

PRAXISPROJEKTE 2025

BACHELOR
SOFTWARE ENGINEERING
UND VERNETZTE SYSTEME



Fachhochschul-Bachelorstudiengang

Software Engineering und vernetzte Systeme



*„Egal, in welchen Bereich des Lebens man blickt, Software ist ein wesentlicher Bestandteil jeder Innovation. Unsere Absolvent*innen sind so der Schlüssel zum Erfolg neuer Lösungen.“*

Franz Knipp, Studiengangsleiter

Das Bachelorstudium verfolgt das Ziel, Software-Entwickler*innen auszubilden, die den Entwicklungsprozess von der initialen Idee bis zur Inbetriebnahme verstehen und auf Basis aktueller Technologien umsetzen können. Neben der Vermittlung des theoretischen Wissens bildet die praktische Umsetzung einen wichtigen Bestandteil der Ausbildung.

„Softwareentwicklung ist mehr als Programmieren. Organisations- und Kommunikationskompetenzen machen ein Projekt erst erfolgreich.“

Christian Büll, Departmentleiter



Im vierten Semester beweisen die Studierenden die erworbenen Kompetenzen im Rahmen eines mehrmonatigen Projekts. Neben den technischen Fertigkeiten der Softwareentwicklung und -architektur sind die Teams auch für das Management des Projekts und die Kommunikation mit der Auftraggeber*in zuständig.

Dieses Projekt wird übergreifend in den Lehrveranstaltungen Praxisprojekt, Softwarequalität und Projektmanagement II umgesetzt.

Inhalt

1	Praxisprojekte	2
1.1	gAmlcode	2
1.2	Elektroauto Lade-App	5
1.3	I NEED HELP	8
1.4	VoiceArt	10
1.5	Atlas der Klima-bezogenen-Gesundheitsrisiken	13
1.6	NEXUS	16
1.7	Pegelhub	19
1.8	Klima-Gesundheitsplanspiel Burgenland	22
2	Projektmitglieder – BSWE Jahrgang 2023	25
3	Projektbetreuer*innen	28
4	Verzeichnisse	29
5	Impressum	31

gAmlcode



Projektauftraggeber*in: Hochschule Burgenland GmbH
Projektbetreuer*in: Dipl.-Ing. Franz Knipp
Projektmitglieder: Nina Hohl, Lukas Hofer, Tobias Pavel, Leon Rasner
Repository: <https://github.com/fhburgenland-bswe/gamification>

*gAmlcode ist eine Lernplattform mit Gamification-Elementen, die im vergangenen Jahr von einem Projektteam entwickelt wurde. Das ursprüngliche Projekt konzentrierte sich primär auf die Umsetzung zentraler Funktionen aus Sicht der Studierenden, um deren Lernerfahrung zu verbessern und sie spielerisch zu motivieren. Im diesjährigen Projekt wurde der Fokus erweitert, um die Plattform zukünftig auch für Lehrende effektiv nutzbar zu machen. Hierzu wurde eine umfassende Lektor*innen-Ansicht entwickelt, die speziell auf die Bedürfnisse und Anforderungen der Lehrenden zugeschnitten ist und unter anderem die Verwaltung von Studierenden sowie das Erstellen und Bearbeiten von Lerninhalten ermöglicht.*

Auftraggeberin

Die Hochschule Burgenland ist eine öffentliche Bildungseinrichtung des Landes Burgenland und wurde 1994 gegründet. Wie bereits im Vorprojekt fungierte *Dipl.-Ing. Patrizia Sailer BSc, MSc, MBA* als Auftraggeberin. Die Plattform ist für sie ein persönliches Herzensprojekt, das sie von Anfang an aktiv begleitet.

Die neu entwickelte Lektor*innen-Ansicht stellt für Frau Sailer eine zentrale Erweiterung dar. Sie nutzt die Plattform bereits selbst im Rahmen ihrer Lehre und möchte diese Funktionalitäten nun auch weiteren Lehrenden zur Verfügung stellen.

Projektbeschreibung

Mit der Entwicklung der Lektor*innen-Ansicht wurde die bestehende gAmlcode-Plattform um wesentliche und kritische Funktionalitäten ergänzt. Diese Erweiterungen zielen darauf ab, die tatsächliche Nutzung der Plattform für Lehrende langfristig zu ermöglichen und die Akzeptanz deutlich zu steigern. Die Implementierung erfolgte auf Grundlage des bestehenden Quellcodes und nutzt den etablierten Tech-Stack, bestehend aus dem Dokumenten-basierten Datenbanksystem MongoDB, dem Web-Framework MeteorJS und dem Frontend-Framework React.

Eine der zentralen Erweiterungen dieses Projekts war die Einführung eines differenzierten Rollensystems, das klar zwischen Studierenden, Lehrenden und Administrator*innen unterscheidet. Dieses Rollensystem gewährleistet eine strukturierte Trennung der Nutzerrechte und ermöglicht eine spezifische, Rollen-basierte Darstellung der relevanten Inhalte und Funktionen innerhalb der Benutzeroberfläche.

Die Lektor*innen-Ansicht gliedert sich in drei zentrale Komponenten:

- Benutzerverwaltung
- Inhaltsverwaltung
- Reporting

Im Rahmen dieses Projekts wurden die Benutzerverwaltung und die Inhaltsverwaltung umgesetzt.

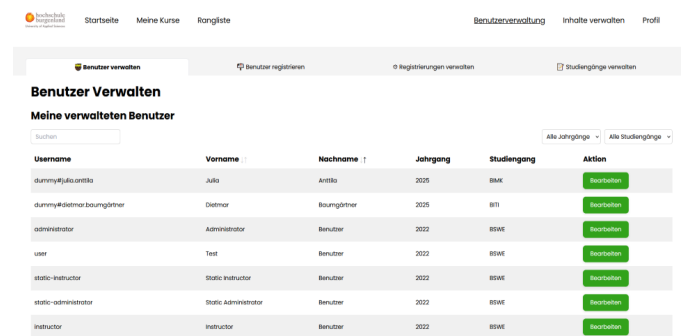
Projektverlauf

Das Projekt begann mit einer umfassenden Planungsphase. Die Anforderungen an die Lektor*innen-Ansicht wurden von der Auftraggeberin detailliert definiert, was die Formulierung konkreter User Stories erleichterte. Bereits in dieser Phase wurden zahlreiche Design-Mockups und Click-Dummies mithilfe von *Figma* erstellt. Das bestehende Plattform-Design konnte für die Lektor*innen-Ansicht übernommen und angepasst werden, sodass kein vollständig neues Design entwickelt werden musste.

Auf die Planungsphase folgten fünf Sprints, in denen die Benutzer- und Inhaltsverwaltung umgesetzt wurden. Die Implementierung des Reportings war ursprünglich ebenfalls vorgesehen, wurde jedoch in Absprache mit der Auftraggeberin als nachrangig eingestuft und daher aus dem Projektumfang gestrichen.

In den ersten drei Sprints lag der Fokus auf der Benutzerverwaltung. Zunächst wurde eine technische Basis geschaffen, insbesondere durch die Im-

plementierung eines globalen Rollenkonzepts sowie entsprechender Anpassungen der zugrundeliegenden Datenstrukturen. Zugleich wurden zentrale Funktionen entwickelt, etwa das Bearbeiten von Benutzerprofilen, das Einladen neuer Benutzer*innen mittels Link oder E-Mail, die Verwaltung der Einladungen sowie die Verwaltung von Studien- und Jahrgängen. Parallel wurden auch bereits Grundlagen für die Kursverwaltung geschaffen um die folgenden Projektphasen zu erleichtern. Etwa durch Anpassungen der Datenbankstruktur und die Erstellung erster Formularelemente in der Benutzeransicht.



Username	Vorname	Nachname	Jahrgang	Studiengang	Aktion
dummy@julia.cantla	Julia	Artlitz	2025	BMK	Benutzer
dummy@dietmar.boaumgöhrner	Dietmar	Boaumgöhrner	2025	BITI	Benutzer
administrator	Administrator	Benutzer	2022	BSWE	Benutzer
user	Test	Benutzer	2022	BSWE	Benutzer
static-instructor	Static Instructor	Benutzer	2022	BSWE	Benutzer
static-administrator	Static Administrator	Benutzer	2022	BSWE	Benutzer
instructor	Instructor	Benutzer	2022	BSWE	Benutzer

Abbildung 1: Übersicht der Benutzerverwaltung

🔥 Registrierung abschließen 🔥

📌 Du hast einen gültigen Registrierungslink geöffnet. Gib deine Daten ein und drücke auf "Registrierung abschließen", um Zugang zur gAmification Plattform zu erhalten.

Username **E-Mail**

Dieses Feld darf nicht leer sein

Vorname **Nachname**

Passwort **Passwort wiederholen**

Mind. 8 Zeichen, 1 Groß-, 1 Kleinbuchstabe & 1 Zahl

Jahrgang **Studiengang** **Rolle**

Registrierung abschließen ✓

Abbildung 2: Benutzerregistrierung über einen neu generierten Link

Sprint 4 und 5 widmeten sich der Inhaltsverwaltung. Ziel war es, die Möglichkeit zu schaffen, Kurse, Challenges und Aufgaben (Tasks) zu erstellen, zu bearbeiten und zu löschen. Diese Elemente bilden die zentralen Lerninhalte, mit denen Lernende später interagieren. Ein besonderer Fokus lag dabei auf der Wiederverwendbarkeit und der freien Verknüpfbarkeit der einzelnen Lernelemente, wobei auch eine Änderungshistorie integriert wurde.

Zusätzlich konnten kleinere, von der Auftraggeberin gewünschte „Nice-to-Have“-Funktionen erfolgreich umgesetzt werden, darunter eine Drag-and-Drop-Sortierung, das Hochladen individueller Kursbilder und die automatische Gewichtung von Tasks und Challenges.

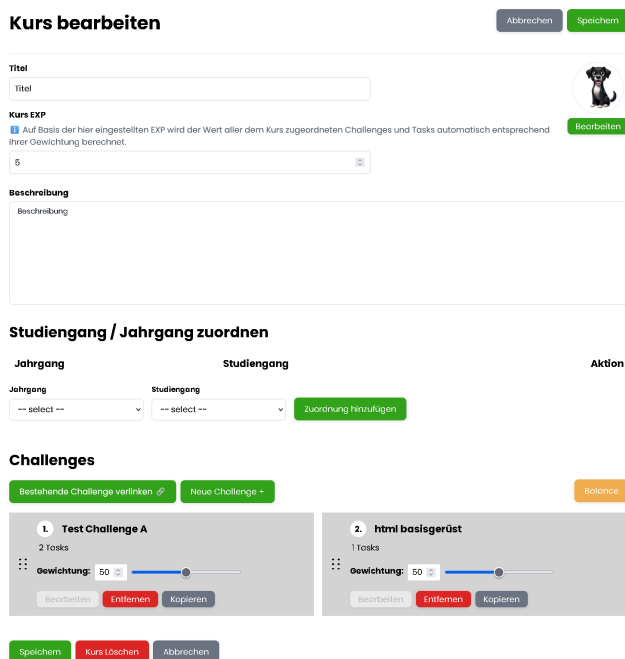


Abbildung 3: Bearbeitungsansicht eines Kurses

Nach Abschluss des letzten Sprints wurde die fertiggestellte Lektor*innenansicht an die Auftraggeberin übergeben, gemeinsam mit ihr in den Produktivzweig des gAmlcode Repositories integriert und in einer Testumgebung veröffentlicht.

