Master Team Project Fall 2024

NeuAnfang

Team 3

Member Name	Role
Zubeena Shireen Sheikh zubeena-shireen.sheikh@informatik.hs-fulda.de	Team Lead and Document Master
Muhammad Zaid Akhter	Backend Lead
Masood Ahmed Mohiuddin	Frontend Lead
Humdaan Syed	GitHub Master
Andrii Kuripka	Team Member (Frontend)

Milestone 1

07/11/2024

Date Submitted	07/11/2024
Date Revised	

1. Executive Summary: (Shireen)

NeuAnfang was created to address a pressing challenge faced by Fulda University students: finding reliable, convenient, and affordable accommodation when moving to the city. For students relocating to pursue their education, navigating a new rental market can be daunting and stressful. NeuAnfang simplifies this journey by combining transparency and trust with housing search. With a focus on the unique needs of students, NeuAnfang facilitates a smoother transition, allowing them to spend less time worrying about finding a home and more time enjoying their new chapter in Fulda.

NeuAnfang serves as a one-stop platform where landlords can post property ads and students can search for listings tailored to their individual needs. They can also view estimated average rent in the area which will help them plan their budget. Plus, the number of interested applicants shown on each listing gives students insight into the popularity of various options. Students can even bookmark listings they are interested in, and landlords can update the listing status to keep students informed, reducing any uncertainty or waiting.

NeuAnfang prioritizes transparency and trust by enabling direct communication between students and landlords through the platform. Unique features like user ratings on listings and the ability to report issues, create a safe and reliable environment for all users. In addition, a dedicated Moderator ensures only trustworthy information remains on the platform. NeuAnfang also provides helpful guidance on the rental process, legal documentation, tips for avoiding scams and many more, making it an invaluable resource for students new to renting.

Our platform intends to support students throughout their entire housing journey, from the initial search to securing a home. With the comprehensive search feature, an intuitive recommendation engine that matches students with potential landlords, and in-app reminders for appointments, it acts as a true wingman for students. For landlords, the platform offers valuable analytics for their ads to help them optimize listings. This dual focus on both students and landlords sets our platform apart, providing value for all users and ensuring they always have something to look forward to.

Our team consists of five enthusiastic students from Fulda University and each of us have experienced the challenges of finding housing here firsthand. We've drawn from these experiences to design a platform that truly addresses the needs of our peers. Together, we've crafted a solution that combines useful features with thoughtful design and functionality. With our strengths in backend development and a keen eye for aesthetics, we're confident NeuAnfang will be the go-to housing solution for students looking for a smooth and informed rental experience.

2. Personae and Main Use cases:

Following are the main personas for the application:

1. Student
Zaid is a full-time student at Fulda University of Applied Sciences. He recently moved from Pakistan to pursue his Master's in Global Software Development. He is also working a part-time job hence he has very limited time. Also, he is new in Germany and he has very little German skills. He is looking for an apartment near the university under a budget of EUR 450 per month.

Pain points:
- Finding affordable housing,
- Communicating with landlords,
- Understanding rental terms

2. Landlord
John is a property owner in Fulda. He recently renovated his apartment complex on Leipziger Strasse, right opposite to the university. He has 3 available apartments and wants to rent them out to students. He is looking for reliable tenants. He wants to be able to efficiently manage inquiries, screen tenants easily and co-ordinate viewings with potential tenants.

Pain points:
- Managing inquiries,
- Screening tenants,
- Coordinating viewings

3. Site Admin
Masood is a member of the startup team. He studies Master's in Global Software Development at Fulda University of Applied Sciences. He has 4 years of discord moderation experience. He wants to be able to easily manage all the site content, ensure site-reliability and ensure the content uploaded on the site is appropriate.

