Swoplet

Milestone 1: Use Cases, High-Level Requirements, and Architecture

Master Team Project / GDSD Summer 2025
Team 6

Team Lead Email: muhammad-hamza.butt@informatik.hs-fulda.de

Team Members	Role
Hamza Butt	Team Lead
Karan Patel	Github Master
Pooja Puthu Vayal	Backend Lead
Akhil Sajan	Frontend Lead
Hammad Asif	Frontend Developer

History

Date	Version	Notes
08/05/2025	First Draft	Initial Submission

1. Executive Summary

At Fulda University of Applied Sciences, students and staff frequently encounter obstacles when trying to buy, sell, or exchange everyday items within a trustworthy local network. Public online marketplaces often fall short, exposing users to privacy risks, scams, and unreliable interactions with strangers outside the university. These challenges not only discourage sustainable practices like reuse and sharing but also hinder the development of a connected, supportive campus community.

To address these issues, we have launched Swoplet, a dedicated, web-based platform designed exclusively for the Fulda University community. Access is strictly limited to students, faculty, and staff with a valid hs-fulda.de email address, ensuring that every participant is a verified member of the university. This closed, secure environment eliminates the uncertainties of public marketplaces, offering a reliable space where users can confidently engage in peer-to-peer transactions. The platform's intuitive interface makes posting, searching, and communicating effortless, empowering users to participate in a safe and supportive campus marketplace.

What sets Swoplet apart are its innovative features, crafted specifically for the Fulda student lifestyle. Users can easily browse or list items with smart filters, searching by category, price, condition, and availability to find exactly what they need. The secure in-app messaging system allows for direct, private communication between buyers and sellers, while all listings are carefully moderated before publication to maintain quality and trust. Community safety is further enhanced by a robust reporting feature, enabling users to flag inappropriate or suspicious content.

Additional highlights include Wishlist tracking, allowing users to monitor items they're interested in; product comparison tools to help make informed decisions; and a flash deal system that promotes limited-time offers, adding excitement and urgency to the marketplace. The platform's responsive design ensures a seamless experience on laptops, tablets, and smartphones, making it convenient for users whether they're on campus or on the move.

Swoplet is developed by a passionate, student-led team at Fulda University of Applied Sciences. Drawing on firsthand experience with the challenges faced by their peers, the team combined technical expertise with a deep understanding of campus life to create a platform tailored to the unique needs of the Fulda community. Their collaborative approach ensured that every feature was designed with user experience, safety, and convenience in mind.

By providing a secure, user-friendly, and feature-rich marketplace exclusively for Fulda University, Swoplet fills a critical gap in campus life. It not only encourages sustainable habits and resource sharing but also strengthens the bonds within the university community. Ultimately, the platform empowers students and staff to connect, collaborate, and thrive in a trusted environment, making Fulda University an even better place to live, learn, and grow.

2. Personae and Main Use Cases

2.1 Persona 1: Lina – Student Seller (Hochschule Fulda)

Relevant Functional Requirements: FR1, FR2, FR3, FR4, FR5, FR10, FR15, FR20

General Characteristics: Lina is a 22-year-old computer science student at Hochschule Fulda. She lives on campus and frequently sells used textbooks and dorm essentials. She's comfortable with digital platforms and prefers mobile-friendly interfaces.

Goals:

- Sell her items quickly and easily.
- Update or temporarily hide listings as needed.
- Ensure buyers are from her university community.

Pain Points:

- Wasting time reposting expired listings manually.
- Difficulty managing multiple listings.
- Insecure selling experiences on public marketplaces.

2.2 Persona 2: Alex – Budget-Conscious Buyer (SFSU Exchange Student)

Relevant Functional Requirements: FR1, FR6, FR7, FR9, FR14, FR16, FR18, FR19

General Characteristics: Alex is a 25-year-old exchange student from SFSU spending one semester in Fulda. He needs basic furniture and appliances but has a tight budget.

Goals:

- Search, filter, and sort listings quickly.
- Track price drops and find deals.
- Save interesting listings for later comparison.

Pain Points:

- Inaccurate or outdated listings.
- Unclear communication with sellers.
- Listings that don’t reflect current prices or availability.

2.3 Persona 3: Sarah – Platform Admin (Graduate Assistant)

Relevant Functional Requirements: FR11, FR12, FR13, FR22, FR23

General Characteristics: Sarah is a graduate assistant tasked with moderating the platform. She works part-time and needs efficient tools to keep the platform clean and compliant.