Fazit

Die Umsetzung des Projekts war von Höhen und Tiefen geprägt. Insbesondere der Einstieg in die Technologien MeteorJS und MongoDB stellte für alle Projektmitglieder eine Herausforderung dar, da diese Technologien für das gesamte Team neu waren und eine intensive Einarbeitung erforderten.

Zu Beginn bestanden durchaus Zweifel an der sehr detaillierten Planungsphase. Im Verlauf des Projekts zeigte sich jedoch, dass diese intensive Vorbereitung äußerst wertvoll war. Da viele zentrale Aspekte bereits im Vorfeld umfassend definiert wurden, konnte sich das Team während der Umsetzung gezielt auf die Entwicklung und Implementierung der geplanten Features konzentrieren.

Ein wichtiger Motivationsfaktor waren die regelmäßig durchgeführten Sprint-Reviews. In diesen erhielten wir stets konstruktives und überwiegend positives Feedback von unseren Betreuer*innen, was uns ermöglichte, mit neuem Schwung in die jeweils nächsten Projektphasen zu starten.

Das Projekt war für das gesamte Team äußerst lehrreich. Neben der intensiven Arbeit mit bisher unbekanntem Technologien lernten wir insbesondere auch, effizient und zielgerichtet im Team zu arbeiten, klare Kommunikationswege zu etablieren und den Projektumfang flexibel an gegebene Ressourcen und Prioritäten anzupassen. Diese Fähigkeiten werden uns auch in zukünftigen Projekten zugutekommen.

Ein besonderer Dank gilt unserem Projektbetreuer Dipl.-Ing. Franz Knipp, der uns stets mit wertvollem technischen Feedback unterstützte und uns zudem eine eigene Testinstanz zur Verfügung stellte. Ebenso bedanken wir uns herzlich bei unserer Auftraggeberin für ihre stetige Unterstützung und freuen uns darauf, die Weiterentwicklung der gAmlcode-Plattform in zukünftigen Projekten aktiv mitgestalten zu dürfen.

Autor*innen: Nina Hohl, Tobias Pavel, Leon Rasner

Elektroauto Lade-App



Projektauftraggeber*in: embyt GmbH
Projektbetreuer*in: Dipl.-Ing. Franz Knipp
Projektmitglieder: Andreas Dötzl, Daniel Rimpfl, Marcel Turobin-Ort
Repository: <https://github.com/embyt/enline-app>

*Über das Backend der Firma embyt werden zahlreiche Ladestationen betrieben. Für Endkunden*innen gibt es bereits ein Web-Interface. Besitzer*innen von Elektroautos können darüber ihre Ladevorgänge einsehen. Dieses System soll durch eine Smartphone App für Android und iOS ergänzt werden, um eine verbesserte Benutzerschnittstelle und weitere Funktionen zur Verfügung zu stellen. Die Erweiterungen beziehen sich unter anderem auf Darstellung freier Ladestationen auf Karten und Listen.*

Auftraggeber

Die Firma *embyt* ist ein Wiener Unternehmen, das sich auf Energie- und Spitzenlastmanagement spezialisiert hat. Gegründet von Martin Daublebsky und Roman Morawek, liegt die Kernkompetenz in der Entwicklung von Software und Algorithmen für eine intelligente Energiesteuerung sowie der lastabhängigen Regelung elektrischer Verbraucher*innen. Technisch fokussiert *embyt* auf die Glättung von Verbrauchslastgängen zur Entlastung der Stromnetze und die Optimierung der Stromverteilung durch *smarte Algorithmen*. Damit leistet das Unternehmen einen aktiven Beitrag zur Energiewende, entlastet die Netze und fördert den Umweltschutz durch effiziente Energienutzung. Mit

mehreren hundert Anlagen in Betrieb und über tausend gesteuerten Ladestationen für Elektroautos ist *embyt* eine bewährte Größe im österreichischen Markt.

Projektbeschreibung

Das Backend der Firma *embyt* betreibt zahlreiche Ladestationen und bietet ein Web-Interface für Endkunden*innen an. Elektroautobesitzer können darüber ihre Ladevorgänge einsehen. Dieses System wird nun durch eine Smartphone-App für Android und iOS erweitert, die eine verbesserte Benutzeroberfläche und zusätzliche Funktionen bietet. Zu den Erweiterungen gehören die Anzeige freier Ladestationen auf Karten und in Listen sowie ein

Verlauf der Ladehistorie. Außerdem ermöglicht die App eine Echtzeit-Darstellung des Ladevorgangs mit Animationen und eine übersichtliche Darstellung des Ladeverlaufs.

und dem bestehenden App-Design. Die Authentifizierung und AWS-Interaktionen führten ebenfalls zu ungewollten Resultaten.

Projektverlauf

Im Verlauf des Projekts wurden insgesamt fünf Sprints durchgeführt, die jeweils unterschiedliche Herausforderungen und Erfolge mit sich brachten.

In der ersten Phase lag der Fokus ausschließlich auf technischen Aufgaben. Es wurden grundlegende Strukturen geschaffen, wie das Aufsetzen des GitHub-Repositories sowie der Aufbau einer Expo-Building-Pipeline inklusive EAS-Update. Zudem wurde eine erste Expo-App erstellt und ein Design-Research zu bestehenden Lade-Apps inklusive Wireframing durchgeführt. Ein technisches Problem trat jedoch auf, das sich über mehrere Phasen erstreckte, bis es gelöst werden konnte: Die Verbindung zum Backend über den GraphQL-Connector konnte nicht fertiggestellt werden, was die geplante Arbeit für die nächsten Schritte deutlich erschwerte.

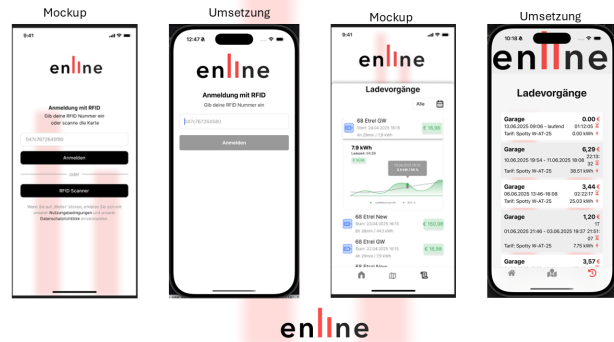


Abbildung 5: Phase 2

Die dritte Phase stellte einen Wendepunkt dar. Die bestehenden Herausforderungen konnten gelöst werden und trotz hoher Anforderungen wurden bedeutende Meilensteine erreicht. Das Projekt wurde komplett neu aufgesetzt und neu geschrieben, wodurch technische Schulden ausgeglichen werden konnten. Meilensteine, die in der zweiten Phase unerreichbar schienen, wurden nun Teil der App – wie zum Beispiel eine Live-Websocket-Verbindung zur Anzeige des Ladestands sowie die Kartenintegration.

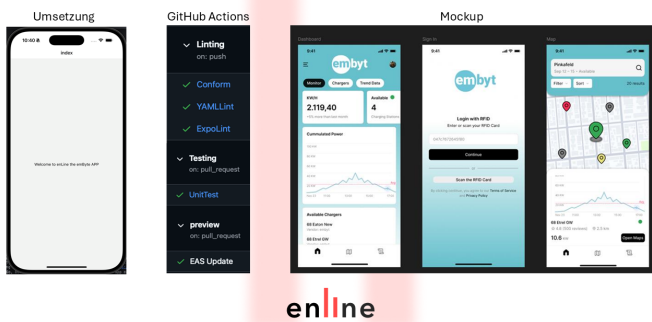


Abbildung 4: Phase 1

Die zweite Phase war von Schwierigkeiten geprägt und führte zum tiefsten Punkt des gesamten Projekts. Die Moral und Motivation des Teams hing am seidenen Faden. Neben einem fehlerhaften Setup von Xray, das zu Managementproblemen führte, gab es Unstimmigkeiten zwischen Auftraggeber



Abbildung 6: Phase 3

Die letzte Phase umfasste hauptsächlich Fehlerbehebungen und Design-Verbesserungen. Einige Punkte blieben jedoch offen, insbesondere die

Behandlung der Token-Expiration und die Implementierung des RFID-Readers für den Login. Zusätzlich wurde auch Fokus auf Push Notifications gelegt, da dieses Feature für den Auftraggeber deutlich von Interesse war. Nach Abschluss der Sprints wurden noch wichtige Verbesserungen vorgenommen, darunter der erwähnte Token-Refresh-Mechanismus und Optimierungen der Kartenfunktion, um die Nutzererfahrung vor der Projektübergabe weiter zu verbessern.

Plattformen keineswegs einfach ist und die Sicherstellung der Kompatibilität auf einer Vielzahl von Geräten eine große Herausforderung darstellt. Expo erwies sich als hilfreiches Werkzeug, das viele Prozesse erleichtert, auch wenn der Einstieg anfangs oft holprig war. Besonders wichtig war eine gut strukturierte und umfassende Dokumentation, die maßgeblich dazu beitrug, die App letztlich effizient und stabil umzusetzen.

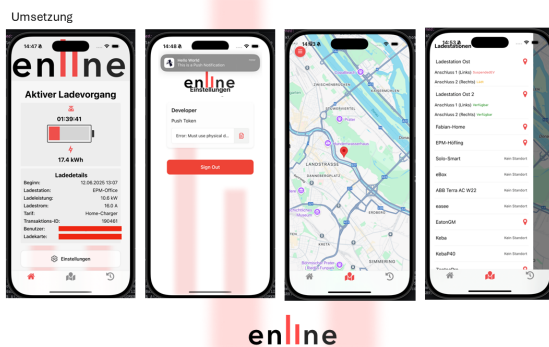


Abbildung 7: Phase 4

Ein Projekt im Team zu entwickeln erfordert eine andere Herangehensweise als die Arbeit alleine. Klare Kommunikation, gute Koordination und regelmäßige Absprachen sind entscheidend, um ein gemeinsames Verständnis zu schaffen und effizient voranzukommen. Auch die Verteilung von Aufgaben, das Einhalten von Standards und der offene Umgang mit Problemen tragen wesentlich dazu bei, dass das Projekt erfolgreich abgeschlossen werden kann.

Diese Erfahrungen bilden eine solide Basis für zukünftige Projekte und ermöglichen es, Entwicklungsprozesse künftig noch gezielter und erfolgreicher zu gestalten.

Fazit

Das Projekt hat deutlich gezeigt, dass die Entwicklung von Apps für Mobil-Geräten wie Android oder IOS ihre ganz eigenen Herausforderungen mit sich bringt. Besonders die nativen Elemente und Einstellungen erwiesen sich als komplex und schwer testbar (Push Notifications, Location, etc), was den Entwicklungsprozess erschwerte. Die Integration von AWS-Auth in einer React-Native-Umgebung stellte sich als besonders anspruchsvoll heraus und erforderte viel Aufwand. Auch die Migration auf Expo Version 53 sowie die Konfiguration der EAS-Builds waren technisch herausfordernd, nicht zuletzt wegen der Warteschlangen im Free-Tier und der komplexen Einstellungen.