Pain points:
- Content review workload,
- Maintaining site quality,
- Handling disputes

Based on the above personas, following are the main use cases of the application:

1. UC-1: List Apartments

John has 3 available apartments and wants to rent them out to students. He visits NeuAnfang to create a new apartment listing. He uploads photos and provides details like rent, location, and amenities. He specifies requirements for potential tenants and waits for administrator approval before the listing goes live. Once the listing is approved John's apartment ad will be online for students to view.

2. UC-2: Search Apartments

Zaid is looking for affordable housing in Fulda. He visits NeuAnfang and sees a list of properties on the home page. He applies search filters to find a suitable apartment. He applies filters to find an apartment near the university. He also wants a fitted kitchen and washing machine. His budget is max EUR 450. After clicking search he sees a list of apartments that match his needs. He can view quick info like postcode, monthly rent, title of the ad from the list view. He can also switch to the map view to see the apartments on a map.

3. UC-3: View Apartment Details

Once Zaid finds an ad he likes. He clicks on it to view the full details of the apartment. He can browse through photos, read the description uploaded by the landlord and see the available amenities. He can also see the exact location of the apartment on a map.

4. UC-4: Real-time Chat

Zaid likes the apartment he is viewing; he initiates contact with the landlord through the platform's messaging system. They can discuss viewing arrangements, ask questions about the property, and share basic contact information for further communication, all while maintaining privacy through the platform.

5. UC-5: Moderate Content

Masood reviews new listings in his queue, checking for compliance with platform guidelines, appropriate content, and accurate information. He can approve, reject, or request modifications to listings. He also monitors user interactions and responds to reported content or user complaints.

3. List of main data items and entities:

User is a generic term used for referring students and landlords together. The application has the following user types:

A **Student** represents a verified user with an hs-fulda.de email domain. They form the primary user base of the platform, with capabilities to search listings, maintain a wishlist of preferred apartments, communicate with landlords, and save their search preferences. Students can report inappropriate content and must be able to demonstrate their affiliation with Fulda University.

A **Landlord** is a verified user who owns or manages rental properties. They can create and manage apartment listings, respond to student inquiries, and track the status of their properties. Landlords must provide valid contact information and maintain their property listings, including updating availability and responding to offers from potential tenants.

The **Site Admin** holds system-level privileges and is responsible for platform integrity. They review and approve new listings, moderate user content, manage user accounts, and handle disputes. Admins also monitor platform activity and can generate system reports for operational insights.

Some other main data items in the application are defined as follows:

An **Ad/Apartment** is the central entity of the platform, representing a rental property listing. It contains comprehensive property information including location, price, amenities, and media content. Each listing has a unique identifier and maintains various states from creation through to rental or archival.

The platform offers two primary viewing modes: **List View** provides a structured, scannable format showing apartment summaries with key details and quick actions, while **Map View** offers a geographical representation of available properties, featuring interactive markers and distance-based searching from campus.

Search Criteria encompasses the filtering and sorting parameters users can apply to find suitable apartments. This includes price ranges, location preferences, property features, and distance from campus. Users can save their search profiles for future use.

The **Chat** system facilitates direct communication between students and landlords. Each conversation is tied to a specific property listing and maintains a complete message history. The system supports basic attachment sharing while adhering to platform security guidelines.

An **Offer** represents a formal expression of interest from a student to a landlord regarding a specific property. It includes proposed terms and maintains various states from initial submission through to acceptance or rejection.

The **Wishlist** feature allows students to save and organize their favorite listings. Users can add personal notes and set alert preferences for saved properties.

Notifications keep users informed of relevant platform activity. These can be system alerts, chat messages, or listing updates. Each notification has a priority level and expiration date where applicable.

A **Session** tracks user activity on the platform, maintaining security and authentication state. It includes login information, device details, and activity timestamps to ensure secure access to the platform.

The platform also maintains several status categories to track the state of various entities:

Listing Status:
Listings can be Pending (awaiting approval), Active (available for rent), Rented (transaction completed), Archived (no longer available), or Rejected (failed approval).

User Status:
User accounts may be Pending Verification, Active, Suspended, or Deactivated, reflecting the user's standing on the platform.

