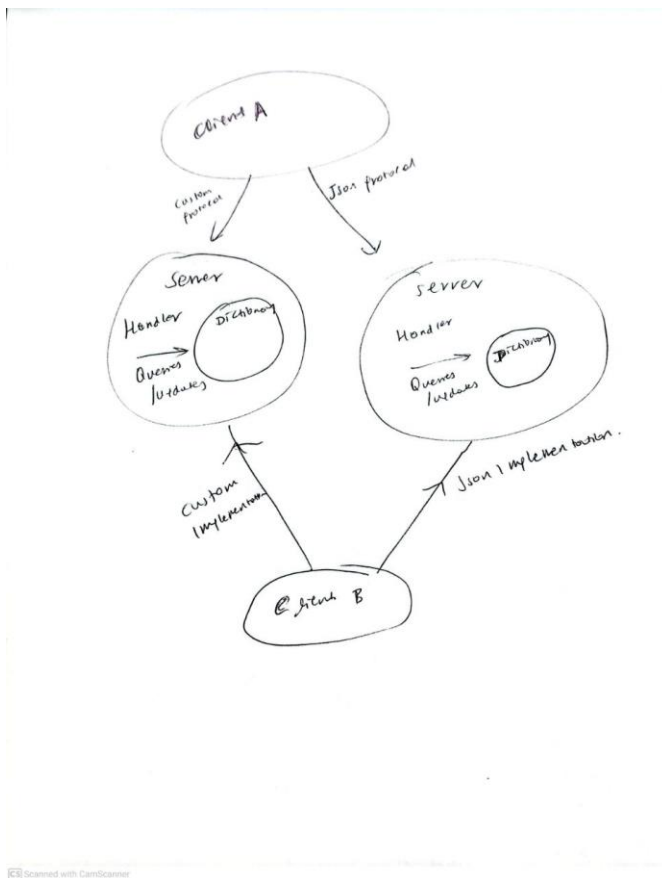


Chatbot Implementation Notebook

- Date/Time: 02/08/2025
 - Project Focus: Design Exercise 1 – Wire Protocols.
- Goals:
 - The high-level objective for the project as I start this project would be to first define the high-level design, project requirements, and design.
 - The high-level design for this project would be:



3. Recent Progress or Context

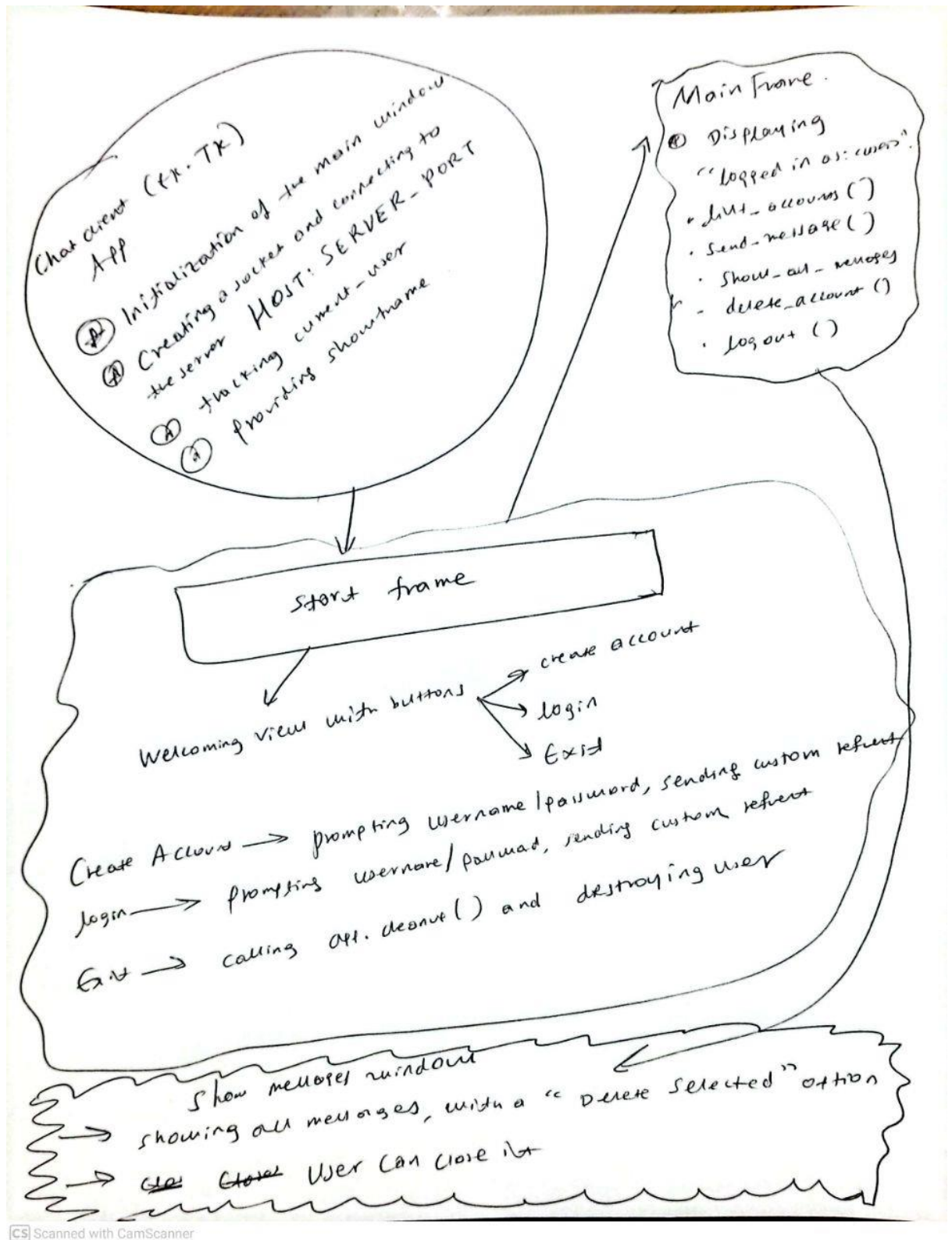
- Conversations after crafting the high-level design:
 - According to the high-level design the project should have two components, the server side and the client side. One person should focus on the client-side implementation and another person should focus on the server-side implementation. We will start with the custom