Goals:

- Approve or reject listings quickly.
- Address inappropriate content or user reports.
- Respond to support queries and maintain platform integrity.

Pain Points:

- High volume of spammy or incomplete posts.
- Limited moderation tools on other platforms.
- Repetitive support questions from users.

2.4 Main Use Cases

UC1 – Create and Manage a Listing (FR1, FR2, FR3, FR4, FR5, FR10, FR15, FR20)

Lina logs into the platform using her university email (FR1). She updates her profile with a photo and short bio (FR2). She wants to sell a microwave, so she clicks on

“Post New Item,” fills in the title, description, price, and category (FR3, FR5), and uploads four photos (FR4). After the item is approved by an admin (FR11), the listing becomes public. A week later, she updates the description and changes the price (FR10). She later hides the listing temporarily while out of town (FR20). If no sale happens, it auto-expires after 30 days (FR15).

UC2 – Discover, Watch, and Buy Items (FR6, FR7, FR9, FR14, FR16, FR18, FR19)

Alex visits the site to find a secondhand bike. He searches “bike” and applies a filter for listings under €100 (FR6, FR7). He bookmarks two options in his Watchlist (FR9). One listing has a visible price history showing a drop from €120 to €90 (FR18), and it's marked in the “Price Drops” section (FR19). After buying, Alex leaves a 5-star review for the seller (FR14). He shares another listing link with his roommate via WhatsApp (FR16).

UC3 – Safe Buyer-Seller Communication (FR1, FR8, FR14, FR24)

After seeing a listing for a used monitor, Alex messages the seller directly through the platform without sharing emails (FR8). They negotiate price and pick-up time. After purchase, both users leave a review and rating for each other (FR14). Later, Alex deletes several old conversations to clean his inbox (FR24).

UC4 – Admin Moderation and User Support (FR11, FR12, FR13, FR22, FR23)

Sarah logs into the admin dashboard and sees 18 pending listings (FR11). She rejects one listing due to offensive content and bans the user for repeated violations (FR12). She reviews a flagged item reported by another user (FR13) and takes appropriate action. She then checks the support inbox where users submitted questions via the Contact Form (FR22). Finally, she updates the FAQ section with new answers based on recurring issues (FR23).

UC5 – User Registration and Recovery (FR1, FR17, FR21)

New student Ahmad wants to join the platform. He registers using his hs-fulda.de email (FR1) and is required to accept the Terms of Service and Privacy Policy (FR17). A few weeks later, he forgets his password and uses the reset option to set a new one (FR21). He logs back in, updates his profile, and begins browsing.

3. Main Data Items and Entities

3.1 User Types and Privileges

User Type	Description	Privileges
Guest	Unregistered visitor	View listings, apply filters
Buyer	Verified Fulda user (buyer)	Register/login, search/filter, message, favorites, compare, report
Seller	Verified Fulda user (seller)	Verified Fulda user (seller) CRUD listings, upload media, respond to messages, temporary hides, report
Moderator	System user with moderation privileges	Approve/reject listings, manage reports, monitor messages, warn/ban users, manage expirations

3.2 Main Data Entities and Descriptions

Entity Name	Description
User	Registered individual (Guest, Buyer, Seller, Moderator)
Product Listing	Item for sale; includes title, description, price, condition, images, etc.
Media File	Images/videos uploaded and moderated
Message	In-app chat communication between users
Favorite	Saved product for later comparison

Report	Complaint against a listing or user
Search History	Log of user search queries and filters
Price Comparison	Table for comparing similar products
Time Limit	Listing expiration settings

3.3 Logical Data Structure (High-Level Overview)

1. User:

- user_id
- name
- email
- password_hash
- role (guest, buyer, seller, moderator)
- profile_picture
- registered_at
- is_active

2. Product Listing:

- product_id
- title
- description
- category
- price
- condition (new, used)
- tags (electronics, laptop, HP)
- location
- created_by_user_id
- status
- created_at
- updated_at

3. Media File:

- media_id
- product_id
- file_path
- is_approved
- uploaded_at

4. Message:

- message_id
- sender_id
- receiver_id
- product_id
- content
- created_at

5. Favorite:

- favorite_id
- user_id
- product_id
- saved_at

6. Report:

- report_id
- reporter_id
- target_type (user, listing)
- target_id
- reason
- Status (pending, reviewed, resolved)
- created_at

7. Search History:

- search_id
- user_id
- filters_used
- timestamp

8. Time Limit:

- time_limit_id
- product_id
- start_time
- end_time
- status (active, expired)

4. Functional Requirements

FR1. User Registration & Affiliation Check

Users must register using a verified university email (e.g., hs-fulda.de) to ensure access is limited to Fulda/SFSU affiliates.