Chat Status:
Chat conversations can be Active, Archived, Blocked, or Reported, managing the flow of communication between users.

Offer Status:
Offers progress through states including Pending, Accepted, Rejected, Expired, or Withdrawn, tracking the rental negotiation process.

4. Initial list of functional requirements: (Masood + Andrii + Humdaan)

1. Apartment Posting and Pricing:
   Landlords can post apartment listings, providing essential details to attract potential student tenants. The system suggests a rental price based on the average cost per square meter, helping landlords set competitive rates.

2. Share listings:
   Landlords can share apartment listings on other social platforms, increasing visibility and reach for the property.

3. Ad Status Management:
   Landlords can indicate the availability of their property to students by mark listings as reserved, archived, or deleted.

4. Student Profiles and Document Upload:
   Students can create profiles and upload relevant documents, such as identification and proof of enrollment, to support their rental applications.

5. List View of Apartments:
   Students can browse all available apartments in a list format, making it easier to compare multiple properties at once.

6. Map View of Apartments:
   The exact location of the apartment can be viewed on an interactive map for a geographic overview and easy navigation, helping students understand its proximity to key areas.

7. Save Search Criteria:
   Students can save custom search criteria to streamline future searches with the same parameters.

8. Detailed Apartment View:
   Students can click on an apartment listing to see detailed information, such as amenities, photos, floor plans, and rental terms.

9. Commute Information:
   Students can view commute times to the university and nearby bus routes for better transportation planning.

10. Wishlist Apartments:
    Students can add apartments to a wishlist and access a dedicated section with all their wishlisted apartments for easy reference.

11. Direct Messaging with Landlord:
    Students can message landlords to inquire about listings, with student profile documents shared automatically with the landlord after approval by the student.

12. Real-Time Chat:
    The platform supports real-time chat functionality, allowing students and landlords to communicate instantly.

13. Chat File Sharing:
    Users can upload images and documents within chat messages, streamlining information exchange.

14. Block Users in Chat:
    Users can block others on chat if necessary, helping maintain privacy and security.

15. Offer Apartment in Chat:
    Landlords can make a direct rental offer to students they are communicating with in the chat.

16. Retract Apartment Offer:
    Landlords have the option to retract an offer before it is accepted or rejected by the student.

17. Accept/Reject Offer and Rate Landlord:
    Students can accept or decline a rental offer from a landlord, with the option to rate the landlord post-decision.

18. View Landlord Profile:
    Students can view a landlord's profile, which includes all apartment listings they have posted on the platform.

19. Follow Landlords:
    Students can follow specific landlords to receive updates on new postings and maintain easy access to their listings.

20. New Listing Notifications:
    Students who follow landlords receive notifications when a new apartment is posted by that landlord, ensuring timely updates.

21. Ad Approval by Admin:
    The site administrator reviews and approves each apartment listing before it becomes visible to students, ensuring that inappropriate/fraudulent ads are not posted.

22. Landlord Dashboard with Analytics:
    Landlords can access a dashboard that provides insights into the performance of their listings (e.g., number of views, inquiries per listing) to help gauge interest and optimize listings.

23. Customizable Notifications:
   Both students and landlords can customize their notification settings (e.g., get notified about new listings in preferred areas, updates on inquiries).

24. Advanced Filtering Options:
   Allow students to filter listings by specific amenities (e.g., furnished, utilities included) for a more tailored search experience.

25. In-App Viewing Scheduler:
   Include a shared calendar feature where landlords and students can schedule viewings directly within the app, with reminders sent out to both parties.

26. Message Templates for Landlords:
   Landlords can create and reuse message templates for common inquiries, such as availability or terms, to save time in responding.

27. Neighborhood Information Section:
   Display information about the neighborhood, such as nearby amenities (e.g., grocery stores, parks), and popular spots for students.

