

Entwicklung einer plattformunabhängigen Chatbot-Frontend-Anwendung

Simon Ludwig

Technische Hochschule Mittelhessen
Fachbereich MND
Wilhelm-Leuschner-Str. 13
61169 Friedberg
E-Mail: simon.ludwig@mnd.thm.de

Prof. Dr. Harald Ritz

Technische Hochschule Mittelhessen
Fachbereich MNI
Wiesenstraße 14
35390 Gießen
E-Mail: harald.ritz@mni.thm.de

Kategorie

Masterarbeit

Schlüsselwörter

App-Entwicklung, Chatbot, Continuous Integration (CI), Continuous Delivery (CD), Frontend-Entwicklung, KI, Plattformunabhängigkeit, Prototyping, Testautomatisierung

Zusammenfassung

Für die Technische Hochschule Mittelhessen, bei der das Leitbild auf der Bewältigung technischer Herausforderungen unter Verwendung neuester Erkenntnisse ausgerichtet ist, griff der Prüfungsausschuss-Vorsitzende der Studiengänge B.Sc. und M.Sc. Wirtschaftsinformatik im Jahre 2021 den Trend der Chatbots auf. Gemeinsam mit Master-Studierenden der Wirtschaftsinformatik leitete er die Entwicklung von „Winfy“, einem Chatbot, der Fragen von Studierenden bezüglich Prüfungsangelegenheiten einfach und schnell beantworten sollte, ein.

Oggleich ebenjener Chatbot Fragen der Nutzer ausgesprochen präzise in seiner vorliegenden Version beantwortet, ist dessen Nutzung zum Zeitpunkt des Verfassens der Abschlussarbeit nur auf Endgeräten mit großen Bildschirmen bzw. einer hohen Auflösung möglich. Das fehlende responsive Verhalten sorgt dafür, dass die Anwendung auf mobilen Endgeräten bisher mit Einschränkungen bedient werden kann. Die seit Jahrzehnten wachsende Nutzung von mobilen Endgeräten wie Smartphones und Tablets zwingt Entwickler und Betreiber von Anwendungen zum Umdenken. Denn es ist davon auszugehen, dass sich dieser Trend fortsetzen, gar intensivieren wird. Durch neue Technologien wie der Virtual-Reality-Brille oder Smartwatches wird der sowieso schon sehr diverse Endgeräte-Markt noch vielfältiger werden.

Ziel der Masterarbeit war es daher, eine plattformunabhängige Anwendung zu entwickeln, die auf allen gängigen Endgeräten lauffähig sein sollte – also sowohl auf Smartphones und Tablets als auch Desktop-Geräten und Notebooks. Neben der Plattformunabhängigkeit umfassten die Ziele weitere Anforderungen, die sich aus der vorherigen Anwendung, dem ISO 25010 und aktuellen Gesetzestexten ableiten ließen.

So sollte neben einer barrierefreien Anwendung, eine mehrsprachige, responsive Oberfläche entstehen. Zudem wurde sich auf die langfristige Wartbarkeit, die Wirtschaftlichkeit der Entwicklung, den Betrieb und die Benutzerfreundlichkeit fokussiert.

Mit den gesetzten Zielen und Rahmenbedingungen wurden mögliche Entwicklungsansätze ermittelt, mit deren Hilfe die Anwendung entwickelt werden konnte. Die möglichen Entwicklungsansätze umfassten dabei den nativen, Cross-Plattform, Hybriden und Web-Entwicklungsansatz. Nach einer umfangreichen Nutzwertanalyse, die alle Interessengruppen, Ziele und Anforderungen berücksichtigten, erwies sich die Entwicklung einer Webanwendung als am sinnvollsten.

Der native Entwicklungsansatz stellte sich als ungeeignet heraus, da für die Entwicklung separate Quellcodes für jedes zu unterstützende Betriebssystem notwendig gewesen wären. Nicht nur die initiale Entwicklung, sondern auch die Wartung mehrerer Projekte hätte einen hohen zeitlichen Aufwand zur Folge gehabt.

Da der Web-Entwicklungsansatz im Vergleich zu der Cross-Plattform- und hybriden Entwicklung einen geringeren Wartungsaufwand, eine schnellere Implementierung, geringere Einstiegshürden und den Aufbau einer CI-/CD-Pipeline ermöglicht, wurde dieser als am sinnvollsten betrachtet. Nachdem ein adäquater Entwicklungsansatz eruiert wurde, sollte die Entwicklung der Anwendung nutzerorientiert erfolgen. Dies mit dem Ziel, den Nutzer bei allen Entscheidungen in den Mittelpunkt zu stellen.

Auf Grundlage der vorherigen Version des Chatbots und der Anforderungen aus dem Kontext der Prüfungsangelegenheiten wurden die Anwendungsfälle ausgearbeitet. In einem durch das Prototyping angepassten Wasserfallmodell wurden auf dieser Grundlage Prototypen der Anwendung entworfen. Durch dieses Vorgehen konnte bereits früh Feedback zur Oberfläche eingeholt und in die Entwicklung einbezogen werden.

Die Oberfläche wurde dabei unter Berücksichtigung aktueller Studien und Untersuchungen aufgebaut. So war es ein Ziel, die kognitive Belastung bei der Bedienung durch den Nutzer zu minimieren. Dabei wurden u.a.

