ChatGPT

Software Requirements Specification (SRS) for HAW Hamburg Chatbot

1. Introduction

The HAW Hamburg Chatbot is envisioned as an AI-driven virtual assistant to support current and prospective students, faculty, and staff. It will operate 24/7, providing instant answers to campus FAQs, administrative processes, and personalized guidance. According to studies of higher-education chatbots, such bots "provide instant answers to FAQs, streamline processes like course registration, and offer personalized guidance" by integrating with systems such as SIS or LMS [1] [2]. The chatbot must be fully GDPR-compliant: based on self-hosted, open-source technology (e.g. Rasa or Botpress) so that all data remains under HAW's control [3] [4]. It will support both English and German, automatically detecting user language or allowing manual selection, which is essential for inclusivity and accessibility [5]. In line with the university's culture, the bot should have a friendly, somewhat humorous "persona" to engage users naturally, while avoiding inappropriate or biased language. When the bot cannot answer a question, it will suggest human contacts or next steps (e.g. "please email Admissions at..."), ensuring smooth handoff.

2. Functional Requirements

The system shall support the following core functional areas and tasks:

- **General University Information:** Answer queries about HAW Hamburg's programs, departments, contact points, campus locations, and history. Provide up-to-date facts (e.g. academic calendar dates, holiday closures, main office hours). Support multilingual FAQs (German/English) and fallback suggestions when needed.
- **Course Enrollment & Scheduling:** Guide students through course enrollment procedures, deadlines, and registration systems (e.g. CampusOnline). Explain prerequisites or major requirements. Offer time-table and schedule information. If applicable, integrate with the Course Catalog or SIS to retrieve specific course offerings.
- **New-Student Onboarding:** Assist incoming students with orientation tasks: obtaining a student ID (CampusCard) [1], logging into the LMS (Moodle/CampusOnline), connecting to campus Wi-Fi, and accessing email. Provide step-by-step guidance (e.g. "To get your CampusCard, go to Student Services with your admission letter").
- **Accommodations & Housing:** Provide information on Studierendenwerk Hamburg services: dormitory applications, housing allowances, shared apartments, and timelines. Suggest official contacts or external resources for apartment searches. Cite the Studierendenwerk hotline or website for detailed procedures.
- **Jobs and Internships:** Answer questions about on-campus jobs, internships, and work rights (especially for international students). Explain employment rules for non-EU students. Link to HAW career services and Stellenwerk Hamburg for job postings. Suggest campus career fairs or workshops (e.g. CV clinics).

- **Career Services & Job Board (Stellenwerk):** Describe how to use Stellenwerk Hamburg (job platform) and HAW's career counseling resources. Recommend attending career events. Provide links to company directories or LinkedIn groups.
- **Campus Navigation & Facilities:** Offer directions using campus maps (buildings, lecture halls, libraries, gyms, cafeterias). Provide facility hours (e.g. library opening times, gym, labs). Answer location-based FAQs ("Where is building S?"). If possible, integrate with an interactive campus map or GPS guidance in future.
- **Library and IT Support:** Answer queries about library services (catalog search, borrowing rules, computer labs). For IT issues (e.g. login problems, Wi-Fi connectivity), provide basic troubleshooting steps. If the issue is unresolved, offer to connect the user to IT helpdesk (similar to Passau's ZIMI bot, which escalates to a ticket system) [6].
- **Student Wellness & Mental Health:** Provide information on counseling centers, peer-mentoring programs, and emergency contacts (e.g. psychotherapeutic services). Direct students to mental health resources on campus (e.g. student health service).
- **Event Notifications & Reminders:** Allow users to subscribe to updates about important deadlines (enrollment deadlines, exam periods), orientation events, and campus news. Send proactive, consent-based notifications (e.g. exam timetable reminders via email or chat).

Each feature should be phrased in a helpful, conversational style, with friendly/humorous touches (e.g. "Don't worry, I don't ask tough exam questions!"). When an answer is uncertain or not found, the chatbot should not leave the user stranded: it will suggest contacts (e.g. "You might want to ask the Student Services office at studserv@haw.de") or relevant FAQs.