5. List of 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. Full resolution free media shall be downloadable directly, and full resolution media for selling shall be obtained after contacting the seller/owner
6. No more than 50 concurrent users shall be accessing the application at any time
7. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
8. The language used shall be English (no localization needed)
9. Application shall be very easy to use and intuitive
10. Application should follow established architecture patterns

11. Application code and its repository shall be easy to inspect and maintain
12. Google analytics shall be used (optional for Fulda teams)
13. No email clients shall be allowed.
14. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
15. Site security: basic best practices shall be applied (as covered in the class) for main data items
16. Application shall be media rich (images, video etc.). Media formats shall be standard as used in the market today
17. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
18. 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
19. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "Fulda University of Applied Sciences Software Engineering Project, Fall 2024 For Demonstration Only" at the top of the WWW page. (Important to not confuse this with a real application).

6. Competitive analysis: (Masood + Andrii)

In Germany, students often face challenges in securing affordable and conveniently located housing. A range of platforms exists to address this need, each offering unique functionalities. Leading platforms include WG-Gesucht, ImmobilienScout24, Studierendenwerk, and Ebay Kleinanzeigen. These services cater to different aspects of the rental market, such as student-specific listings, shared housing options, and traditional apartment rentals. Our platform aims to combine the strengths of these competitors while offering enhanced features for student users, including tailored search filters, in-app communication, and streamlined rental application processes.

Comparison Table

Feature	WG-Gesucht	ImmobilienScout24	Studierendenwerk	Ebay Kleinanzeigen	Our Platform
Primary Focus	Shared housing, student rentals	General rentals	Student dormitories	General classified listings	Student-focused rentals

Student Verification	No	No	Yes	No	Yes
In-app Messaging	Yes	No	Limited?	No	Yes (real-time chat)
Map View	Yes	Yes	Yes	Limited	Yes (with commute times)
Rental Price Suggestions	No	Yes	No	No	Yes
Property Management Tools	Limited	Yes	Yes	No	Yes (dashboard for landlords)
Saved Searches & Notifications	Yes	Yes	No	Limited?	Yes(Customizable)
User Ratings and Reviews	No	No	No	No	Yes (landlord & student reviews)
Profile Customizatio n	Limited	Limited	No	No	Yes (for students and landlords)
Application Tracking	No	No	Yes (for student housing)	No	Yes
Availability for International Users	Yes	Yes	Limited to German students ?	Yes	Yes (auto-translated listings)

Exploring Floor Plans	No	Yes	No	No	Yes

Our platform stands out by focusing specifically on students' needs, with features such as verified student profiles, in-app communication, and an easy-to-navigate dashboard for both students and landlords. Additionally, advanced features like commute time visualization, customizable profiles, and a streamlined application process make it more convenient and transparent for users, helping them find housing that best fits their lifestyle and academic requirements.

New content

Our platform stands out among competitors like WG-Gesucht, ImmobilienScout24, Studierendenwerk, and Kleinanzeigen by focusing specifically on student rentals and offering unique features tailored to the needs of student tenants and landlords. Here are the key advantages of our platform:

1. **Student-Focused Rentals**: Unlike general rental or classified platforms, our platform is exclusively student-focused, ensuring listings and services cater directly to students' needs.
2. **Enhanced Verification and Safety:** Our platform includes student verification, which adds an extra layer of trust and security that general platforms lack, making it safer for both students and landlords.
3. **Real-Time In-App Messaging:** We provide real-time chat for easy communication, surpassing platforms like ImmobilienScout24 and Studierendenwerk that lack this feature, and going beyond the basic messaging systems on others.
4. **Commute Times and Map View:** Our map feature includes commute times to university, which is especially useful for students balancing housing location with academic schedules—an enhancement over standard map views on other platforms.
5. **Rental Price Recommendations:** We offer rental price suggestions, providing students and landlords with fair pricing insights, encouraging them to rely on this suggestion to avoid unreasonable expectations or overpricing. This feature is designed to promote fair pricing for students, a service not commonly available on other platforms.
6. **Comprehensive Landlord Dashboard:** Our platform supports landlords with a dedicated dashboard for managing properties, a feature that most competitors lack.
7. **User Ratings and Reviews:** By incorporating reviews for both students and landlords, we help foster transparency and trust, unlike most other platforms that do not offer user feedback systems.
8. **Profile Customization:** Both students and landlords can customize their profiles, allowing for a more personalized experience that sets our platform apart from others with limited or no customization options.

