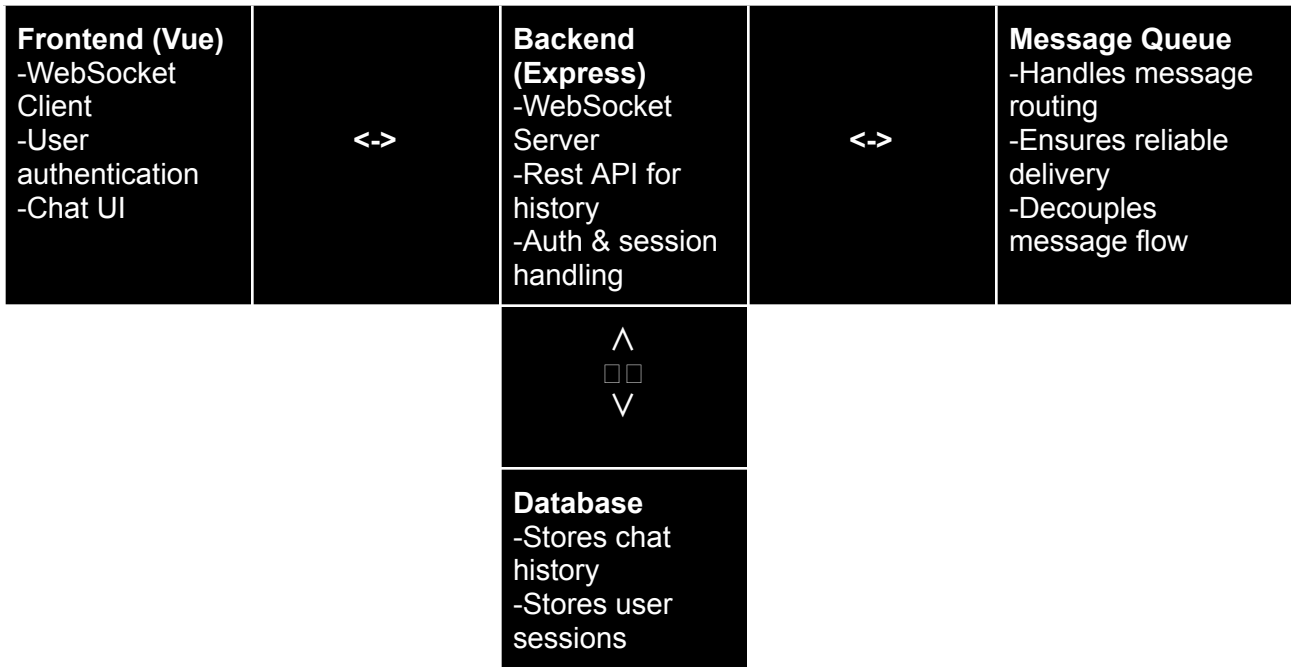
Hypervisor Enhanced Logistics Program (H.E.L.P)

Granjeon, Lucas
Ayub, Umer
Lebo, Drew

Chapter 1

Many businesses require a scalable and efficient help desk solution for customer support. The project covers critical topics such as WebSockets, message queues, and distributed deployment. Using a message queue allows for handling high volumes of messages efficiently. This project aims to develop a real-time chat application designed as a helpdesk platform. It will feature a responsive frontend using Vue.js, a backend powered by Express.js with WebSocket support, and a message queue system to ensure reliable communication.



The Frontend uses Vue.js. It runs in a Docker container with an Nginx web server to serve the application. Communication with the backend via Websockets for real-time chat.
For the Backend, we will be using Express.js. Handling user authentication, session management, and Websocket connections. REST API will be used for retrieving data from past messages. This will all be run inside a Docker container, making it easy to deploy and scale.
The message Queue will use RabbitMQ/Kafka. It will ensure scalable and reliable message handling as well as allow easier communication between frontend-backend. This will be run on a separate Docker container.
Lastly we will run a database like MongoDB to store chat history, user sessions, and authentication data. This will run inside its own container to ensure that data is persistent and secure.