Trotz dieser Schwierigkeiten konnten wertvolle Erkenntnisse gewonnen werden. Es wurde klar, dass Development für mehrere Betriebssysteme und



Abbildung 8: Die Übergabe

Autor*innen: Andreas Dötzl, Daniel Rimpfl, Marcel Turobin-Ort

I NEED HELP



Projektauftraggeber*in: Maler Ivo GmbH
Projektbetreuer*in: Dipl.-Ing. Franz Knipp
Projektmitglieder: Marcel Wagner, Stefan Hahnl, Jan Jungwirth
Repository: https://github.com/hochschule-burgenland-i-need-help/I_need_help_frontend

Drei Studierende des Studiengangs Software Engineering und Vernetzte Systeme (BSWE) an der Hochschule Burgenland haben eine mobile Applikation mit dem Namen „I NEED HELP“ entwickelt, die in Notfällen schnell und unkompliziert Hilfe leistet. Die App zeigt, standortbasiert die wichtigsten Notrufnummern für das jeweilige EU-Land an, bietet eine Direktanruf-Funktion und ermöglicht über das GPS-Signal des Mobiltelefon die Navigation zur nächsten Einsatzstelle der jeweiligen Blaulichtorganisation. Die Applikation wurde im Rahmen des Praxisprojekts agil umgesetzt.

Auftraggeber

Die gute Idee stammt von einem Malerunternehmen (Maler Ivo GmbH). Während der Inhaber mit seiner Familie am Strand in Kroatien lag, spielte er gedanklich ein Szenario durch, in dem ihm bewusst wurde, dass viel zu viele Menschen – er selbst eingeschlossen – in einer Notfallsituation nicht die richtige Notrufnummer des jeweiligen Landes kennen. Es müsste eine einfache Möglichkeit geben, die Notfallkette in Gang zu setzen. Genau hier setzt die App „I NEED HELP“ an.

Projektbeschreibung

Nach der erstmaligen Installation erkennt die App automatisch den aktuellen Standort der Nutzer*in und zeigt die jeweils korrekten Notrufnummern für das Land an, in der sie sich gerade befindet. So wird sichergestellt, dass im Ernstfall keine Zeit mit der Suche nach der richtigen Notrufnummer verloren geht. Zusätzlich unterstützt die App dabei, die Rettungskette möglichst schnell in Gang zu setzen: Per Fingertipp kann direkt ein Anruf bei der zuständigen Organisation – etwa Rettung, Feuerwehr oder Polizei – ausgelöst werden. Darüber hinaus bietet die Anwendung eine Routing-Funktion, bei der mit Hilfe externer Navigationssoftware wie

Google Maps eine Wegbeschreibung zur nächstgelegenen Blaulichtorganisation bereitgestellt wird.

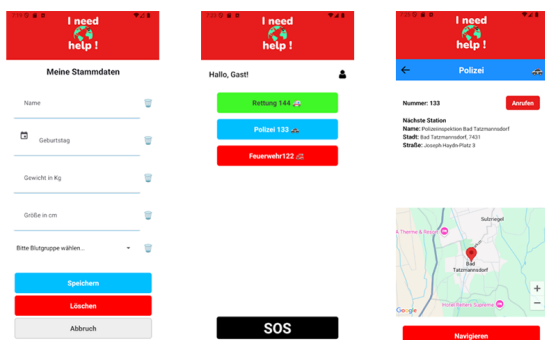


Abbildung 9: Die Applikation

Projektverlauf

Der Projektverlauf war von Anfang an facettenreich und lehrreich. Ein ursprünglich sehr umfangreiches Konzept mit zahlreichen Funktionen – von Notruf- und Pannendiensten bis hin zu Werbemodulen – wurde im Austausch mit dem Auftraggeber sinnvoll auf das Wesentliche reduziert. Der Fokus liegt nun auf der standortbasierten Anzeige relevanter Notrufnummern sowie der direkten Kontaktaufnahme und Navigation zur nächstgelegenen Einsatzorganisation. Diese Reduktion ermöglichte eine klare Zielsetzung und eine technisch nachhaltige Umsetzung.

Die Zusammenarbeit im Team verlief äußerst koordiniert und effizient. Die Aufgabenverteilung orientierte sich an einem ticketbasierten Workflow, der sämtliche Projektphasen – vom UI/UX-Design über die Entwicklung bis hin zur Qualitätssicherung – strukturiert abbildete. Regelmäßige interne Abstimmungen und kurze Retrospektiven trugen maßgeblich zur Qualität der Zusammenarbeit bei. Die teaminterne Organisation, insbesondere die Projektkoordination, erwies sich als äußerst hilfreich und trug wesentlich zum Projekterfolg bei.

Aus technischer Sicht stellte die Entwicklung der mobilen Anwendung für Android und iOS mit React Native (Expo) eine neue Herausforderung dar,

da das Team bis dahin vorrangig Erfahrung in Web- und Backend-Entwicklung gesammelt hatte.

Besonders einprägsam war der Moment, in dem der erste funktionsfähige Prototyp erfolgreich getestet werden konnte. Dieses Erfolgserlebnis wirkte stark motivierend und bestätigte das Team in seiner Arbeit.

Fazit

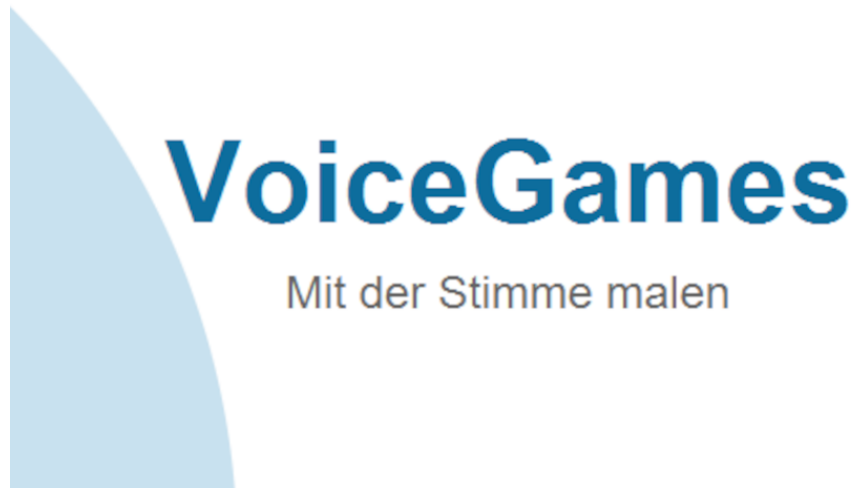
Das Praxisprojekt hat uns nicht nur fachlich gefordert, sondern auch in vielerlei Hinsicht persönlich wachsen lassen. Besonders wertvoll war die Erfahrung, ein reales Projekt eigenverantwortlich umzusetzen – von der ersten Idee bis zur funktionsfähigen App. Wir konnten unser theoretisches Wissen aus dem Studium in der Praxis anwenden, aber auch mit Herausforderungen umgehen, die man im Hörsaal nicht simulieren kann: Kommunikation mit dem Auftraggeber, Priorisierung von Anforderungen, technische Hürden und das Arbeiten im Team unter Zeitdruck. Gleichzeitig wurde uns bewusst, wie wichtig einfache, zugängliche digitale Lösungen in sicherheitsrelevanten Situationen sein können. Abschließend bleibt die Erkenntnis, dass selbst kleine Ideen – wie eine Notrufnummern-App – großes Potenzial haben, wenn sie mit Engagement und einem klaren Ziel umgesetzt werden.



Abbildung 10: Das Team

Autor*innen: Marcel Wagner, Stefan Hahnl, Jan Jungwirth

VoiceArt



Projektauftraggeber*in: Hochschule Burgenland, Bachelorstudiengang Logopädie
Projektbetreuer*in: Dipl.-Ing. Franz Knipp
Projektmitglieder: Gloria Brenner, Dario-Benjamin Dzacic, Michael Heckenast
Repository: <https://github.com/fhburgenland-bswe/ss2025-VoiceArt>

*„VoiceArt“ – im späteren Projektverlauf umbenannt zu „VoiceGames“ – ist eine Applikation zum Trainieren der Stimme. Im Fokus lag dabei immer der motivierende und spielerische Charakter beim Üben der Stimme. Dazu wurden vier Spiele entwickelt, welche mittels Tonhöhe und Lautstärke gesteuert werden. Anpassungsmöglichkeiten bei der Profilerstellung gewährleiten den Anwender*innen ein angenehmes Spielerlebnis. Außerdem ermöglicht die implementierte Galerie einen Rückblick auf den Trainingsverlauf und macht etwaige Verbesserungen im Umgang mit der Stimme sichtbar. VoiceGames wird in Zukunft als Lehrmittel im Studiengang Logopädie der HAW Burgenland zum Einsatz kommen.*

Auftraggeber

Der Studiengang Logopädie der Hochschule Burgenland hat zum Ziel, seine Studentinnen und Studenten zu Logopädinnen und Logopäden auszubilden. Diese helfen Patient*innen u.a. bei Sprach-, Stimm- und Schluckstörungen.

Nicole Feichtgraber, die Studiengangsleiterin, übernahm dabei die Rolle der Auftraggeberin. Frau Feichtgraber ist auch praktizierende Logopädin und erfahren im Umgang mit Stimmdiagnostik-Tools und konnte uns so auch wertvollen Input zu unserem Projekt liefern, wenn auch der Fokus nicht auf der Diagnostik lag.

Projektbeschreibung

Das Projekt zielt darauf ab, fernab der Diagnostik ein Programm zu entwickeln, mit der Benutzer*innen ihre Stimme spielerisch trainieren können. Zudem war auch wichtig, dass die Stimmeingabe mit handelsüblichen Mikrofonen, wie sie auch in Webcams eingebaut sind, funktioniert, denn nur so ist gewährleistet, dass man die App auch zuhause verwenden kann, ohne speziellen Schallpegelmesser, welcher in der logopädischen Praxis zu finden ist. Die App bietet neben der Auswahl von männlichem oder weiblichem Stimmprofil auch die Möglichkeit, ein benutzerdefinier-

tes Profil zu erstellen, wo Benutzer*innen den angezeigten Lautstärke- und Frequenzbereich selbst definieren können. Herzstück der Anwendung sind die vier Spielmodi.

FreeDraw gewährt Benutzer*innen freie Hand Stimme beim Erstellen von Kunstwerken. Mit der Lautstärke bewegt man den Stift entlang der y-Achse, mit der Frequenz entlang der x-Achse. Farbauswahl erfolgt über ein Dropdown. Erstellte Kunstwerke können per Knopfdruck als Bilddatei exportiert werden.

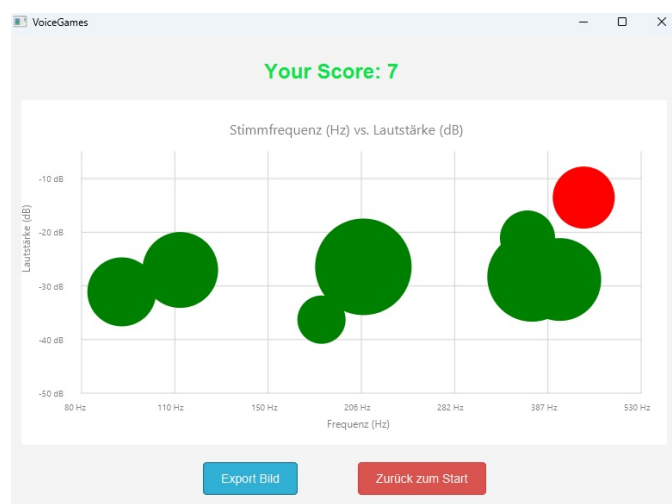


Abbildung 11: Ergebnis-Screen von *Hit The Points!*

Bei *Hit The Points!* haben die Spieler*innen die Aufgabe, mit ihrer Stimme vorgegebene Punkte „einzufangen“. Nach jedem eingefangenen Punkt wird der nächst-generierte kleiner, sodass es immer schwieriger wird, in die richtige Stimm-Range zu gelangen.