FR2. User Profile Management

Users should be able to view, edit, and update their profile details, including name, photo, and short bio.

FR3. Post New Item/Service

Authenticated users can create listings for products or services by entering a title, description, price, and other details.

FR4. Upload Media for Listing

Users can upload multiple images (or a video) for each listing to showcase their item or service.

FR5. Categorize Listings

Users can select categories and tags while posting, to help others find listings more easily.

FR6. Browse and Search Listings

Users can search listings by keyword or browse through categorized sections.

FR7. Filter and Sort Listings

Users can filter search results by category, price range, tags, etc., and sort by newest, oldest, or price.

FR8. Internal Messaging

Buyers and sellers can communicate through an in-platform messaging system without sharing personal emails.

FR9. Add to Watchlist

Users can save listings to a personal watchlist to revisit them later.

FR10. Seller Dashboard

Sellers can view and manage their listings, including edit/delete actions and listing status (e.g., active, expired, sold).

FR11. Admin Approval for Listings

All posted listings must be reviewed and approved by an admin before becoming publicly visible.

FR12. Admin Moderation Capabilities

Admins can remove inappropriate listings or ban users when necessary.

FR13. Report Inappropriate Listings

Users can flag listings or messages as inappropriate to alert admins for review.

FR14. Flash Sale Listing

Allow sellers to mark a listing as a "Flash Sale," highlighting it prominently for a limited time to quickly attract buyers—ideal for urgent situations like moving out.

FR15. Automatic Listing Expiry

Listings will automatically expire after a set period (e.g., 30 days), with an option to renew.

FR16. Share Listings Externally

Users can share listing URLs via social media, email, or other messaging platforms.

FR17. Accept Terms and Privacy Policy

Users must accept the site's Terms of Service and Privacy Policy during registration.

FR18. Track Price History

Each listing will show a basic price change history so users can see if the price has been reduced over time.

FR19. Highlight Discounted Listings

Display listings whose prices have dropped from their original value in a special "Price Drops" section to catch buyer attention.

FR20. Temporarily Hide Listing

Sellers can temporarily pause a listing from public view without deleting it—useful for restocking or updating.

FR21. Password Reset Flow

Enable users to reset forgotten passwords and set new ones to regain access to their account.

FR22. Contact/Support Form

A built-in "Contact Us" form for users to submit questions or issues, routing entries to a monitored support inbox.

FR23. FAQ/Help Page

Provide a static FAQ section covering common questions (e.g., how to post, how to flag content) so users can self-serve basic support.

FR24. Bulk Message Deletion

In the messaging area, let users select and delete multiple old conversations at once for inbox cleanup.

5. Non-Functional Requirements

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in Milestone 0. Application delivery shall be from chosen cloud server
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. All or selected application functions must render well on mobile devices
4. Data shall be stored in the database on the team's deployment cloud server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English (no localization needed)
8. Application shall be very easy to use and intuitive
9. Application should follow established architecture patterns
10. Application code and its repository shall be easy to inspect and maintain
11. Google analytics shall be used (optional for Fulda teams)
12. No email clients shall be allowed.
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
15. Application shall be media rich (images, video etc.). Media formats shall be standard as used in the market today

16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
17. For code development and management, as well as documentation like formal milestones required in the class, each team shall use their own GitHub to be set-up by class instructors and started by each team during Milestone 0
18. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "Fulda University of Applied Sciences Software Engineering Project, Summer 2025 For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive Analysis

Feature	eBay Kleinanzeigen	Facebook Marketplace	Shpock	Swoplet
Account creation & profile	✅	✅	✅	✅
Post/edit/delete listings	✅	✅	✅	✅
Advanced search & filter	Basic filters	Moderate	✅	Custom filters
In-site messaging	✅	✅	✅	✅
Admin approval for listings	✅	✅	✅	✅
Favorites/bookmarked listings	✅	✅	✅	✅
Time-limited listings	❌	❌	Premium	✅
Report users/listings	✅	✅	✅	✅
Role-based access control	✅	✅	✅	✅

University-only access	❌	❌	❌	✅ (hs-fulda.de)
Compare	❌	❌	❌	✅

7. High-Level System Architecture

7.1 Architecture Overview

Our application follows a three-tier architecture consisting of the Presentation Layer (Frontend), Application Layer (Backend), and Data Layer (Database), all hosted on a cloud-based infrastructure.

