen, da es vor der Implementation nicht getestet wurde.

1.2 Herausforderungen in der Programmierung

Da die Programmierkenntnisse der Entwickler auf einem sehr unterschiedlichen Stand sind und dies im Vorfeld nicht deutlich kommuniziert wurde, sind die geplanten Features des vertikalen Prototypen sehr hoch skaliert worden. Daher konnte nicht alles implementiert werden, was in M1 und M2 geplant wurde.
Bei der Programmierung des Clients lag die Herausforderung darin, die Daten des Servers in einer visuell ansprechenden und übersichtlichen Art und Weise darzustellen. Dafür wurden in M2 detaillierte Wireframes entwickelt, welche in der Programmierung sehr genau umgesetzt wurden. Aufgrund der Komplexität der darzustellenden Daten mussten eigene 'Adapter' aus 'Vieholdern' erstellt werden. Dies beinhaltete eine ausführliche Recherche und relativ viel Zeit bei der Implementation und Durchführung der Tests. Letztendlich konnte dies jedoch erfolgreich umgesetzt werden und die, in der Modellierungsbegründung (Kapitel 4.2) formulierten, Prüfkriterien wurden in den implementierten Aktivitäten und Fragmenten beachtet und eingehalten.
Eine weitere Herausforderung war die Implementierung der Optical Character Recognition Funktion. Der Scan eines Kassenzettels und die erfolgreiche Erfassung der Daten ist zwar möglich, aber da die Daten als zusammenhängender String erfasst werden, war es den Entwicklern nicht möglich, im Zeitrahmen von M3, die gewonnen Daten in ein

JSON-Format umzuwandeln, mit dem der Server arbeiten kann. Daher wurde die Möglichkeit zur manuellen Eingabe von Kassenzettel-Daten implementiert. Dort sind jedoch noch einige Fehler vorhanden, die aus Zeitmangel noch nicht behoben werden konnten. So ist es z.B derzeit nicht möglich für verschiedene Artikel unterschiedliche Teilnahmewerte zu speichern. Der Client überschreibt diese für jeden Artikel.
Bei der Programmierung des Servers lag die Herausforderung darin, die komplexen, verschachtelten Kassenzetteldaten für die Abrechnung auslesen und weiterverarbeiten zu können. Bisher wurde dies mit forEach-Schleifen implementiert. Dies ist jedoch nicht besonders performant. Zudem gab es dort Probleme mit der asynchronen Funktionsweise der Node.js-Engine. Da diese nicht wartet, bis vorhergehende Funktionen beendet sind (es sei den man verwendet 'Promises'), mussten synchrone Funktionen in 'Callbacks' verschachtelt werden. Dies ist jedoch kein besonders guter Programmierstil, da es zum einen sehr unleserlich wird und die Performanz beeinträchtigt. Eine bessere Lösung hätte das Modul 'async' geboten, welches für die Vereinfachung der Callbacks entwickelt wurde. Leider fehlte den Entwicklern die Erfahrung und die Zeit dies bis zur Abgabe von M3 umzusetzen. So sind noch eine Fehler in der Abrechnungslogik vorhanden: Eine Abrechnung eines Kassenzettels mit mehreren Artikel, führt zu falschen Abrechnungswerten. Dies liegt an der unkontrollierten Abfolge der forEach-Schleifen. Abrechnungen von Kassenzetteln mit nur einem Artikel funktionieren jedoch fehlerfrei.