Mit *VoiceZone* haben Spielende die Möglichkeit, isoliert Lautstärke oder Frequenz zu trainieren. Auch hier werden sie mit einer gewissen Range gepromptet, jedoch muss ihre Stimme sich einige Sekunden lang im Zielfeld befinden, bevor ein nächstes erscheint.

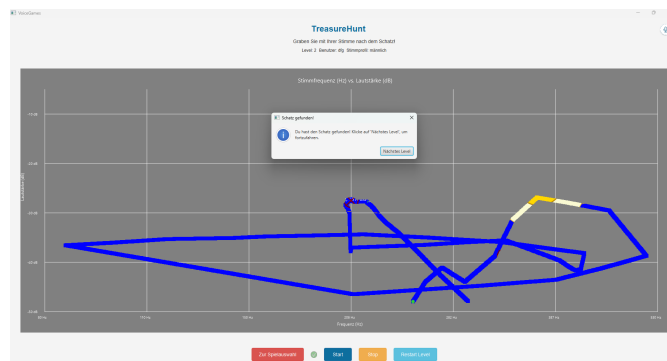


Abbildung 12: *TreasureHunt*

In *TreasureHunt* suchen Spieler*innen nach einem verborgenen Schatz, den sie mit ihrer Stimme Stück für Stück freigraben. Mit der Stimme legt man langsam eine Schatzkarte frei, die indiziert, wo sich der Schatz befindet. Hat man mit seiner Stimme den Schatz getroffen, wird ein neuer – jetzt kleinerer – generiert, und die Schatzsuche wird anspruchsvoller.

Hinzu bietet die App noch Features zur Mikrofon-Auswahl und das Loggen von Stimmdateien der einzelnen Spiel-Sessions.

Projektverlauf

Das Projekt durchlief fünf Sprints, wobei in jedem Sprint ein anderer Fokus gelegt wurde. Zuvor noch hielten wir ein ausführliches Anforderungsgespräch mit unserer Auftraggeberin, in dem geklärt wurde, wie das fertige Produkt aussehen soll. Anhand dieser Requirements planten wir unsere Sprints. Im Sprint 1 beschäftigten wir uns vorwiegend mit dem Erlernen der für das Projekt notwendigen Technologien. JavaFX und das Arbeiten mit Audio-Input war für jedes Gruppenmitglied neu, also planten wir Spikes ein, um mit solider Wissensbasis in die folgenden Sprints zu gehen. Wichtig war uns hier, zu evaluieren, welche Fast Fourier Transformation (FFT)-Library wir für unser Projekt einsetzen möchten. Also baute jeder von uns einen kleinen Prototypen zur Audio-Visualisierung, jedoch jeweils mit einer anderen FFT-Library. Außer-

dem wurde im ersten Sprint eine GitHub-Pipeline eingerichtet, um das weitere Entwickeln und Testen angenehmer zu gestalten.

Im Sprint 2 lag der Fokus auf Grundfunktionalitäten, die für die spätere Entwicklung der Spiele vonnöten waren: Die Gestaltung des Koordinatensystems, Einbinden des Mikrofons (inkl. Mikrofonauswahl), Audioverarbeitung mittels FFT und das Zeichnen der Werte ins Koordinatensystem. Hier wurde uns zum ersten Mal klar, dass unser ursprüngliche Plan – dem User mit seiner Stimme geometrische Figuren nachzeichnen lassen – zwar machbar ist, jedoch keinesfalls motivierend erscheinen wird. Unsere Selbstversuche ergaben, dass es viel zu schwierig ist, als Lai*in präzise mit der Stimme zu zeichnen. Diese Bedenken teilten wir unserer Auftraggeberin und unserem Projektbetreuer mit, woraufhin wir den Scope so veränderten, dass nun Spiele zu entwickeln waren, die den motivierenden Charakter der App wahren sollten.

Glücklicherweise hatten wir Sprint 3 schon mit Profilerstellungs- und Profilverwaltungs-Funktionen geplant, sodass genügend Zeit war, uns Spiele für Sprint 4 auszudenken. Plötzlich jedoch funktionierte unser Programm nicht mehr. Es brauchte einige Zeit, bis wir herausfanden warum: das Repository unserer FFT-Library, die wir mit Gradle einbanden, ging einfach offline. Das Problem entziffert, konnten wir schnell einen Workaround basteln und ungehindert weiterarbeiten.

In Sprint 4 wurden die oben beschriebenen Spiele entwickelt. Besonders in dieser Phase zeigte sich, wie wichtig unsere Testfälle waren. Wir hatten Sprint 4 so eingeteilt, dass jeder von uns ein Spiel entwickelt und das Spiel eines anderen Gruppenmitglieds testet. Egal, wie sauber die Entwickler*in glaubte gewesen zu sein, die Tester*in fand immer einen kleinen Bug.

Mit übertrainierter Stimme starteten wir in Sprint 5. Der Fokus lag auf Usability-Features und dem Finalisieren der Projektdokumentation. Zu den Featu-

res gehörten Custom-Profile, Farbauswahl im FreeDraw, das Speichern von Bildern sowie das Loggen von Session-Daten. Im Sprint-Review 4 wurde uns zudem empfohlen, die lineare Skala im Koordinatensystem zu einer logarithmischen umzubauen, da diese besser zur menschlichen Stimme passt. Nach anfänglichen Schwierigkeiten, da die Änderung jedes Spiel betraf, konnten wir auch dies umsetzen. Bei der Projektübergabe durften wir, nach einer Installationsanleitung, ein letztes Mal unsere meisterlichen Stimmen präsentieren.

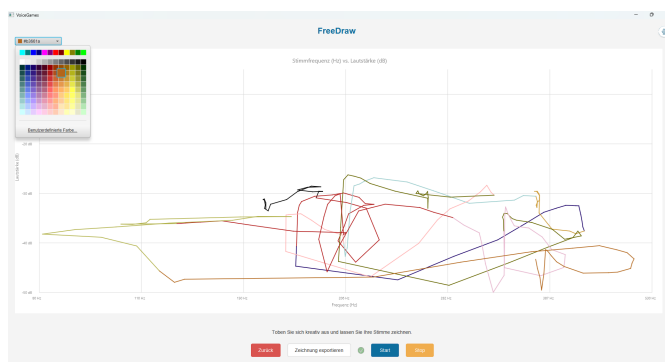


Abbildung 13: FreeDraw mit Farbauswahl

Fazit

Das Projekt stellte uns vor einigen Herausforderungen, zeigte uns aber auch, dass mit guter Recherche Dinge umsetzbar sind, von denen man sich zuvor noch nie beschäftigt hat. Außerdem erkannten wir durch die vermehrten Anforderungsänderungen, wie wichtig eine agile Arbeitsweise in der Softwareentwicklung ist. Dank einer wunderbaren internen Kommunikation und dem wertvollem Austausch mit Projektbetreuer und Auftraggeberin konnten wir auf auftretende Problemstellungen schnell reagieren. Wir verlassen das Projekt nicht nur mit gestählter Stimme, sondern auch mit vielen dazugewonnenen Entwickler*innen-Fähigkeiten und ein bisschen Stolz in Hinblick auf das, was wir in so begrenzter Zeit geschaffen haben.

Autor: Michael Heckenast

Atlas der Klima-bezogenen-Gesundheitsrisiken

big5health

Projektauftraggeber*in: big5health
Projektbetreuer*in: Dipl.-Ing. Wolfram Rinke
Projektmitglieder: Stephan Auer, Manuel Neuhold, Ábel Horváth-Szarka, Richard Gellen
Repository: <https://github.com/fhburgenland-bswe/ss2025-klimaatlas>

*Der Atlas der Klima-bezogenen-Gesundheitsrisiken ist mit dem Hintergedanken Hautgesundheit konzipiert worden. In dem initialen Meeting wurde, sowohl vom Team als auch vom Auftraggeber, schnell ein Design im Sinne von Google angestrebt, da mögliche Nutzer bereits damit vertraut sind und ein gewohntes Gefühl der Nutzer vorhanden ist. Im Verlaufe des Projektes wurde schnell klar, dass das Team im Rahmen des Projektes die Grundlage schafft und das Projekt so designet, dass andere Programmierer*innen das Projekt leicht mit zusätzlichen Datensätzen erweitern können. Im Laufe des Projektes wurde klar, dass nicht alle initial gewünschten Datensätze eingepflegt werden können und somit ein provisorisches Content Management System (CMS), um die gewünschten Tipps und Risiken zumindest darstellen zu können.*

Auftraggeber

big5health ist ein Verein zur Vermittlung von Wissen und zur Steigerung von Gesundheitskompetenz und Lebensfreude. Im Mittelpunkt der Aktivitäten des Vereins stehen die „Big 5“ der chronischen Erkrankungen: Adipositas und Diabetes, Herz-Kreislauf-Erkrankungen, Asthma und chronisch-obstruktive Lungenerkrankungen (COPD), Schmerzerkrankungen sowie psychische Erkrankungen. Der Rahmen seiner Arbeit besteht unter anderem darin, Fortbildungen und Schulungen für diverse Gruppen, darunter Schulen und

medizinische Professionisten durchzuführen. Der diesjährige Themenschwerpunkt Hautgesundheit bietet eine Möglichkeit, eine interaktive Informationsplattform für diverse Gesundheitstipps und -risiken zu erstellen. Mithilfe von Datensätzen von Partnern, darunter Geosphere Austria (vormals Zentralanstalt für Meteorologie und Geodynamik (ZAMG)) und AGES, soll eine Plattform geschaffen werden, die Mithilfe von Daten wie z.B. Temperaturen, Luftfeuchtigkeit, Sonneneinstrahlung und Moskitos möglichst spezifische Tipps und Risiken erfassen und basierend darauf einen Infotext darstellen kann.

Projektbeschreibung

big5health hat das Team beauftragt, eine Plattform begrenzt auf Österreich zu erstellen, die diverse Funktionen beinhalten soll. Die Kernfeatures der Plattform sollen um ein Google Maps-ähnliche Applikation gebaut werden, damit neue Nutzer sich darauf fokussieren können die Plattform zu verwenden, anstatt ein neues UI lernen zu müssen.

Um dies umzusetzen, will big5health mit Partnern zusammenarbeiten um Datensätze für Österreich zu bieten, die das Team folgend in die Applikation integrieren kann. Es wird geplant, die Applikation in einer eigenen Unterseite der bereits bestehenden big5health.at Webseite leben zu lassen, jedoch war zu Beginn des Projektes noch unklar, ob das Team oder der Verwalter der jetzigen Website die Einbindung übernehmen wird. Die bereits existierende Website besteht auf Wordpress und wird bei World4You gehostet, somit konnten wir uns als Team darauf fokussieren, eine funktionale Applikation zu erstellen, da wir mit den im Studium gelernten Technologien arbeiten können.

