

FORTGESCHRITTENENPRAKTIKUM

Portierung der Bavarian App in Ionic

Patrick Frömel, Tim Romonath

27. Oktober 2021

Inhaltsverzeichnis

1	Kurzbeschreibung	2
1.1	Zur Portierung	2
1.2	Das Team	3
2	Entwicklungsprozess	5

1 Kurzbeschreibung

Die App „Welcome to Bavaria“ in der Version 3.0 ist eine Portierung der gleichnamigen Version 2.0 auf das Ionic-Framework um diese auf verschiedenen Betriebssystemen - insbesondere iOS und Android - mit geringen Entwicklungsaufwand zur Verfügung zu stellen. Die App wurde dabei im Zeitraum von Juli bis Oktober 2021 Entwickelt.

Die App beruht auf dem gedrucktem Heft „Migraboarisch - Von uns fia eich. Ein Wegweiser durch den Dialekt“ welcher im Schuljahr 2016/17 von Kelheimer Schülern Entwickelt wurde. Ziel war es dabei Neuankömmlingen in Bayern den Bayrischen Dialekt näher zu bringen.

Die App wurde in der Version 1.0 von Marburger Studenten 2019/20 entwickelt und um verschiedene Sprachaufnahmen ergänzt. Für die Version 2.0 wurden weitere Sprachaufnahmen und ein primitives Archivment-System eingebracht.

1.1 Zur Portierung

Es wurde sich für Ionic mit dem Angular-Framework entschieden um die Vorteile der Webentwicklung bezüglich Ihrer Anpassungsfähigkeit in verhältnismäßig einfachen Frameworks zu nutzen und damit die Reichweite der App zu erweitern. Sogenannte „Web-Apps“ sind weit verbreitet und stellen - vereinfacht ausgedrückt - Websites in speziellen Browsern dar so dass der Quellcode auf unterschiedlichen Betriebssystemen stets der selbe bleiben kann. So müssen bei der Wartung nicht verschiedene Programme parallel Entwickelt werden.

Ionic mit Angular wird in dieser Kombination von quasi allen Seiten propagiert und wurde in unserer unbedarftheit dankend angenommen. Ionic stellt viele mehr oder minder komplexe Bausteine zur Webentwicklung bereit (von Textfeldern über Knöpfen bis hin zu Slidern mit Animation) während Angular zum einen HTML um viele nützliche „Funktionen“ erweitert (Loops, If-Else, . . .) und zum anderen komplexe Mechanismen im Hintergrund verwaltet so dass beispielsweise eine Website mit verschiedenen wechselnden Komponenten hoch dynamisch gestaltet werden kann.

In der Portierung wurde versucht sich an S.O.L.I.D-Prinzipien zu halten, auch wenn dies mit Type-Script als Programmiersprache bisweilen kompliziert gestaltete. Während die Datenbank-Mechanismen noch verhältnismäßig gut ohne Abhängigkeiten funktionieren und dank Schnittstellen (Interfaces) auch einfach ausgetauscht werden können sind die Seiten der UI sehr eng an verschiedene „Services“ gebunden welche wiederum untereinander stark abhängig sind. Dies ist aber keine Notwendigkeit und das refactoren dieser kann Gegenstand eines Nachfolgeprojekts sein. Kern der Bemühungen ist ein „Request-Response“-Pattern in welchen ein Request über ein Interactor gestellt wird welcher wiederum von den Datenbankmechanismen eine Response-Methode aufrufen. Sehr großer Vorteil ist hier das sich die asynchrone Programmierung in Grenzen hält - eine Aufgabe kann und wird auch erst dann zuende gebracht wenn eine Antwort vorhanden ist.

TODO DATENBANKTECHNOLOGIE

1.2 Das Team

Leitung

Hanna Fischer

Koordination und Betreuung

Hanna Fischer, Christoph Bockisch, Peter Kaspar

Systementwicklung

Version 3.0: Patrick Frömel, Tim Romonath

Version 2.0: Lea Fischbach

Version 1.0: Benedikt Batton, Lester Brühler, Lea Fischbach, Hannah Greß,
Anh-Thy Truong, Qi Yu

Mitarbeit

Milena Gropp

Sprachaufnahmen

Melanie Bajrami – Regensburg (Nordmittelbairisch)

Christina Böhmländer – Zwiesel (Nordmittelbairisch)