In summary, our platform combines student-centric features with advanced functionalities, providing a safer, more user-friendly experience tailored specifically for the student rental market. This specialization and range of features make it a superior choice for both students and landlords compared to broader rental or classified platforms.

FEATURES	OUR PLATFORM	WG-GESUCHT	IMMOBILIEN-SCOUT24	KLEIN ANZEIGEN	STUDIERENDEN WERK
EXCLUSIVELY FOR STUDENTS	✓	✗	✗	✗	✓
IN-APP MESSAGING	✓	✓	✗	✓	✗
USER RATING/REVIEWS	✓	✗	✗	✗	✗
RENTAL PRICE SUGGESTIONS	✓	✗	✗	✗	✗
TIME TO UNI ROUTES	✓	✗	✗	✗	✗
IN-APP CALENDAR	✓	✗	✗	✗	✗
ALL APPARTMENTS MAP VIEW	✓	✗	✗	✗	✗
PROFILE CUSTOMIZATION	✓	✗	✗	✗	✗

7. High-level system architecture and technologies used: (Humdaan + Zaid)

Following are the main technologies that will be used to develop the application:

Server Host: AWS EC2 t2.micro 1 vCPU 1 GiB RAM
Operating System: Ubuntu 22.04 Server
Database: MySQL 8.0 on AWS RDS
Web Server: NGINX 1.12.2
SSL Cert: Let's Encrypt (CertBot)
Server-Side Language: NodeJS
Additional Backend Technologies: ExpressJS, socket.io
Frontend Framework: ReactJS
Additional Front End Technologies: Material UI

Now moving on to system architecture. The application will use a three-tier web application design. Let's break down each tier:

1. Client Tier:
- Features three types of users: Students (who view ads), Landlords (who post ads), and Admins (who moderate content)
- Frontend is built using React.js
- Users interact with a web interface for various operations

2. Server Tier
- Running on Ubuntu VM in AWS EC2
- NGINX Web Server
    - Running on ports 443 (HTTPS) and 80 (HTTP)
    - Acts as a reverse proxy to Node.js server on localhost:3000
- Node.js Server
    - Running on port 3000
    - Handles API requests and business logic
    - Communicates with storage and database tiers

3. Storage & Data Tier:
- AWS S3
    - Used for storage bucket
    - Handles direct upload/download with signed URLs

- AWS RDS
  - Running MySQL database
  - Handles SQL queries and data persistence

Key Integration Points:
- HTTPS/REST API requests flow from React App through NGINX to Node.js
- AWS SDK is used for object operations with S3
- Direct file uploads/downloads use signed URLs for S3 access
- Database queries are executed from Node.js to MySQL