Chapter 2

System Architecture

Utilizing a Vue.js frontend that runs inside of a Docker container running Nginx, we can utilize REST API calls to retrieve historical data while providing real-time chat updates via WebSockets. With an Express.js backend we can handle user authentication, session management, websocket connections, and the RESTful API all inside of another docker container. Running a message queue in a dedicated container allows for users to asynchronously message and it ensures messages are to be delivered reliably. Finally, running a database in a dedicated container to store the required information allows for secure and reliable data handling.

Database Design

Utilizing MongoDB, a document-based structure, having efficient collections is necessary. The primary focus is to have indexes for quick lookups and securely stored data for efficiency.

Users Collection

The users table should contain a handful of information, such as the table below. The biggest concern about the table is a security concern, which is why storing the password as a hash is necessary. Utilizing **bcrypt**, a securely stored password can be achieved.

Another concern is quick lookups, which is why the username and email indexes should be utilized.

_id	**username**	**email**	**password_hash**	**role**	**created_at**	**updated_at**	**status**
ObjectId	string	string	string	string	ISODate	ISODate	string

- **Role** can be admin, support_agent, or customer
- **Status** can be active, inactive, or banned

Sessions Collection

This collection stores all the information for user session tokens. Quick lookup IDs are included for the user's id and the session token. The session token should be signed and stored securely

_id	user_id	session_token	expires_at
ObjectId	ObjectId	string	ISODate

Messages Collection

This collection is a crucial security point in the application. The primary concern of this table is input sanitization to prevent attacks via NoSQL Injection. For quick lookups we have the sender and receiver ids and the timestamp.

_id	sender_id	receiver_id	content	timestamp	status
ObjectId	ObjectId	ObjectId	string	ISODate	string

- The status string can be sent, delivered, or read

Tickets Collection

Like the messages collection, this collection is also a crucial part. Depending on the volume of data, it might be important to consider **message sharding**, but for the time being the following table should be sufficient. The quick lookup indexes are the customer id, the agent id, and the status.

_id	customer_id	agent_id	status	priority	messages	created_at	updated_at

ObjectId	ObjectId	ObjectId	string	string	[
					{
					“sender_id” : “ObjectId”,
					“content” : “string”,
					“timestamp” : “ISODate”
					}
					]
						ISODate	ISODate

- **Status** string can be open, pending, resolved, or closed
- **Priority** string can be low, medium, high, or urgent

Security Considerations

Authentication and Authorization

To handle the login, user roles, and session tokens with optimal security, a few technologies will need to be implemented.

- JWT-based authentication
    - Provides a stateless login
- Role-based access control (RBAC)
    - Restricts access
- Session Expiration Policy
    - No user should be capable of being signed in forever.

Data Protection

To handle data security, the following technologies are crucial.

- Password encryption
    - Using bcrypt
- Sanitizing data input
    - Prevents NoSQL injection

For future reference, using HTTPS for all communications and Rate-limiting are also ways to improve security.

WebSockets Security

The primary importance of security for WebSockets is connection authentication. Without a WebSocket connection, the session should not be established. The WebSocket should also ensure message size limits and message format validation are implemented to prevent abuse.

Message Queue

Considering the unsureness of which technology to use between Kafka and RabbitMQ, here's a breakdown comparing the two.

Feature	**RabbitMQ**	**Kafka**
Message Model	Queue-based (push)	Pub/Sub (pull-based)
Message Order	Guaranteed per queue	Ordered within partitions
Persistence	Durable Queues (if enabled)	Logs messages by default
Scalability	Good for small workloads	Better for high throughput
Use Case	Real-time chat, event-driven	Large-scale streaming

Given the comparisons, it seems RabbitMQ is better suited for the needs of this project. Since the project is real-time chat and event-driven, RabbitMQ makes more sense, especially since it seems to be easier to set up.

An example workflow for the message queue could look like: Frontend sends message via WebSocket -> Backend publishes message to a RabbitMQ exchange -> Consumer service (express.js application) pulls message from the queue -> Message is stored in MongoDB and forwarded to the recipient.

Deployment Strategy

The frontend (Vue) and the backend (Express.js + WebSockets + REST API), RabbitMQ, and MongoDB, each run inside of their own Docker container.