3. Non-Functional Requirements

- **Multilingual Support:** The chatbot shall operate fully in German and English. It must accurately detect the user's language or allow switching on demand. Responses should maintain consistent tone across languages. (This aligns with best practices for accessibility and inclusivity [5].)
- **Performance and Availability:** The system shall be available 24/7 with high reliability (aim for $\geq$99% uptime). Queries should be processed quickly (target response time under 2 seconds under normal load). The system should scale to handle peak loads (e.g. start of semester) without significant degradation.
- **Usability and Persona:** The interface shall be intuitive and responsive on desktop and mobile browsers. The chatbot's dialogue persona should be warm, helpful and occasionally humorous (e.g. light jokes, emotive language) to improve engagement. (Evidence shows "social cues" like human-like phrasing increase user satisfaction [1].) However, it must avoid any inappropriate remarks.
- **Open-Source/Self-Hosting:** All core technology must be open-source and hostable on HAW's own servers. Recommended platforms include Rasa or Botpress. Rasa is a mature open-source conversational AI framework that runs fully on-premises [4]; Botpress similarly offers an open-source NLU and dialog builder [3]. This avoids reliance on third-party cloud services and ensures GDPR compliance.
- **Security:** The system shall use HTTPS for all client-server communication. All administrative interfaces shall be protected by strong authentication (university SSO if possible). Data at rest (logs, knowledge base) shall be encrypted. Only necessary user data (e.g. session identifiers) will be stored.
- **Data Privacy (GDPR):** The chatbot shall minimize data collection. No personal identifiers (names, student IDs, birthdates, health info) will be stored unless explicitly needed and with user consent. By default, the bot should operate anonymously: e.g. as with TH Köln's StudiCoachBot, all user

interactions must be fully anonymized and stripped of identifiable data [7] [8]. Any conversation logs will remove personal details before storage. If third-party services (e.g. language models) are used, user input must be anonymized first. An explicit privacy notice will be presented before starting chat, and users can opt out or reset the conversation at any time.
- **Maintainability:** The content database (FAQ pairs, knowledge base) should be easy for HAW staff to update via an admin UI or simple files. The system should log questions and feedback (anonymously) so that administrators can identify missing answers and improve coverage over time.

4. System Architecture

The chatbot system will follow a modular architecture with the following components:

- **User Interface:** A web-based chat widget embedded on HAW Hamburg's website (and possibly a mobile app or messaging platforms later). It should support text input (and future speech input), and display rich responses (text, images, links).
- **Chatbot Engine:** The core NLP and dialog manager, implemented with an open-source framework like Rasa or Botpress. This handles intent classification, entity extraction, dialogue flow, and response generation. It will access internal knowledge sources to formulate answers. Rasa, for example, uses a "stories" approach to dialogue training and can be extended with custom actions [4].
- **Knowledge Base and Datastore:** A centralized repository of campus information. This could be a combination of structured data (course catalog, event calendar, building info) and unstructured FAQ documents. The system might use a search index (e.g. Elasticsearch) or RAG (Retrieval-Augmented Generation) to fetch relevant content. For instance, Ulm's "Ulma" bot uses RAG on verified university documents to ensure factual accuracy [9]. All data (course catalog, calendars, housing info) should be obtained via APIs or local databases maintained by the university.
- **External Integration Services:** Interfaces to HAW systems and third-party services. Key integrations include:
- **SIS/Enrollment APIs:** For real-time course schedules or exam dates (if permitted), enabling personalized responses.
- **Student Services Portal (CampusOnline/Moodle):** To guide students on accessing resources.
- **Authentication Service:** If login is supported, integrate with the university's Single Sign-On (CAS) to recognize users.
- **Studierendenwerk and Stellenwerk Feeds:** APIs or RSS feeds for housing and jobs info.
- **Event Calendar / Notification Service:** To push reminders about deadlines or events (subscriptions).
- **Logging and Analytics:** Internal logging of queries (anonymized) and performance metrics. An analytics dashboard for admins to review question trends and bot effectiveness. Logs will be stored securely with any PII redacted [8].
- **Deployment Infrastructure:** The components will be containerized (e.g. Docker) and deployed on HAW Hamburg's servers or cloud under the university's control. A load balancer can ensure availability. Maintenance tasks (updates, backups) will follow university IT policies.

This architecture ensures all data remains on-premises (critical for GDPR) while allowing flexible integration. Open-source platforms like Rasa and Botpress have been successfully used in universities and support enterprise features (visual flow editing, versioning) [3] [4].

5. Security and Privacy Safeguards

The chatbot must adhere to strict security and privacy standards:

- **GDPR Compliance:** By default, the bot will not store personal data. Any needed personal data (e.g. when customizing notifications) will be processed only with explicit consent. A privacy notice will explain data handling and allow users to withdraw consent.
- **Data Minimization and Anonymization:** Following TH Köln's example, conversation logs will be purged of identifiers before analysis [8]. For example, names or ID numbers will be stripped from transcripts. This "anonymized logs" approach ensures analytics and training data contain no direct identifiers.
- **Local Processing:** All sensitive processing will occur on university servers. For example, in HHU Düsseldorf's hAInrich chatbot, the language model runs entirely in the user's browser (via ONNX and Transformers.js) so "your data does not leave the browser" [10]. While our bot may use a server backend, we will similarly minimize external calls. If any API calls to third-party LLMs are made, they will use only abstracted queries (no PII) and ensure encrypted channels.
- **Encryption and Access Control:** All communications will use TLS/HTTPS. Database and log storage will be encrypted at rest. Admin interfaces require authenticated login (preferably tied to staff AD accounts). Regular security audits and code updates will mitigate vulnerabilities.
- **Consent Management:** The bot will use cookie-consent mechanisms for any stored data (e.g. session cookies). Users will be given clear options to accept or decline optional features (like saving favorites).
- **Error Handling and Privacy:** In cases of fallback or unrecognized queries, the bot will not query external sources with private user inputs. It will either handle internally or politely ask the user to rephrase.