Felicitas Erhardt – Walleshausen (Schwäbisch-Mittelbairisches Übergangsgebiet)

Christian Ferstl – Regensburg (Nordmittelbairisch)

Edith Funk – Krumbach (Schwäbisch)

Markus Kunzmann – Mühldorf am Inn (Mittelbairisch)

Barbara Neuber – Altenstadt an der Waldnaab (Nordbairisch)

Michael Schnabel – Bayreuth (Ostfränkisch)

Übersetzung

Arabisch: Clara Mikhail

Englisch: Jeffrey Pheiff

Tschechisch: Andrea Königsmarková

Grafiken

Udo Butler

Beteiligte Lehrerinnen und Schülerinnen

Peter Kaspar, Daniela von Schultz

Karam Alayoubi, Sali AlBoulad, Yazan Aljleilati, Ahmad AlShalabi, Mustafa AlShalabi, Arreeya Banlo, Anita Blaha, Nicolas Bruckmaier, Florian Fahle, Nico Fichte, Lukas Fischer, Tina Föhre, Nico Horn, Aisa Jusic, Alexej Kobzev, Karina Kluger, Max Kohlmeier, Samuel Konschelle, Damian Kowalski, Andreas Lening, Laura Lohwasser, Markus Müller, Felix Neuhauser, Franziska Nigl,

David Nguyen, Samuel Noy, Daniel Reißmann-Toledo, Lena Ritzinger, Anna Ro-
idl, Joshua-David Roland, Heba Sandafi, Julian Schanderl, Andreas Stemmer,
Wahyu Susilo, Liviu Vasiu, Mihai Vasiu, Maria Waldhier, Markus Weber, Ste-
fanie Weber, Amadeus-Daniel Wiselka, Burak Yilmaz, Andreas Zellner, Tobias
Zinkl, Thomas Zirmer

2 Entwicklungsprozess

Der Entwicklungsprozess selbst wurde zunächst keiner komplexen Systematik unterworfen. Es gab die Trennung, dass Patrick Frömel die Datenbankmechaniken und Interfaces inkl. Interactor entwickelt während Tim Romonath sich in Ionic und Angular einarbeitet und das User-Interface entwickelt.

Die ersten 5 Wochen haben wir an den Wochentagen in den Semesterferien uns morgens getroffen, den aktuellen Stand besprochen und dann 6 - 8 Stunden entwickelt. Anfangs haben wir mit Agantty (Webtool zur Aufgabenplanung) gearbeitet, aber schnell festgestellt dass dies nicht unseren Bedürfnissen entsprach. Danach folgte eine Pause von 4 Wochen in welcher wir anderen Dingen nachgehen mussten. Nach dieser Pause haben wir nur schwer wieder den Einstieg in das Projekt gefunden. Zur Unterstützung haben fortan mit Trello gearbeitet, ein einfaches Webtool für Kanban-Boards.

2.1 Agiles vorgehen

Unser vorgehen kann am ehesten noch als „agil“ beschrieben werden. Wir haben die zugrundeliegende App Stück für Stück nachgebaut - zunächst einen Funktionierenden Kern (Datenbankmechaniken, auslesen und übertragen dieser in die UI - Soundwiedergabe, grundlegendes Routing usw.), dann wichtige Kernfeatures (Bilder, Fortschritt, Archievments, usw.) und nach und nach alle anderen implementierten Features (Dialektauswahl, App zurücksetzen, usw.).

Allerdings ist unser Vorgehen nicht besonders inkrementell und iterativ gewesen - eher modular chaotisch. Wir haben im Grunde gemacht was gerade anfiel und für den Großteil des Projekts auf Projektmanagement verzichtet. Das hat ob der geringen Komplexität gut funktioniert - ist aber für zukünftige Projekte nicht ratsam. Ein wirkliches, agiles Vorgehen kam erst am Ende mit Trello auf - dort wurden Aufgaben in einem Backlog gesammelt und nach und nach fertig gestellt. Allerdings gab es zu diesem Zeitpunkt nur noch ein Zwischenziel: Fertig werden.

2.2 Verwendete Tools

Agantty

Agantty ist ein Webtool um Projektmanagement zu unterstützen. Es implementiert dabei ein Gantt-Diagramm, welches zeitliche Aktivitäten als Balken darstellt. Wir haben Agantty sehr schnell verworfen, da unser prinzipiell agiles Vorgehen nicht gut mit dieser Art Diagram