The architecture follows modern security practices with HTTPS and proper separation of concerns across tiers. It's scalable and uses industry-standard AWS services for storage and database operations. This architecture is better described by the diagram below:
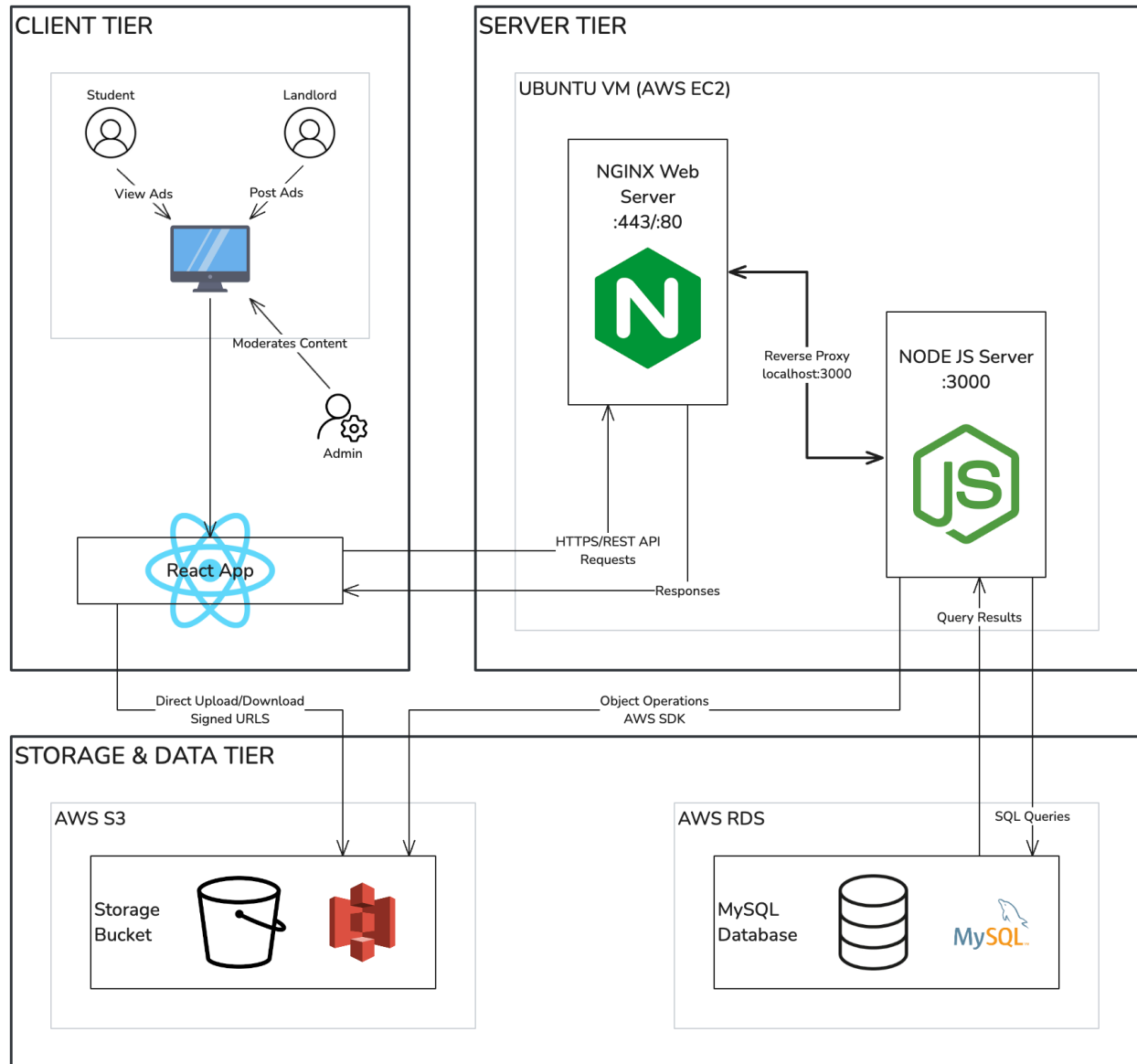
8. Team and roles:

Member Name	Role
Zubeena Shireen Sheikh	Team Lead and Document Master
Muhammad Zaid Akhter	Backend Lead
Masood Ahmed Mohiuddin	Frontend Lead
Humdaan Syed	GitHub Master

Andrii Kuripka	Team Member (Frontend)

9. Checklist

Task	Status
Team found a time slot to meet (online) outside of the class	**DONE**
GitHub master chosen	**DONE**
Team decided and agreed together on using the listed SW tools and deployment server	**DONE**
Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	**ON TRACK**
Team lead ensured that all team members read the final M1 and agree/understand it before submission	**DONE**
GitHub organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	**ON TRACK**