Konzepte wie die Daumen-Zone und die Daumen-Impact-Area aufgegriffen. Weiterhin wurde sichergestellt, dass eine leichtgewichtige Anwendung entsteht, die sowohl mit einer schnellen Lade- als auch Bediengeschwindigkeit aufwartet. Mehrere Studien und Untersuchungen zeigen die Wichtigkeit schnell reagierender Anwendungen auf. Eines der daraus abgeleiteten Ziele ist es daher, die Schwellwerte einzuhalten. Mit einer initialen Ladezeit von etwa 700 Millisekunden und einer Ladezeit beim wiederholten Besuch von 200 Millisekunden, wurden die gesetzten Ziele weit übertroffen. Damit die Zufriedenheit der Interaktion mit dem Chatbot gemessen werden kann, wurde die Oberfläche um das Bewerten der Antworten erweitert.

Ein mit der Benutzerfreundlichkeit eng verwobenes Konzept stellte die Qualitätssicherung der Anwendung dar. Dieses spielte bei der Entwicklung eine entscheidende Rolle. Eine minderwertige Anwendung wirkt sich so nicht nur negativ auf das Benutzererlebnis, sondern auch auf die Wartbarkeit, die Wirtschaftlichkeit und die Plattformunabhängigkeit einer Anwendung aus. Eine Ausfallzeit kann beispielsweise nicht nur imageschädigende Auswirkungen zur Folge haben, sondern beeinträchtigt auch das Benutzererlebnis. Auch Laufzeitfehler im Quellcode würden dazu führen, dass die Anwendung nachträglich korrigiert werden müsste. Es galt daher sowohl während, als auch nach der Entwicklung qualitätssichernde Maßnahmen zu ergreifen, um Fehler zu vermeiden und diese proaktiv behandeln zu können.

Für diesen Zweck wurden neben automatisierten Tests und statischen Quellcodeanalysen auch Akzeptanztests entwickelt. Mithilfe der statischen Quellcodeanalyse, der Unit- und End-To-End-Tests sollten Fehler bereits vor der Inbetriebnahme einer neuen Version gefunden werden. Zur Reduzierung der Auswirkungen eines Ausfalls des Systems wurde zusätzlich ein Monitoring-System implementiert, das sowohl die Frontend- als auch die Backend-Anwendung kontinuierlich überwacht. Laufzeitfehler der Anwendung sollen so schnell erkannt und entsprechende Maßnahmen eingeleitet werden. Im Fall eines Fehlers sollte der Betreiber durch ein Ticket benachrichtigt werden.

Damit Änderungen schnell und unkompliziert eingepflegt und in Betrieb genommen werden können, bestand ein weiterer Aspekt in der Implementierung einer CI-/CD-Pipeline. In Verbindung mit einem dazu passenden Branching-System und automatisierten Testfällen wurde so ein Prozess entwickelt, der sicherstellt, dass ausschließlich Änderungen in Betrieb genommen werden, die vorher getestet wurden und den Qualitätsansprüchen genügen.

Die entwickelte Pipeline wurde so konfiguriert, dass sicherheitsrelevante Schlüssel und Passwörter innerhalb des Quellcodeverwaltungssystems, nicht aber innerhalb des Projektes, gespeichert werden.

Fazit

Die Entscheidung, eine Webanwendung zu entwickeln, ermöglichte es, die Anwendung plattformunabhängig zu entwickeln. Endgeräte mit einem Browser ermöglichen den barrierefreien Zugang zum Chatbot. Sowohl Smartphones, Tablets, Laptops als auch Desktop-PCs unterstützen die Anwendung. In über 20 Tests wurde bewiesen, dass die Anwendung auf modernen Browser funktionsfähig ist. Die Anwendung wurde auf den o.g. Geräten mehreren Tests unterzogen.

Die multilinguale, responsive Oberfläche, die Einhaltung von gängigen Barrierefreiheitsregeln und das Implementieren einer Text-To-Speech- und Dark-Mode-Funktionalität ermöglichen es, dass die Anwendung auch von Menschen mit körperlichen Einschränkungen genutzt werden kann. Durch die entwickelte CI-/CD-Pipeline wird sichergestellt, dass die Anwendung leicht zu warten ist, die Software den Qualitätsrichtlinien entspricht und neue Versionen einfach und schnell in Betrieb genommen werden können. Durch eine entsprechende Dokumentation und Installationsanweisungen wurde zudem sichergestellt, dass die Anwendung von zukünftigen Entwicklern unkompliziert gewartet und verwaltet werden kann. Das nutzerzentrierte Vorgehen stellte die Nutzer des Chatbots bei allen Entscheidungen in den Mittelpunkt. Das Ergebnis dieses Vorgehens ist eine Anwendung, die sich durch eine einfach zu bedienende Oberfläche und eine sehr schnelle Lade- und Bediengeschwindigkeit auszeichnet.

Mit der Entwicklung einer modernen, plattformunabhängigen Webanwendung wurden Hürden in der Bedienung des Chatbots abgebaut und die Anwendung einem noch breiteren Publikum zugänglich gemacht. Die Anwendung kann sich nun zu einem zentralen Anlaufpunkt bei Fragen rund um das Studium entwickeln.

Literatur

Baxter, Kathy; Catherine Courage; Kelly Caine: Introduction to User Experience. in: Understanding your Users. Waltham USA. Elsevier Science (2015)

Boehm, Barry; Basili, Victor: Software Defect Reduction Top 10 List. in: IEEE Computer 34. Washington, D.C USA. IEEE Computer Society Press (2001).

França, Joyce; Soares, Michel S.: SOAQM: Quality Model for SOA Applications based on ISO 25010. Barcelona Spain. SciTePress (2015)

Ritz, Harald; Tansel, Dogus: Entwicklung eines KI-basierten FAQ-Chatbots für die Hochschule im Bereich Prüfungsangelegenheiten, in: AKWI, 17 / 2023, S.81-92, DOI: <https://doi.org/10.26034/lu.akwi.2023.n17>