Frontend (Presentation Layer)

- Developed using React.js for a dynamic and responsive Single Page Application (SPA).
- Styled using Tailwind CSS, enabling fast and consistent UI design.
- Uses RESTful for communicating with the backend API.
- Hosted behind NGINX, which serves the static frontend files and proxies API requests to the backend.

Backend (Application Layer)

- Built with Node.js and Express.js, providing a lightweight and efficient REST API.
- Handles all core logic such as user authentication, listing management, messaging, and admin approvals.
- Implements middleware for validation, role-based access control, and error handling.
- Ensures secure communication using SSL/TLS encryption.

Database (Data Layer)

- *Uses MySQL 8.0 as the relational database.*
- *Stores structured data related to users, listings, messages, reviews, and admin records.*

- *Connected securely via internal networking and accessed only from within the cloud server.*

Web Server & Hosting

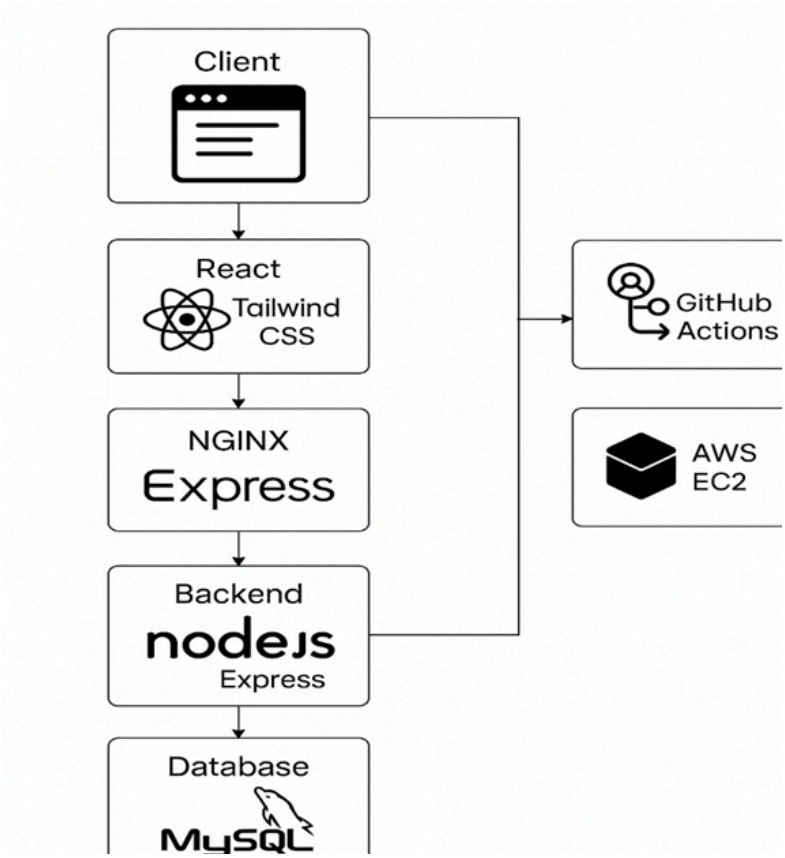
- NGINX is configured as a reverse proxy, serving frontend files and forwarding API traffic.
- The entire application stack is hosted on AWS EC2 (t2.micro) running Ubuntu 22.04 LTS.

CI/CD & Deployment

- GitHub Actions is used for Continuous Integration and Deployment.
- Code pushed to the main branch triggers an automated build and deployment to AWS.
- All environment variables and credentials are securely managed through AWS and GitHub secrets.

Security Measures

- HTTPS via Let's Encrypt.
- Input validation and sanitization to prevent SQL injection and XSS attacks.
- Admin approval system for listings and media moderation.



System Architecture

8. Team and Roles

Team Members	Role
Hamza Butt	Team Lead
Karan Patel	Github Master
Pooja Puthu Vayal	Backend Lead
Akhil Sajan	Frontend Lead
Hammad Asif	Frontend Developer

9. Checklist

1. The team found a time slot to meet (online) outside of the class. - DONE
2. GitHub master chosen. - DONE
3. The team decided and agreed together on using the listed SW tools and deployment
4. server. - DONE
5. Team ready and able to use the chosen back and front-end frameworks and those
6. who need to learn are working on learning and practicing. - ON TRACK
7. The team lead ensured that all team members read the final M1 and
8. agreed/understood it before submission. - DONE
9. GitHub is organized as discussed in class. - DONE