Dabei wurde im Backend ein besonderer Fokus auf die Datensätze von Geosphere Austria und der Moskito Datensatz der AGES gelegt. Da diese Datensätze eine gute Repräsentation bieten, wie man Datensätze einpflegt, können künftige Datensätze leichter und schneller mit aufgenommen werden. Im Frontend haben wir zu Beginn einen Prototypen gebaut, der Elemente von Google Maps übernommen hat und diesen mit UI-Elementen gebaut.

Projektverlauf

Das Team hat sich zu Beginn Rollen zugeteilt um Arbeit mit möglichst hoher Qualität in den jeweiligen Teilen präsentieren zu können und Spezialisierungen der Mitglieder ausnutzen zu können. Darunter fiel die Kommunikation mit dem Auftraggeber und somit der Teamlead auf Stephan Auer. Daraus folgten zwei Duos, die in Zusammenarbeit jeweils Frontend und Backend übrig hatten. Stephan

Auer und Manuel Neuhold fokussierten sich auf das Backend, Richard Gellen und Abel Horváth-Szarka auf das Frontend. Dies lief zu Beginn des Projektes ganz gut, da die Teams jeweils ihre Expertisen am Besten nutzen konnten. Im Verlaufe des Projektes verschwammen die Grenzen jedoch und jeder half den anderen dort, wo er konnte.

Im Backend haben wir angefangen eine Applikation zu bauen, die mehrere Möglichkeiten hat, Datensätze zu verwalten. Zu Beginn hatten wir geplant, eine Datenbank zu verwenden, um die diversen Datensätze zu verwalten, kamen dann aber schnell zum Entschluss, dass es – wenn überhaupt – eine In-Memory Datenbank sein müsste. Somit haben wir uns dazu entschieden, dass wir Caching verwenden, um Datensätze In-Memory auf dem Backend-Server zu speichern. Das Hauptproblem bei der Implementierung war ein sehr strenges Ratelimit, speziell auf dem Geosphere API SPARTACUS v2.1 Tagesdaten Endpunkt. Durch das niedrige Limit von maximal 120 Anfragen pro Stunde mussten wir also eine Lösung finden, wie wir die Anfragen auf den Endpunkt minimieren, aber auch in Bedacht auf zukünftige Endpunkte. Zu Beginn hatten wir also „nur“ geplant, ein Caching-System einzurichten, welches jede Abfrage die vom Frontend kommt, in einen Speicher schreibt, sodass jede weitere Anfrage auf den selben Punkt nicht mehr auf den SPARTACUS-Endpunkt geht.

Die Lösung, auf die wir uns dann geeinigt haben, war es mit Repräsentationspunkten zu arbeiten, welche beim Start in den Cache geladen werden und in einem File verwaltet werden können. In diesem File stehen Koordinaten für Populationszentren in Österreich, darunter die Landeshauptstädte und andere größere Städte, aus denen der Großteil der Anfragen kommt und somit keine zusätzlichen Anfragen auf den SPARTACUS Endpunkt mehr verursachen.

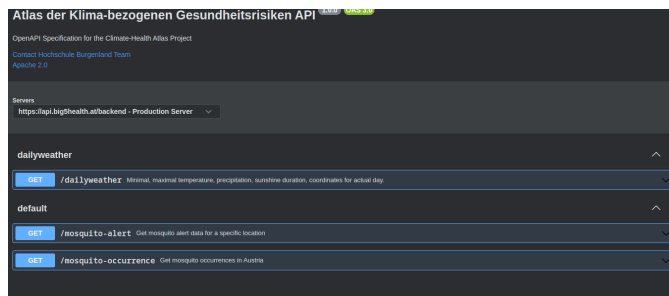


Abbildung 14: API Endpunkte des Caching Backend Servers

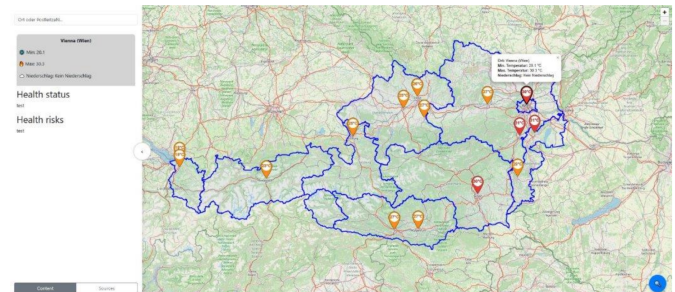


Abbildung 15: Finales UI der Applikation

Im Frontend hatten wir weniger Probleme mit der Umsetzung, jedoch kamen im Verlauf des Projektes einige Kundenwünsche hinzu. Zu Beginn waren wir hier voll und ganz darauf fokussiert, ein UI zu bauen, mit dem potenzielle Nutzer schon vertraut sind. Somit entschieden wir uns, uns am Google Maps UI zu orientieren, und begannen, dies mit den Funktionen unseres Backend Servers nachzubauen. Wir planten die Verwendung von mehreren Ebenen, die mit den einzelnen Datensätzen verknüpft sind. Diese Ebenen werden mit Pins auf eine Österreich-Karte dargestellt. Mittels eines Dropdown-Menüs kann die angezeigte Ebene gewechselt werden.

In den einzelnen Ebenen werden Datenpunkte aus dem Backend mittels Pins auf der Österreich-Karte dargestellt. Beim Geosphere-Endpunkt werden die Temperaturen der jeweiligen Orte mittels Farben dargestellt. Beim Moskito-Endpunkt werden Vorkommen von besonderen Moskitoarten auf der Karte dargestellt.

Im Verlauf des Projektes wurde klar, dass man mit den Datensätzen, die wir bis zum Ende des Projektes einpflegen können, keine genaue Aussagen bezüglich der Tipps und Risiken machen kann. Aus diesem Grund haben wir dann übergangsweise ein CMS erstellt, um manuell Tipps und Risiken eingeben und darstellen zu können.

Fazit

Abschließend lässt sich sagen, dass wir durch dieses Projekt gelernt haben, spezielle Kundenwünsche umzusetzen, ohne im Hintergrund die Architektur großartig anpassen zu müssen. Gleichzeitig errichteten wir im Backend einen Server, der mehrere Möglichkeiten anbietet, zukünftige Datensätze zu implementieren, ohne neue Technologien in die Struktur zu integrieren.

Final möchte ich mich bei meinen Kollegen für die gute Mitarbeit bedanken. Ohne deren Geduld wäre es wohl nicht so schön über die Bühne gelaufen, da wir doch einige Änderungen im Frontend durchführen mussten und im Backend Probleme bei der Implementation auftraten.

Autor: Stephan Auer

NEXUS



Projektauftraggeber*in: Forschung Burgenland GmbH
Projektbetreuer*in: Dipl.-Ing. Wolfram Rinke
Projektmitglieder: Georg Trnka, Miklos Komlosy, Rene Souri
Repository: <https://github.com/Mikomy/ppr-ss2025-group-f-frontend>

Das Projekt NEXUS macht Sensordaten aus dem Gewächshauslabor der Forschung Burgenland sichtbar. Entwickelt wurde eine Webanwendung, die Umweltfaktoren wie Temperatur oder Bodenfeuchte verständlich aufbereitet und über die Zeit analysierbar macht. Damit schafft NEXUS die Grundlage für datenbasiertes Beobachten und Vergleichen von Pflanzenbedingungen.

Auftraggeber

Die Forschung Burgenland steht für anwendungsorientierte und international ausgerichtete Forschung mit starkem regionalem Bezug. Als zentrale Schnittstelle für Forschungs- und Entwicklungsprojekte arbeitet sie erfolgreich mit öffentlichen Einrichtungen, Non-Profit-Organisationen und namhaften Unternehmen der Region zusammen. Als Tochtergesellschaft der Hochschule für angewandte Wissenschaften Burgenland ist sie an den beiden Hochschul-Standorten Eisenstadt und Pinkafeld tätig.

DI Dr. Lukas Gnam, Hochschullehrender am Department Energie & Umwelt, der das Projekt initiierte, begleitete das Projektteam während der Umsetzung in beratender Funktion.

Projektbeschreibung

Das Projekt NEXUS widmete sich der Entwicklung einer Webanwendung zur Darstellung und Auswertung von Umweltdaten, die im Rahmen der lokalen Wachstumskontrolle von Pflanzen im gleichnamigen Labor der Forschung Burgenland erfasst werden. Die dort eingesetzte Sensorik misst Parameter wie Temperatur, Bodenfeuchte und Luftfeuchtigkeit in verschiedenen Hochbeeten. Ziel des Projekts war es, eine technische Lösung zu schaffen, mit der diese Daten übersichtlich visualisiert und für künftige Analysen zugänglich gemacht werden können.

Da ein direkter Remote-Zugriff auf die Infrastruktur des Labors nicht vorgesehen war, arbeitete das Projektteam auf Basis eines bereitgestellten Datenexports aus der produktiven InfluxDB-Instanz. Die

Sensordaten wurden in eine lokal installierte InfluxDB-Version 1.x importiert und bildeten die zentrale Grundlage für die Entwicklung der Backend-Logik. Diese wurde mit Spring Boot realisiert und stellt die Messdaten über eine REST-API zur Verfügung. Das Backend ermöglicht neben dem Abruf spezifischer Messungen auch Gruppierungen nach Sensoren, Berechnungen aggregierter Statistiken sowie die gezielte Abfrage aktueller Datenpunkte innerhalb definierter Zeiträume. Ergänzt wurde die Schnittstelle um einen Alias-Mechanismus, der es erlaubt, komplexe Measurement-Namen in sprechende API-Endpunkte umzuwandeln.

Das Frontend wurde mit Angular umgesetzt und nutzt die bereitgestellten REST-Endpunkte, um Messwerte je nach Auswahl interaktiv darzustellen. Neben einer Dashboard-Ansicht mit aktuellen Statistiken, Minima, Maxima und einem Kurzüberblick (siehe Menüpunkt „Live-Überblick“) bietet die Anwendung auch eine Analyseansicht, in der mehrere Messreihen gleichzeitig konfiguriert, dargestellt und visuell verglichen werden können. Der Benutzer kann Messungen individuell kombinieren, Farben und Zeitfenster wählen sowie Ansichten persistent im lokalen Speicher ablegen (siehe Menüpunkt „Diagramm Analyse“).



Abbildung 17: Screenshot Page – Diagramm-Analyse

Ergänzt wird die Anwendung durch ein Statistikmodul, das Vergleichsanalysen zwischen Sensorgruppen ermöglicht. Hierbei werden unter anderem Mittelwert, Median sowie Interquartilsabstand berechnet und auffällige Ausreißer anhand der Tukey-IQR-Methode identifiziert. Die Ergebnisse werden in Form von Scatterplots visualisiert (siehe Menüpunkt „Statistik“). Zusätzlich steht eine Tabellenansicht zur Verfügung, die eine detaillierte, filterbare Darstellung einzelner Messwerte in paginierten Datentabellen erlaubt (siehe Menüpunkt „Tabelle nach Sensorauswahl“).