Chapter 3

Utilizing Docker to containerize our application is the key focus for this project. By containerizing, we can provide a layer of abstraction between the frontend, backend, message queue, and database.

Infrastructure

We use Docker for isolation of components, while utilizing Docker Compose to streamline the deployment process while also keeping our containers together in the same network and cluster. We mount volumes for our database, our frontend, and our backend for persistent data and also rapid deployment.

Deployment

We implemented executable files that are compatible with Mac/Linux and Windows. These executables clear the previously built images and then run Docker Compose.

A problem that arose was with the line endings from writing the scripts on Windows, which resulted in issues on deploying to Mac and Linux machines.

LUCAS GRANJEON

CERTIFICATIONS

Occupational Skills Award for Information Systems
December 2022

Occupational Skills Award for Infomation Security
December 2022

Google Data Analytics Certificate (ongoing)

SKILLS

Hard:
Java - Python - C++ - VS Code - Github

Soft:
Communication - Problem-Solving - Motivation

LANGUAGE

French: Native
English: Fluent
Spanish: Conversational

HOBBIES

Fly Fisher
Car enthousiast
PC builder

Computer Science Student at West Chester University

@ granjeonl@gmail.com / Linkedin: Lucas Granjeon / West Chester, PA

SUMMARY

Motivated Computer Science student at West Chester University with a passion for Artificial Intelligence. I have a solid foundation in programming, algorithms, and data structures. Proficient in languages such as Python, Java, and C++, with hands-on experience in software development, problem-solving, and teamwork. I am eager to apply my academic knowledge to real-world challenges.

PROJECTS

CSC240 - Text Processing Project

-Developped a text processing tool in Java to analyze and extract key information from large text datasets.
-Designed the tool to efficiently handle diverse text formats, optimizing performance for large-scale datasets.
-Implemented document summarization and content-based categorization, reducing time required for text analysis.
-Created a user-friendly interface to navigate through the data gathered.

CSC301 - AI in Cybersecurity (Research Paper and Presentation)

-Researched and analized the role of AI and machine learning techniques in improving cybersecurity measures
-Developed a research paper exploring the use of AI for threat detection, anomaly detection, and intrusion prevention in modern systems.
-Created a presentation summarizing the findings, focusing on practical applications and emerging trends in AI-driven cybersecurity
-Delivered the presentation to an academic audience, effectively communicating complex technical concepts and their real-world implications.

EDUCATION

Bachelor in Computer Science — 2020-2021
Moorpark Community College — Moorpark,Ca

Associate in Computer Information System — 2021-2023
McLennan Community College — Waco,Tx

Bachelor in Computer Science — 2023-Ongoing
West Chester University — West Chester,Pa

Drew Lebo

(They/He)

215-539-7122 | andrewlebo2@gmail.com | linkedin.com/in/andrewlebo | github.com/ALebo5193

Education

West Chester University of Pennsylvania — West Chester, PA
Bachelor of Science in Computer Science — *August 2021 – May 2025*

- **GPA**: 3.349
- **Relevant Courses**: Data Structures & Algorithms, Software Engineering, Software Security, Data Communications & Networking, Computer Security, Programming Language Concepts & Paradigms, Introduction to Cloud Computing, Modern Malware Analysis, Artificial Intelligence

Experience

Spray Park Attendant — May 2023 – August 2024
Bristol Township Parks and Recreation — *Bristol, PA*

- Maintained water chemistry by keeping ORP between 750-770 mV and the pH between 7.4-7.6 to keep in ordinance with the department of health.
- Followed a structured maintenance schedule for routine filter upkeep and strainer cleanings, enhancing the team's workflow efficiency and resulting in maintenance operations being completed more efficiently.
- Consistently delivered timely updates to coworkers and department leadership, ensuring effective communication and swift information flow.

Dining Room Busser — June 2020 – March 2021
King George II Inn and Tavern — *Bristol, PA*

- Independently managed all bussing operations until March 2021, when a new team member was trained and integrated.
- Championed the safe handling and disposal of 200+ pounds of kitchen waste weekly.
- Streamlined operations to reduce table wait times and improve customer service efficiency.

