



«Kirchenfeldrobotics» «Clever planen, bewusster essen»

Technische Informationen für die Jury



Technische Informationen für die Jury

Unser Team



(von links nach rechts) Nick (16), Jakob(17), Phileas(15), Valéry(15), Nils(16)

Wir sind Jakob, Nick, Nils, Phileas und Valéry. Zusammen besuchen wir dieselbe Klasse am Gymnasium Kirchenfeld, Bern im zweiten Jahr.

Für uns alle ist dies unser erster Hackathon, aber einige von uns haben bereits Wettbewerbserfahrungen an andere Technik Wettbewerben und Olympiaden.

An diesem Hackathon wollen wir vor allem Spass haben und eine neue Erfahrung haben. Es ist uns auch wichtig neues zu lernen und unsere skills zu verbessern und praktische Erfahrung zu sammeln.

Wir möchten dem Organsiationsteam und allen Mithelfenden danken für die Organisation dieses tollen Wettbewerbs, es hat mega spass gemacht!

Aktueller Stand des Sourcecodes

Link zu Repo: https://github.com/Kirchenfeldrobotics/hackathon_bern

Im Repository befindet sich unser Sourcecode im aktuellen Stand.

Leider konnten wir noch nicht alle Features detailliert ausbauen. Dennoch funktionieren sämtliche Kommunikationswege mit der API sowie mit unserem Raspberry-Pi-Server problemlos, und alle relevanten Dateien werden erfolgreich übertragen.

Ausgangslage

• Worauf habt ihr euch fokussiert?

Unser (Haupt-) Fokus lag darauf, eine Lösung zu entwickeln, die nicht nur während des Hackathons funktioniert, sondern auch darüber hinaus anwendbar ist.

Deshalb sind wir bewusst das Risiko eingegangen, einen eigenen Server auf Basis eines Raspberry Pi 5 zu entwickeln.

Diese Entscheidung bringt mehrere Vorteile mit sich:

- Die Berechnungen erfolgen serverseitig.
- Alle API-Tokens befinden sich ausschließlich auf dem Server.
(Wären sie in der App gespeichert, könnten sie durch Dekompilierung extrahiert werden.)

Letzteres ist ein gutes Beispiel dafür, wie wir unsere technische Lösung auf die Anforderungen der Industrie und Sicherheitsaspekte ausgerichtet haben.

Ein weiterer Fokus lag auf der Skalierbarkeit: Wir wollten eine Lösung schaffen, die in den Details weiterentwickelt werden kann, aber bereits eine funktionierende Grundlage bietet. Uns war wichtig, dass am Ende des Hackathons keine vollständig ausgereifte Lösung stehen muss, sondern ein solides Fundament für die Weiterentwicklung.

Auch unser Team war ein zentraler Aspekt: Drei Mitglieder verfügen über Wettbewerbserfahrung und umfangreiche Programmierkenntnisse. Zwei Teammitglieder nahmen zum ersten Mal an einem Wettbewerb teil und hatten bislang wenig praktische Erfahrung. Trotzdem entschieden wir uns bewusst für ein gemeinsames Team und legten großen Wert auf Integration und gemeinsames Lernen.

Gerade die weniger erfahrenen Mitglieder konnten stark profitieren, etwa durch das Kennenlernen des GitHub-Workflows oder durch die Erfahrung, wie man effizient im Pair Programming arbeitet.

• Welche technischen Grundsatzentscheide habt ihr gefällt?

Die wichtigste Entscheidung war die klare Trennung zwischen Frontend, Backend, Server und externen APIs.

Wir haben eine mobile App mit Flutter entwickelt, die Firebase nutzt, um Nutzerdaten zu speichern. Diese Lösung ist gut skalierbar: Benutzerkonten können angelegt und Menüs sowie Rezepte direkt mit dem jeweiligen Profil verknüpft werden.

Die Flutter-App kommuniziert mit unserem Raspberry-Pi-Server, der sämtliche API-Keys und Berechnungen übernimmt.

Unser Server basiert auf einem Raspberry Pi 5 und erfüllt drei zentrale Funktionen:

1. **Sicherheit:** API-Keys sind nicht in der App gespeichert und somit vor Dekompilierung geschützt.
2. **Effizienz:** Rechenintensive Filter-Skripte laufen serverseitig, was die App ressourcenschonend und zugänglicher macht.
3. **Datenreduktion:** Der Server verarbeitet die Daten der externen APIs direkt und sendet nur bereinigte, strukturierte Informationen an die App.

Wir haben uns für zwei APIs entschieden:

- **Google Vision API:** Führt Objekterkennung auf Bildern durch und liefert Ergebnisse und Wahrscheinlichkeitswerte.
- **TheMealDB:** Eine externe Rezeptdatenbank mit Filterfunktionen, welche die Grundlage für unsere Menüplanung bildet.

Die Daten beider APIs werden direkt auf dem Server verarbeitet und in Dictionaries umgewandelt, sodass nur „cleane“ Daten an die Flutter-App gesendet werden.

Technischer Aufbau

• Welche Komponenten und Frameworks habt ihr verwendet?

- Raspberry Pi 5 als Backend-Server
- Firebase zur Verwaltung der Benutzerkonten (Speicherung der Nutzerdaten)
- Flutter für die mobile App
- Flask als Schnittstelle zwischen Server-Skripten und Frontend Flutter
- Google Vision API zur Objekterkennung
- TheMealDB als externe Rezeptdatenbank

• Wozu und wie werden diese eingesetzt?

- Der Raspberry Pi führt alle Skripte und Berechnungen aus (in Python).
- Flask dient als Brücke zwischen den Server-Skripten und der App.
- JSON wird für die gesamte Kommunikation verwendet, nur bereinigte Daten (in Dictionary-Form) werden an das Frontend übermittelt.
- Firebase speichert Nutzerdaten wie Lieblingsrezepte oder das Zutateninventar.
- Die Google Vision API erkennt Objekte auf Bildern und hilft beim automatisierten Einspeichern von Zutaten.
- TheMealDB liefert Rezepte, Zutatenlisten, Bilder und weitere Details.

Implementation

• Gibt etwas Spezielles, was ihr zur Implementation erwähnen wollt?

Wir sind bewusst das Risiko eingegangen, einen eigenen Server zu entwickeln – und haben es geschafft, alle Komponenten erfolgreich zu verbinden. Dadurch haben wir eine solide Grundlage geschaffen, auf der sich weitere Features effizient aufbauen lassen.

• Was ist aus technischer Sicht besonders cool an eurer Lösung?

Unsere Entscheidung für einen eigenen Server war mutig, aber sie hat sich bewährt. Die Lösung ist modular, sicher und skalierbar und damit bestens geeignet für eine Weiterentwicklung über den Hackathon hinaus. Mit dieser Entscheidung haben wir eine solide, anwendbare Lösung gefunden.

Abgrenzung / Offene Punkte

• Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert? Weshalb?

Wir haben bewusst weniger Fokus auf das UI-Design und die User Experience gelegt. Stattdessen lag unser Schwerpunkt auf der technischen Basis und der funktionalen Integration. Wir wollen eine Basis schaffen auf den zukünftigen Lösungen aufbauen können.

• Beispiel:

Zwar sind alle Skripte bereits in Flutter eingebunden und liefern korrekte Ergebnisse zurück, jedoch fehlen noch Buttons, Submenus und andere UI-Elemente zur vollständigen Nutzung. Da die technische Grundlage steht, können diese Details jedoch schnell und gezielt ergänzt werden.

[illegible][illegible]

BÄRNPÄKT 2024