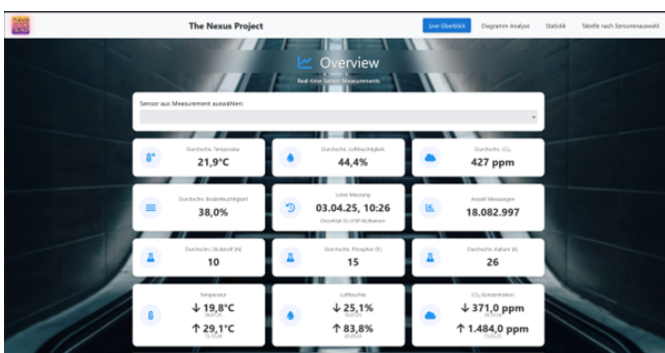


Abbildung 16: Screenshot Page – Live-Überblick

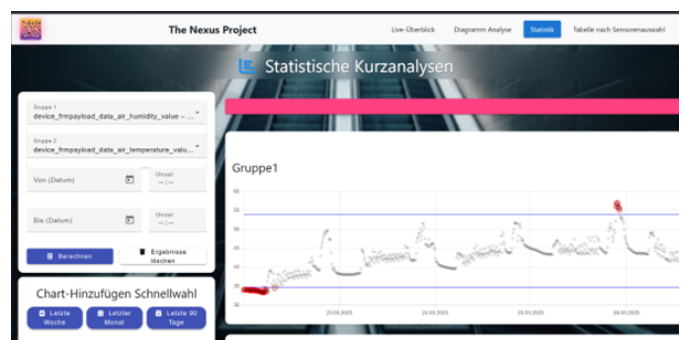


Abbildung 18: Screenshot Page – Statistik



Abbildung 19: Screenshot Page – Tabelle nach Sensorwahl

Die gesamte Anwendung wurde bewusst so konzipiert, dass sie vordergründig lokal betrieben werden kann. Eine Nutzung externer Cloud-Dienste zur Speicherung der Daten war im Rahmen des Projekts ausdrücklich nicht gewünscht. Um einen stabilen Betrieb zu gewährleisten, wurden gezielte Validierungen und Fehlerbehandlungen auf API-Ebene umgesetzt. So werden ungültige Zeitbereiche, fehlende Daten oder fehlerhafte Sensoranfragen strukturiert abgefangen und mit konsistenten HTTP-Fehlercodes beantwortet.

Projektverlauf

Der Projektverlauf gliederte sich in fünf aufeinander aufbauende Sprints. Zu Beginn stand die Sichtung und Aufbereitung der vorhandenen Sensordaten im Fokus. Dabei wurden die Struktur der InfluxDB, das Datenmodell sowie relevante Measurement- und Sensor-Kombinationen analysiert. Bereits früh wurde ein funktionaler Prototyp des Backends implementiert, der einfache Abfragen an die Datenbank erlaubte. Parallel dazu entstand ein erster Entwurf des Angular-Frontends, das einfache Zeitverläufe grafisch darstellte.

Im weiteren Verlauf wurden die Funktionalitäten sukzessive erweitert. So wurden im Backend einheitliche Filterparameter für Zeiträume und relevante Messungen eingeführt, während im Frontend interaktive Konfigurationsmöglichkeiten für

die Nutzer*innen geschaffen wurden. Der Fokus lag stets auf einer hohen Flexibilität und einer verständlichen, klar strukturierten Oberfläche. Besondere Aufmerksamkeit galt der Performance und der Robustheit der Abfragen, insbesondere bei der Darstellung umfangreicher Zeitverläufe und bei parallelen Zugriffen auf mehrere Messreihen.

Mit Fortschreiten des Projekts wurde die statistische Analyse erweitert und vollständig in die Benutzeroberfläche integriert. Dabei wurde Wert auf eine verständliche Interpretation der Ergebnisse gelegt – nicht nur durch die mathematische Darstellung, sondern durch die gezielte Visualisierung in Form von Balkendiagrammen, Scatterplots und tabellarischen Zusammenfassungen. Gegen Ende des Projekts wurde die Codebasis durch umfangreiche Tests sowie Qualitätsprüfungen abgesichert.

Fazit

Mit dem Projekt NEXUS konnte eine praxisnahe, funktional umfangreiche Anwendung umgesetzt werden, die Sensordaten aus einem realen Hochschullabor nutzerfreundlich zugänglich macht. Die Lösung zeichnet sich durch ihre modulare Architektur, die klare Trennung von Frontend und Backend sowie die gezielte Integration statistischer Verfahren aus. Die Einhaltung lokaler Datenschutzvorgaben und der bewusste Verzicht auf Cloud-Dienste unterstreichen den pragmatischen, datensensiblen Ansatz des Projekts. Die entstandene Anwendung stellt eine solide Grundlage für eine künftige direkte Anbindung an die InfluxDB des Labors NEXUS dar und lässt sich einfach erweitern. Für das Projektteam bedeutete die Arbeit an NEXUS nicht nur eine technische Herausforderung, sondern auch eine wertvolle Lernerfahrung in der Koordination, Dokumentation und praxisorientierten Umsetzung einer modernen datengetriebenen Webanwendung.

Autoren: Georg Trnka, Miklos Komlosy, Rene Souri

Pegelhub



Projektauftraggeber*in: Johannes Groisz, viadonau
 Projektbetreuer*in: Dipl.-Ing. Wolfram Rinke
 Projektmitglieder: Armin Grebić, Christian Schlögl, Dominique Haller, Efnan Evelioglu
 Repository: <https://github.com/viadonau/Pegelhub-Core.git>

Pegelhub ist ein Projekt der viadonau und wird laufend weiterentwickelt, sowohl von Hochschulen als auch von einer externen Firma. Aktuell befindet es sich bereits in der fünften Version. Im Rahmen unseres Praxisprojekts wurde die Funktionalität des IEC-Connectors implementiert und eine Cluster-Lösung für den Pegelhub-Core erarbeitet. Diese Maßnahmen waren entscheidend, um die Ausfallsicherheit und Effizienz der Pegeldatenverteilung sicherzustellen.

Auftraggeber

Die *viadonau*, die Österreichische Wasserstraßen-Gesellschaft-mbH, ist das nationale Kompetenzzentrum für die Wasserstraßen der Donau. Ihre Hauptaufgabe ist die Entwicklung und Bewirtschaftung des Donauraums als Lebens-, Natur- und Wirtschaftsraum. Dazu zählen Instandhaltung, Betrieb und Ausbau der Wasserstraßeninfrastruktur (Schleusen, Uferanlagen, Treppelwege), aber auch internationale Zusammenarbeit mit anderen Ländern, durch die die Donau fließt, sowie aktiver Schutz der Ökosysteme. Die Infrastruktur beinhaltet auch die Bereitstellung von Daten für die Schifffahrt und den Hochwasserschutz, wofür Pegelhub zukünftig eine wichtige Rolle spielen wird.

Projektbeschreibung

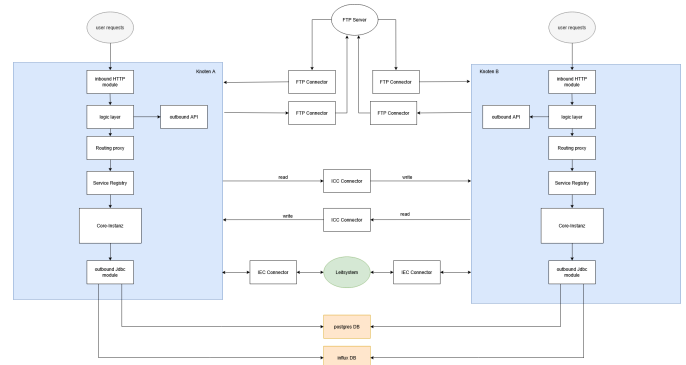


Abbildung 20: Architektur von Pegelhub

Pegelhub soll grundsätzlich als eine moderne Software für die Verteilung von Pegeldaten dienen und

in Zukunft veraltete Systeme ablösen. Es dient als Datendrehscheibe, bei der Pegelbetreiber ihre Pegel­daten mit klar definierten Metadaten bereitstellen können.

Die zentralen Komponenten von Pegelhub sind:

- Pegelhub Core
- FTP-Connector: Verbindung vom Core zu einem FTP Server
- ICC-Connector: synchronisiert Daten zwischen zwei Knoten
- IEC Connector: Verbindung vom Core zum Leitsystem
- InfluxDB: Messdaten
- PostgreSQL: Metadaten

Die Hauptaufgaben unseres Projektteams waren, den IEC-Connector lauffähig zu machen, sowie eine Cluster-Lösung zu implementieren, um den Core gegen Ausfälle abzusichern.

Der IEC-Connector dient dem Austausch von Telemetrie- und Messdaten zwischen dem Pegelhub-Core und einem IEC-Server. Dazu wird das IEC 60870-5-104 Protokoll verwendet.

Für die Umsetzung des Clusters haben wir uns für Docker Swarm entschieden. Docker Swarm ist eine native Orchestrierungslösung von Docker, mit der mehrere Server (Nodes) zu einem logischen Verbund zusammengefasst werden können.

Projektverlauf

Die erste Hürde in unserem Projekt war das Einlesen in die Materie sowie das Verstehen des vorhandenen Codes. Keiner von uns hatte Erfahrung mit dem Pegelwesen und den dazu notwendigen Connectoren. Dazu kam, dass der bestehende Code zwar teilweise mit Javadoc versehen und dokumentiert war, aber es dennoch sehr schwierig war, diverse Fachbegriffe zu verstehen, auf die man

in Bezug auf Pegel­daten immer wieder stößt. Zum Glück verfügte unser Auftraggeber über sehr umfangreiches Wissen zum bestehenden System und war jederzeit für uns erreichbar, um Fragen zu beantworten. Die ersten zwei Sprints haben wir also hauptsächlich damit verbracht, Dokumentationen zu lesen und den vorhandenen Code zu analysieren. Auch Skizzen des Systems sind in dieser Zeit entstanden, die im weiteren Verlauf sehr geholfen haben, die Funktion zu verstehen und die Auswirkung von Änderungen einzuschätzen.

Leider ist uns die Zeiteinteilung nicht besonders gut gelungen und so wurde es vor allem im letzten Sprint sehr stressig. Die Systemtests des IEC-Connectors konnten erst gegen Ende des Projekts durchgeführt werden, da es in Bezug auf den IEC-Testserver einige Verzögerungen gab. Ein Testsystem, das wir vom Auftraggeber erhalten hatten, erwies sich als zu umfangreich, um es so spät im Projekt noch einzurichten, also fiel unsere Wahl letztendlich auf das FreyrScada System. Dieses war leider ebenfalls kompliziert einzurichten und enthielt keine gute Dokumentation zur Bedienung, weshalb es sehr viel Zeit in Anspruch nahm, das System richtig zu konfigurieren und mit unserem IEC-Connector zu verbinden. Nach vielen Fehlversuchen gelang es uns, mit dem überarbeiteten IEC-Connector Daten zum IEC-Testserver zu lesen und auf den Testserver zu schreiben. Ein großer Erfolg für uns, der die Stimmung nach viel Frustration wieder verbesserte.