Projects

Skill Tree Web Application | *React, Bootstrap, Node.js, Express.js, MongoDB/Mongoose* — February 2025

- Develop a full-stack web application with a dynamic front-end
- Build a RESTful API backend to support the use of a React front-end.

Hug Nugs | *Godot, GDScript* — Fall 2024

- Collaborated with a team to create a platformer video game
- Developed the player script.
- Patched player interaction bugs in the project during merges.

Sons of Nug | *Godot, GDScript* — Fall 2024

- Collaborated with a team to create a multiplayer video game
- Developed the targeting system for the towers.
- Handled multiplayer synchronization between clients and server.

Portfolio Website | *CSS, HTML, Javascript* — June 2024

- Created a visually appealing website with a reactive user interface that adapts to screen sizes.
- Implemented a navigation header bar with contents that contain hover conditionals.

Technical Skills

Languages: Java, HTML/CSS, JavaScript, MongoDB
Frameworks: ReactJS, Node.js, RESTful API
Developer Tools: VSCode, Docker, Git, Cloudflare, Linux
Algorithms: Merge sort, Breadth-First Search, Depth-First Search, Uniform Cost Search

Umer Ayub

623 Saxony Drive, Fairless Hills, PA, 19030
267-981-9752 // ayubumer79@gmail.com // LinkedIn: umer-ayub

Summary

Computer science student at West Chester University of Pennsylvania. Proficient in Java, Python, and C, with experience in software development, data analysis, and machine learning applications. Developed projects including a book recommendation system, an email spam detection classifier, and command-line pattern generation using object-oriented principles. Experience as a Computer Tech at West Chester University, maintaining classroom technology and providing IT support while also having strong leadership and communication skills as a Shift Manager at 7-Eleven managing inventory, cash flow, and staff training.

Education

West Chester University, West Chester, Pennsylvania — Expected May 2026
Bachelor of Science, Computer Science
3.93 GPA
Relevant Courses: Data Structures and Algorithms, Computer Systems, Intro to Cloud Computing

University of Maryland, College Park, Maryland — Sep 2021 - 2023
Bachelor of Science, Undecided

Pennsbury High School, Fairless Hills, Pennsylvania — June 2021

Skills

Programming Languages: Java, Python, C
Software Packages: PyCharm, JDK Java Compiler, VSCode
Languages: Proficient in German, Hindi and Urdu

Projects

Book Recommendation System — Jan 2025
Personal Project
- Developed a book recommendation system using python which compiled multiple statistics in order to deliver personalized book recommendations towards the user
- Managed book data using JSON files and implemented scripts for recommendations, focusing on recommendation logic

Email Spam Detection Classifier — Dec 2024
Object Oriented Programming, West Chester University
- Developed a program to determine if an email is spam by the criteria of duplicate words, usage of URLs and word count using logistic regression
- Created multiple classes such as EmailDataset and Spam, using object-oriented principles for data-management and extensibility

Calculator June 2024

Personal Project

- Created a Java based calculator which is capable of performing addition, subtraction, multiplication and division
- Implemented object oriented principles as well as use of multiple classes
- Incorporated exception handling in cases of division by zero and invalid inputs

Command Line Pattern Art Feb 2024

Computer Science Principles, West Chester University

- Developed a program to output a specific pattern in terminal
- Utilized java abstraction and for loop logic to simplify the design

Experience

Computer Tech September 2024 - Present

College of Education and Social Work Tech Center, West Chester University

- Conducted maintenance and troubleshooting on classroom technology, ensuring smooth operation of devices such as projectors, computers, and other equipment
- Maintained inventory of thousands of technology items for the college and distributed by request to staff and students
- Assisted staff with technical issues, providing support for both software and hardware problems for both instructors and students

Shift Manager December 2020 - Present

7-Eleven

- Audited thousands of dollars' worth of stock to make sure everything was properly accounted for in the system
- Conducted daily cash reports in order to handle the thousands of dollars of daily sales
- Managed and trained multiple employees: ensuring proper use of the POS system and how to handle customer needs
- Increased sales from this location by 5.7% from the years 2021 to 2022