These measures ensure compliance with the EU's GDPR (e.g. data minimization, user consent) and reflect practices used by other German university chatbots like StudiCoachBot, which explicitly assures users of full anonymization [7] [8].

6. Integration Points

To provide accurate and up-to-date information, the chatbot will integrate with existing HAW Hamburg systems and external data sources:

- **Student Information System (SIS)/Enrollment API:** To fetch current course lists, exam dates or enrollment statuses. Integration with an SIS allows tailored guidance (e.g. "You're enrolled in 3 courses next semester" or "Registration closes in 2 days") [2].
- **Learning Management System (LMS) – CampusOnline/Moodle:** To assist with login issues or provide links to course materials. For example, the bot can guide students on resetting Moodle passwords or accessing lecture slides.
- **CampusCard/ID System:** While student ID issuance is offline, the bot can link to relevant online forms or appointment bookings for card pickup.
- **Studierendenwerk Hamburg API/Website:** If available, pull housing application deadlines and dorm availability. Otherwise, present static FAQs about student housing and link to the Studierendenwerk portal.

- **Stellenwerk Hamburg and Career Services:** Link to the university's job board (Stellenwerk) and the career center's announcements (e.g. internship fairs). These could be fetched via RSS or simple web queries.
- **Campus Map/Navigation Service:** Use or develop a simple map interface. Possibly integrate with a GIS service or embed Google Maps with campus layers.
- **Library System:** Query library catalog APIs to assist with book searches, or provide current opening hours and contact info.
- **Event Calendar:** Fetch data from the official academic calendar (SIS or uni CMS) for events, exam periods, and deadlines to automate reminders.
- **Authentication (Optional):** If personalized features are offered, integrate with HAW's Single Sign-On (Shibboleth/CAS) so students can log in with their credentials.
- **Notification Services:** To send email or SMS notifications, the bot can integrate with the university's existing mailing list software or notification API. Users would opt in via the chat interface.

These integrations follow higher-education chatbot best practices: bots that connect to SIS/LMS can deliver tailored experiences [2]. All integration endpoints must be secured (HTTPS, API keys) and respect data privacy (e.g. do not share more data than needed).

7. User Roles and Access Levels

The chatbot system will recognize different user roles with appropriate access:

- **Guest (Unauthenticated User):** Any visitor to the website (prospective student, parent, staff) can ask general questions. They have no special access and all responses are generic.
- **Student (Authenticated):** Registered HAW students who log in (optional). They may receive personalized information (e.g. "Your next exam is tomorrow") if SIS integration is enabled. They can also subscribe to personal notifications (e.g. exam reminders).
- **Staff/Faculty:** University employees might use the bot to quickly check administrative procedures (e.g. booking a classroom, ordering office supplies) if relevant. Their role can be distinguished by login, offering advanced options like submitting feedback on the bot's answers.
- **Administrator:** HAW IT or chatbot team members have an admin dashboard to update the knowledge base, review analytics, and manage system settings. Admins can view anonymized logs and user feedback. They can push updates to the FAQ database and training data.
- **Bot Maintainers:** Developers or data curators (e.g. content specialists) who train the NLP model and refine answers. They typically use separate tools (like Rasa X) and need privileged access to retrain or modify the dialog flows.

Each role has appropriate data access: e.g., a student's personal schedule should not be exposed to guests. Admin interfaces will be protected by role-based authentication.

8. Future Roadmap

Beyond the initial launch, the following enhancements are planned:

- **Voice Input/Output:** Integrate speech recognition (STT) and synthesis (TTS) so users can talk to the bot (on phone or in-person kiosk). This can be built using cloud STT/TTS APIs or local open-source tools. (Platforms like Rasa can support voice by adding STT/TTS layers [11].)