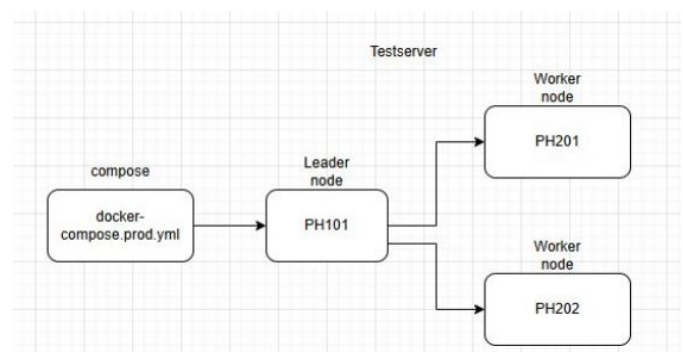


Abbildung 21: Aufbau des Clusters

Auch die Implementierung des Clusters nahm viel Zeit in Anspruch. Zu Beginn des Projekts mussten wir viel recherchieren und so fiel die Entscheidung zuerst auf Kubernetes, da wir mit Kubernetes bereits im Unterricht gearbeitet hatten und so ein wenig Erfahrung vorweisen konnten. Hier kam es aber zu einer kurzfristigen Änderung seitens des Auftraggebers zu Docker Swarm, da die Implementierung von Kubernetes bereits viel Zeit in Anspruch genommen hatte und die Befürchtung bestand, dass wir damit in der gegebenen Zeit nicht fertig werden. Also musste mit Docker Swarm noch einmal von vorne begonnen werden. Zum Glück war dessen Einrichtung aber relativ schnell erledigt und wir konnten zu Ende des Projekts die Cluster-Lösung auf drei Testserver der viadonau deployen.

Der Pegelhub-Cluster besteht aus drei Servern:

- ph101: Manager-Node bzw. Leader steuert und überwacht den Cluster
- ph201 und ph202: Worker-Nodes, führen die vom Manager zugewiesenen Services aus

<input type="checkbox"/>	Name	Container ID
<input type="checkbox"/>	lec-connector	cb8daf489ec5
<input checked="" type="checkbox"/>	pegelhub-core	-
<input type="checkbox"/>	proxy-1	2b6f8f8e2e6e
<input type="checkbox"/>	starter-1	b4cc35633cb2
<input type="checkbox"/>	meta-db-1	c7e936643eb0
<input type="checkbox"/>	data-db-1	366d67995b1c
<input type="checkbox"/>	registry-1	6b2ef58059e5

Abbildung 22: *Pegelhub in Docker*

Ebenfalls gelungen ist uns das Erstellen einer Pipeline für Github, die bei Pushes auf den Main-Branch neue Images erstellt und diese auf Github Containers (GHCR) veröffentlicht. Von dort können die Images gepullt werden, was das Starten der Komponenten wesentlich vereinfacht.

Fazit

Nach einigen Schwierigkeiten, vor allem im Bezug auf Projektmanagement und Testing, schafften wir es dennoch bis zum letzten Review, die wichtigsten Anforderungen zu implementieren. Bei der Übergabe des Projekts an unseren Auftraggeber konnten wir alle neuen Funktionen erfolgreich demonstrieren.

Im Bereich des Projektmanagements lernten wir besonders viel dazu. Unsere ursprüngliche Planung hatte sich schnell als unvorteilhaft erwiesen, da wir ursprünglich die Abarbeitung aller Ziele nacheinander festgelegt hatten. Letztendlich erwies es sich als wesentlich praktikabler, das Projektteam aufzuteilen und an unseren Zielen parallel zu arbeiten.

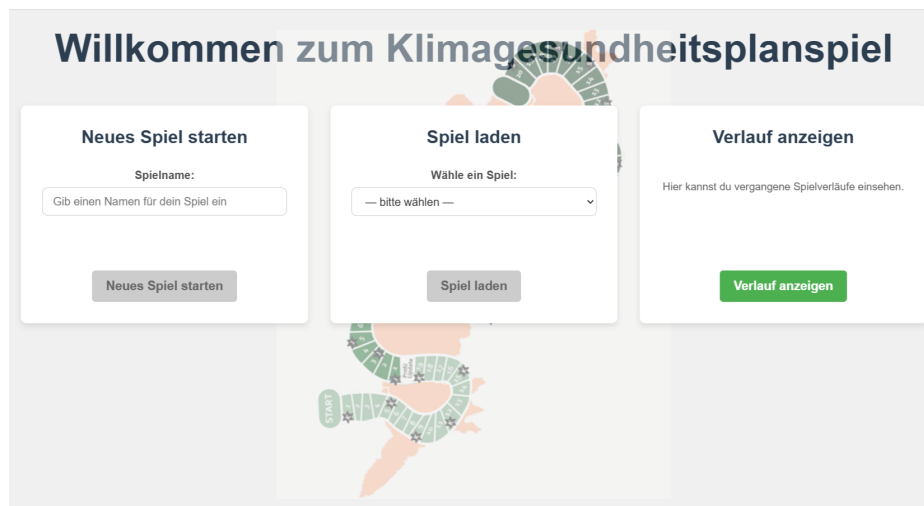
Abschließend war unser Projekt auf jeden Fall eine sehr lehrreiche Zeit und ein interessanter Einblick in die Welt des Pegelwesens. Besonders spannend war unser Besuch bei der Schleuse Nussdorf, wo Pegelhub in Zukunft eingesetzt werden soll. Dort konnte wir sehen, wie ein Pegel funktioniert, nachdem wir uns davor nur theoretisch damit befasst hatten.



Abbildung 23: *Übergabe an der Schleuse Nussdorf, Wien*

Autor*innen: Dominique Haller

Klima-Gesundheitsplanspiel Burgenland



Projektauftraggeber*in: Hochschule Burgenland, Department Gesundheit und Soziales
Projektbetreuer*in: Dipl.-Ing. Wolfram Rinke
Projektmitglieder: Lara Kammerer, Lukas Neuberger, Vanessa Prinz, Sascha Schitter
Repository: <https://github.com/fhburgenland-bswe/ss2025-klimagesundheitsplanspiel-backend>

*Das Klima-Gesundheitsplanspiel Burgenland wurde für das Department Gesundheit und Soziales der Hochschule Burgenland digital erweitert. Das bereits analog existierende Brettspiel wird nun durch eine digitale Anwendung für PC und Smartphone ergänzt. Diese unterstützt Spielleiter*innen und Spieler*innen durch eine übersichtliche Darstellung der Avatar-Steckbriefe, Lebensphasen und Ereignisse bei den unterschiedlichen Spielfeldern. Ziel ist es dadurch, Diskussionen im Spiel zu fördern, Spielzüge effizienter zu gestalten, Spielverläufe besser nachvollziehbar zu machen und die Verwaltung für die Spielleitung zu erleichtern. Darüber hinaus soll die digitale Anwendung die sozialpädagogische Lehre an der Hochschule sowie bei Bildungspartner*innen stärken.*

Auftraggeber

Die Hochschule für Angewandte Wissenschaften Burgenland hat ihren Hauptsitz in Eisenstadt sowie einen weiteren Campus in Pinkafeld. Prof. (FH) Mag. Dr. Erwin Gollner, MPH MBA ist in seiner Position als Leiter des Departments Gesundheit und Soziales der Auftraggeber. Ansprechpartnerin für das Projekt ist Anita Eggenbauer, BA. Weiters wirkt Dipl.-Psych. Andrea Stitzel unterstützend, da sie das Spiel bereits in physischer Form in der FH Kärnten in gesundheitsbezogenen Studiengängen eingeführt hat. Das Klima-Gesundheitsplanspiel

soll die Auswirkungen auf den Verlauf des Lebens durch unterschiedliche sozioökonomische Umstände (z.B. Geschlecht, Herkunft, gesellschaftlicher Status) hervorheben. Es kommt ursprünglich aus Kanada und wurde bereits von mehreren Ländern mit deren Gegebenheiten übernommen. Das Spiel als solches existiert bisher nur als Brettspiel. Das Department Gesundheit und Soziales möchte dieses Spiel ebenfalls adaptieren und in der Lehre der Hochschule Burgenland sowie bei diversen Bildungspartner*innen einsetzen. Zur Unterstützung des Spielverlaufs sollen Teile des Spiels digitalisiert werden.

Projektbeschreibung

Das bereits von der FH Kärnten übernommene Brettspiel „Klima-Gesundheitsplanspiel Kärnten“ soll für den Einsatz an der Hochschule Burgenland aufbereitet werden. Das Spiel soll an die Gegebenheiten des Burgenlandes angepasst werden und ein dazu passendes neues Spielbrett erhalten.

Die Aufgabe des Projektteams bestand darin, eine digitale Unterstützung für Spielleiter*innen sowie Spieler*innen zu entwickeln. Die Anwendung soll die Durchführung einzelner Spielzüge vereinfachen, indem Informationen leichter zugänglich und schneller verteilt werden, was wiederum mehr Raum für Diskussion schafft. Zudem fördert die Digitalisierung des Spiels die Nachvollziehbarkeit des Spielverlaufs zum Ende des Spiels. Dies wird erreicht durch eine übersichtliche Darstellung der einzelnen Lebensphasen, die ein Avatar bewältigt hat.

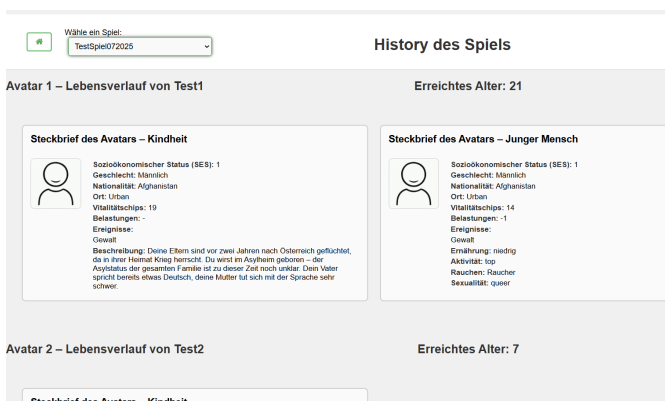


Abbildung 24: Historyseite-Ansicht der Spielleitung

Die Anwendung wurde mit Java Spring Boot im Backend sowie Angular als Frontend-Framework in Verbindung mit einer PostgreSQL Datenbank realisiert. Die Wahl fiel auf diese Technologien, da diese bereits im Team bekannt waren und es keine Vorgaben seitens der Auftraggeber*innen diesbezüglich gab. Es wurden zwei Frontends implementiert, jeweils eines für Spielleiter*in und Spieler*innen, um die Funktionalitäten zu trennen.

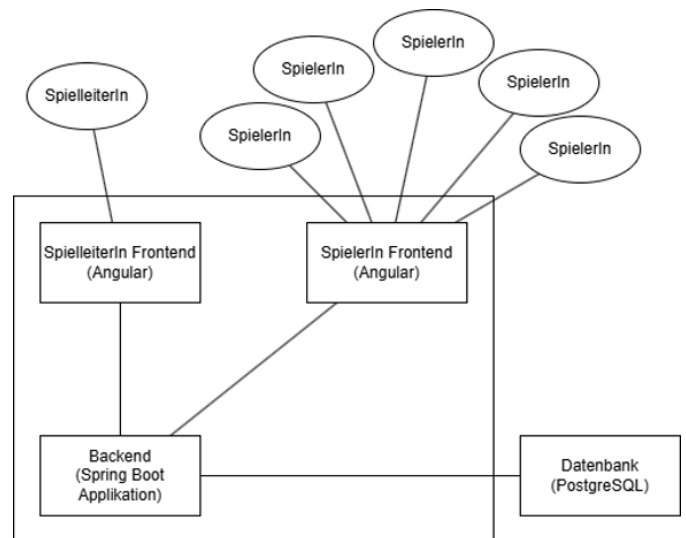


Abbildung 25: Übersicht der Systemarchitektur

Für das Spiel wurden die folgenden Funktionalitäten implementiert, die auch als Anforderung durch die Auftraggeber*innen festgelegt wurden:

Funktionen für die Spielleitung:

- Spiele erstellen, laden und verwalten
- Spielende einladen und Avatare zuweisen
- Spielfelder-Interaktionen darstellen
- Vitalitätschips, Gemeinschafts- und Belastungskarten verwalten
- Steckbriefe der Avatare anzeigen und bearbeiten
- Spielverlauf einsehen

Funktionen für die Spieler*innen:

- Eigenen Avatar-Steckbrief mit aktuellen Werten einsehen
- Änderungen der Spielleitung mitverfolgen
- Ereignisse selbst eintragen
- Namen der Spieler*innen anpassen

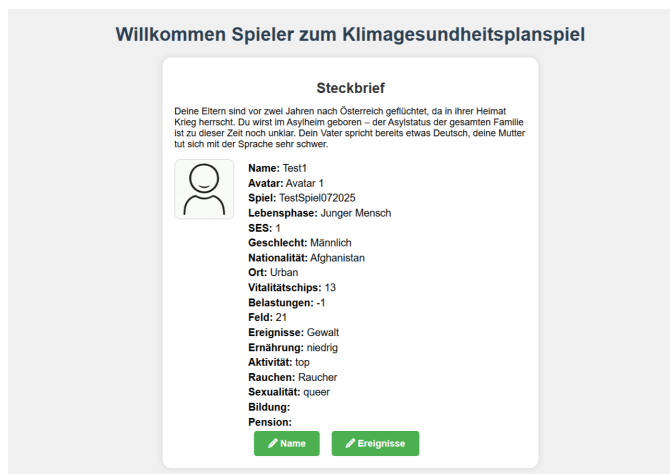


Abbildung 26: Avatarseite-Ansicht der Spielenden

Projektverlauf

Mit dem ersten Refinement mit den Auftraggeber*innen am 10. März 2025 startete das Projekt. Insgesamt wurde das Projekt in einer Vorbereitungsphase und fünf Sprints realisiert. Ein Sprint umfasste einen Zeitraum von zwei Wochen und wurde jeweils durch ein Sprint Planning zu Beginn und ein Sprint Review zum Ende eingegrenzt. Einen Abschluss fand das Projekt mit der Abschlusspräsentation am 13. Juni 2025 sowie der Endübergabe an die Auftraggeber*innen am 24. Juni 2025.

In der Anfangsphase des Projekts wurden nach dem ersten Gespräch mit den Auftraggeber*innen Wiremocks erstellt und direkt Feedback eingeholt. Diese dienten zu einem gemeinsamen Verständnis des Endprodukts. Auf Basis dessen konnte ein ER-Modell gemeinsam mit einem ersten Entwurf einer API erstellt werden. Durch diese Vorbereitungen konnte ein grundsätzlicher Überblick über die Gesamtarchitektur gewonnen werden. Zudem ermöglichte diese Vorgehensweise eine weitgehend parallele und unabhängige Bearbeitung einzelner User Stories.

Zur Realisierung des Projekts wurden bei großen User Stories Subtasks zur Trennung der Frontend- und Backend-Funktionalitäten angelegt. Die jewei-

ligen Wiremocks wurden in die dafür vorgesehene User Story hinzugefügt. Dies diente als Unterstützung der Akzeptanzkriterien und half somit beim Testen und bei der Abnahme durch den PO.

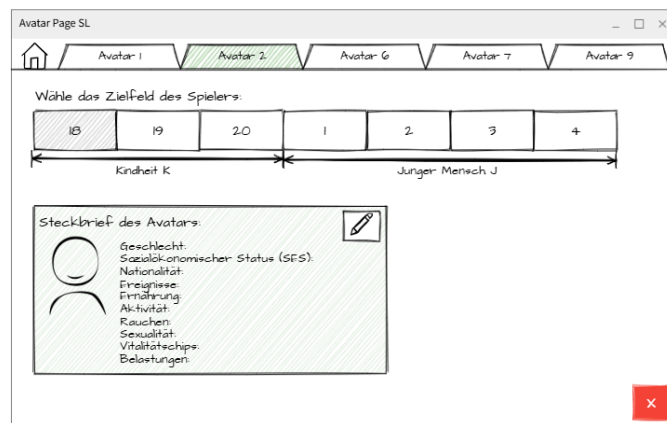


Abbildung 27: Wiremock der Avatarseite

Fazit

Das Praxisprojekts brachte zahlreiche wertvolle Erkenntnisse, sowohl im technischen als auch im organisatorischen Bereich. Die Umsetzung mit den gewählten Technologien festigte bestehendes Wissen und half dem Team, dieses weiter zu vertiefen. Darüber hinaus ermöglichte das Projekt, praktische Erfahrungen in der Entwicklung komplexer Anwendungen zu sammeln. Der Einsatz agiler Methoden sowie projektunterstützender Tools trugen wesentlich zum Projekterfolg bei.

Eine zentrale Erkenntnis war, wie wichtig eine klare Kommunikation sowohl innerhalb des Team, als auch mit den Auftraggeber*innen ist. Ebenso entscheidend war ein gemeinsames Verständnis von dem gewünschten Endprodukt.

Abschließend lässt sich sagen, dass das Projekt nicht nur unsere Fähigkeiten als Entwickler*innen gestärkt hat, sondern auch verdeutlichte, wie herausfordernd und komplex Projektmanagement sein kann.

Autorinnen: Lara Kammerer, Vanessa Prinz

Projektmitglieder – BSWE Jahrgang 2023

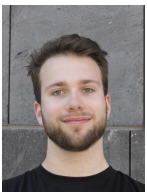
gAmlcode



Nina Hohl
Hochschule Burgenland



Lukas Hofer
Hochschule Burgenland



Leon Rasner
Hochschule Burgenland



Tobias Pavel
Hochschule Burgenland

Elektroauto Lade-App



Andreas Dötzl
Hochschule Burgenland



Daniel Rimpfl
Hochschule Burgenland



Marcel Turobin-Ort
Hochschule Burgenland

I NEED HELP



Marcel Wagner
Hochschule Burgenland



Stefan Hahnl
Hochschule Burgenland



Jan Jungwirth
Hochschule Burgenland

VoiceArt



Gloria Brenner
Hochschule Burgenland



Dario-Benjamin Dzakic
Hochschule Burgenland



Michael Heckenast
Hochschule Burgenland

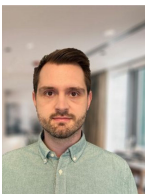
Atlas der Klima-bezogenen-Gesundheitsrisiken



Stephan Auer
Hochschule Burgenland



Manuel Neuhold
Hochschule Burgenland



Ábel Horváth-Szarka
Hochschule Burgenland



Richard Gellen
Hochschule Burgenland

NEXUS



Georg Trnka
Hochschule Burgenland



Miklos Komlosy
Hochschule Burgenland



Sourì Rene
Hochschule Burgenland

Pegelhub



Armin Grebić
Hochschule Burgenland



Christian Schlögl
Hochschule Burgenland



Dominique Haller
Hochschule Burgenland



Efnan Evelioglu
Hochschule Burgenland

Klima-Gesundheitsplanspiel Burgenland



Lara Kammerer
Hochschule Burgenland



Lukas Neuberger
Hochschule Burgenland



Vanessa Prinz
Hochschule Burgenland



Sascha Schitter
Hochschule Burgenland

Projektbetreuer*innen



Dipl.-Ing. Franz Knipp

Studiengangsleiter des Bachelorstudiengangs Software Engineering und vernetzte Systeme an der Hochschule Burgenland



Dipl.-Ing. Wolfram Rinke

Fachhochschullehrer am Department für Informationstechnologie an der Hochschule Burgenland



Julia Pausch, BSc MSc MA

Externe Lektorin für die Lehrveranstaltung Projektmanagement



Renate Weichselbraun

Externe Lektorin für die Lehrveranstaltung Softwarequalität



Hans Sowa, BSc

Externer Lektor für die Lehrveranstaltung Softwarequalität

Verzeichnisse

Abbildungsverzeichnis

1	Übersicht der Benutzerverwaltung	3
2	Benutzerregistrierung über einen neu generierten Link	3
3	Bearbeitungsansicht eines Kurses	4
4	Phase 1	6
5	Phase 2	6
6	Phase 3	6
7	Phase 4	7
8	Die Übergabe	7
9	Die Applikation	9
10	Das Team	9
11	Ergebnis-Screen von Hit The Points!	11
12	TreasureHunt	11
13	FreeDraw mit Farbauswahl	12
14	API Endpunkte des Caching Backend Servers	15
15	Finales UI der Applikation	15
16	Screenshot Page – Live-Überblick	17
17	Screenshot Page – Diagramm-Analyse	17
18	Screenshot Page – Statistik	17
19	Screenshot Page – Tabelle nach Sensorwahl	18
20	Architektur von Pegelhub	19
21	Aufbau des Clusters	20
22	Pegelhub in Docker	21
23	Übergabe an der Schleuse Nussdorf, Wien	21
24	Historyseite-Ansicht der Spielleitung	23
25	Übersicht der Systemarchitektur	23
26	Avatarseite-Ansicht der Spielenden	24
27	Wiremock der Avatarseite	24

Abkürzungsverzeichnis

AGES	Österreichische Agentur für Gesundheits- und Ernährungswissenschaften.
Angular	Javascript/TypeScript-Framework für Web-Frontends.
API	Application Programming Interface.
AWS	Amazon Web Services.
CMS	Content Management System.
COPD	Chronic Obstructive Pulmonary Disease.
EAS	Expo Application Services.
ER	Entity-Relationship-Modell.
FFT	Fast Fourier Transformation.
FTP	File Transfer Protocol.
GHCR	GitHub Container Registry.
HTTP	Hypertext Transfer Protocol.
ICC	Inter-Cluster Communication.
IEC	International Electrotechnical Commission.
InfluxDB	Zeitserienorientierte Open-Source-Datenbank (InfluxData).
IQR	Interquartilsabstand (engl. <i>interquartile range</i>).
PO	Product Owner.
REST	Representational State Transfer.
RFID	Radio Frequency Identification.
Spring Boot	Java-Framework zur schnellen Erstellung von Web-Backends.
Tukey	Methode zur Ausreißerererkennung basierend auf dem Interquartilsabstand.
UI	Userinterface.
ZAMG	Zentralanstalt für Meteorologie und Geodynamik.

Impressum

Hochschule für Angewandte Wissenschaften Burgenland GmbH

Studienzentrum Eisenstadt | Campus 1 | 7000 Eisenstadt

Studienzentrum Pinkafeld | Steinamangerstraße 21 | 7423 Pinkafeld

Tel.: +43 5 7705

E-Mail: office@hochschule-burgenland.at | www.hochschule-burgenland.at



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz. Diese Lizenz erlaubt es, den Inhalt zu vervielfältigen, zu verbreiten und öffentlich aufzuführen unter folgenden Bedingungen: Der Name des Autors/Rechtsinhabers muss genannt werden. Dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden. Bearbeiteter oder in anderer Weise umgestalteter Inhalt darf als neu entstandener Inhalt nur unter Verwendung identischer Lizenzbedingungen weitergegeben werden. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>