- **Advanced AI Modules:** Incorporate more sophisticated language models for open-ended queries. For example, using a small fine-tuned LLM (on HAW data) with Retrieval-Augmented Generation can improve coverage without risking hallucinations (as Ulm's "Ulma" did [9]). Any generative model would be used with strict filtering and only to suggest information drawn from verified sources.
- **Multi-Platform Deployment:** Expand to other channels such as a dedicated mobile app, MS Teams/Slack integration, or social media (Facebook Messenger) to meet students where they communicate.
- **Expanded Language Support:** Eventually add more languages (e.g. Turkish, Russian) to serve the diverse student body, as Ulm did for its international office [12].
- **Personalization and Context:** Improve context retention across sessions and allow user profiles/preferences (e.g. saved favorite topics). Integration with the campus portal could enable truly personalized recommendations (courses, events) based on a student's major and interests.
- **Data Analytics & AI Coaching:** As with the StudiCoachBot, analyze anonymized user interactions to identify stress points (like common confusing policies) and continually refine the bot. Potentially add a "coaching" sub-module for student well-being using AI-driven motivational dialogues, following ethical guidelines [13].
- **Accessibility Improvements:** Ensure the chatbot conforms to WCAG standards (screen reader compatibility, high contrast mode). Possibly add a sign language avatar or text simplification.
- **Monitoring and A/B Testing:** Implement A/B testing of different response styles or flow designs to optimize user satisfaction. Use analytics to iteratively improve the knowledge base and dialog flows.

These roadmap items keep the system up-to-date with technology trends and evolving user needs.

9. Market Research

Several German universities have already launched GDPR-compliant chatbots, providing valuable lessons for HAW Hamburg. The table below compares four notable examples:

Chatbot	Technology (Platform)	Key Features	Privacy/Compliance	Official Info
Ulma (Ulm University)	Hybrid AI (rule-based + LLM), Retrieval-Augmented Generation [9]	International Office chatbot; answers study programs, scholarships, student life; ~30 languages [12] [14]	On-premises; uses only verified university data (via RAG) to ensure accuracy [9]; developed by Kauz.ai (German provider)	Ulm University News (July 2022) [15] [9]

Chatbot	Technology (Platform)	Key Features	Privacy/Compliance	Official Info
StudiCoachBot (TH Köln)	Hybrid Rasa chatbot (rule-based + LLM via OpenAI) [13] [7]	Self-coaching chatbot for exam anxiety; interactive Q&A flow; provides coping strategies	Explicit privacy design: no personal data collected, conversations anonymized. Runs on TH Köln servers; user inputs anonymized before reaching LLM [7].	TH Köln Project Page [7] [13]
ZIMI (University of Passau)	AI chatbot (likely local rule-based/ NLP engine) [16]	IT support assistant (German/ English); answers tech FAQs; available 24/7; can escalate to helpdesk ticket [6]	Accessible only to users with campus ID ("ZIM ID") [17]; data from large internal knowledge base. Presumably on-premise with no external data sharing.	Uni Passau News (July 2025) [6] [17]
hAInrich (HHU Düsseldorf)	Client-side AI search (all-MiniLM-L6-v2 via ONNX, Transformers.js) [10]	Lightweight Q&A for Master's program; answers questions based on FAQ page; no generative "hallucinations" [18]	Entirely client-side processing: user queries are matched locally, **"your data does not leave the browser"** [10]. Since no server storage of queries, GDPR risk is minimal.	HHU Düsseldorf FAQ Page [18] [10]

Each example demonstrates strong data protection. For instance, TH Köln's bot explicitly anonymizes all inputs before use [7], and HHU Düsseldorf's hAInrich operates entirely locally so no data is sent to a server [10]. Ulm's Ulma ensures only vetted institutional data is used in answers [9]. These cases confirm the feasibility of a GDPR-compliant, bilingual campus chatbot. The key lessons are to use on-premises or client-only processing, anonymize logs, and rely on open-source frameworks (Rasa, ONNX, etc.) [3] [4].

Sources: Official university pages and project descriptions were used for this analysis [15] [9] [7] [13] [17] [6] [18] [10]. These highlight each bot's technology stack and GDPR safeguards.

---

[1] [2] [5] How to Use Chatbots for Higher Education in 2025
https://botpress.com/blog/chatbot-for-education

3 4 11 14 Best Open Source Chatbot Platforms to Use in 2025
https://botpress.com/blog/open-source-chatbots

6 16 17 Meet ZIMI – the IT support chatbot for students and staff • University of Passau
https://www.uni-passau.de/en/bereiche/university-staff/news-for-staff-members/news/meet-zimi-the-it-support-chatbot-for-students-and-staff

7 8 13 StudiCoachBot - TH Köln
https://www.th-koeln.de/anlagen-energie-und-maschinensysteme/studicoachbot_100424.php

9 12 14 15 Digital owl supports students with counselling|KI chatbot from the International Office speaks 30 languages - Universität Ulm
https://www.uni-ulm.de/en/university-news/news-details/article/digitale-eule-unterstuetzt-bei-der-beratung-von-studierenden-in-30-sprachenio-der-uni-ulm-startet-als-erstes-im-land-mit-eigenem-ki-chatbot/

10 18 Get answers quickly
https://www.heicad.hhu.de/en/lehre/masters-programme-ai-and-data-science/hainrich