implementation and make modifications, keeping the main components, only when we need to. We will have files for logging and testing after the basic requirements of the projects are done. Here are the basic requirements of the project:

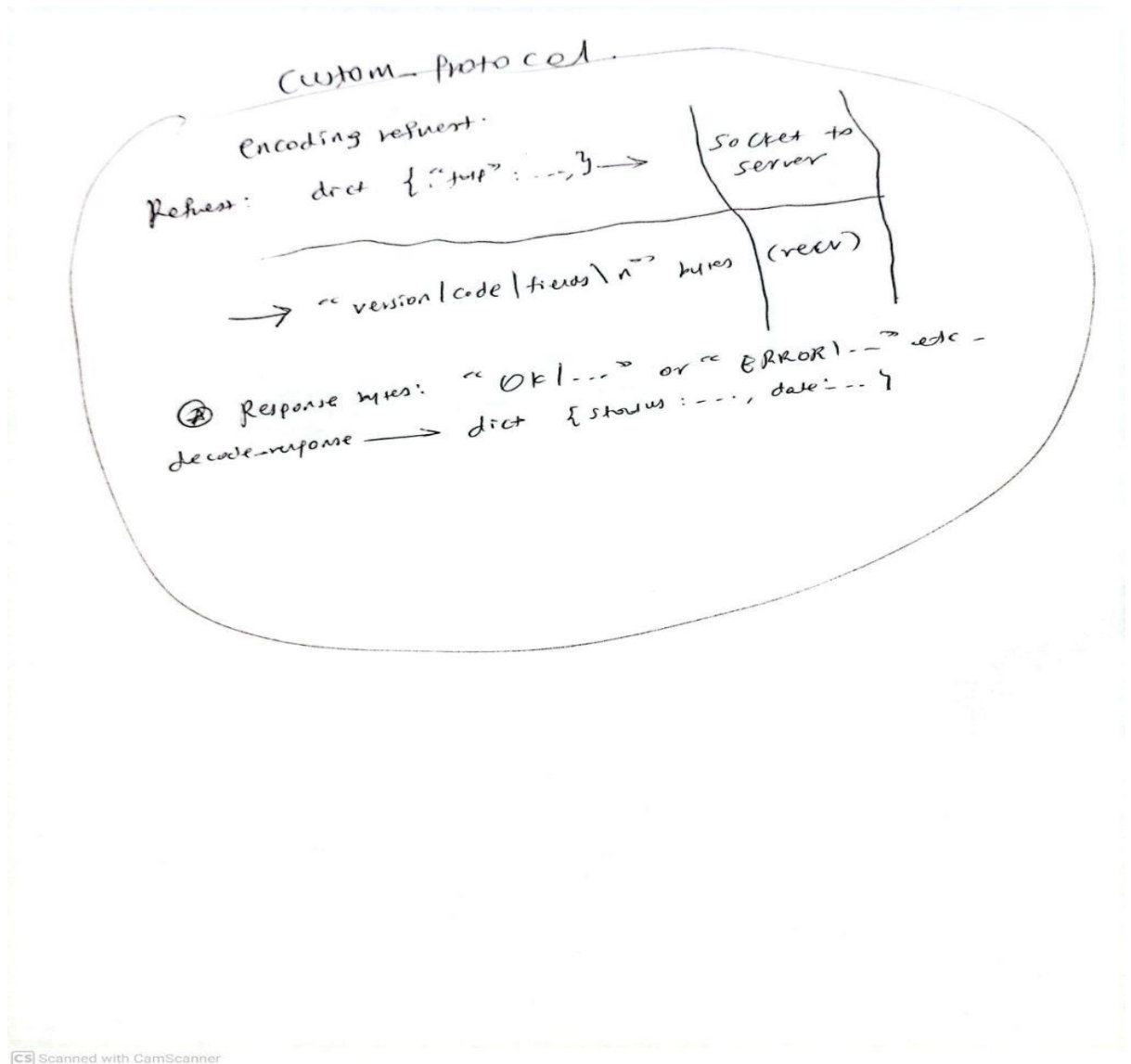
- Create Account: User supplies login name + password (or logs in if already exists).
- Secure Passwords: Never send/store in plaintext.
- Log In: Verify credentials; show number of unread messages on success.
- List Accounts: Possibly filter by pattern; handle pagination if many accounts.
- Send Message: Immediate delivery if recipient is online; store otherwise.
- Read Messages: Fetch unread messages; user can limit how many at once.
- Delete Message(s): Remove selected messages permanently.
- Delete Account: Specify policy for unread messages; remove account data.
- Two Protocols: Custom (efficient) + JSON (readable).
- Size Comparison: Measure data transfer for each protocol; note efficiency impact.
- Multiple Clients: Server must handle concurrent connections.
- Config: Client uses command-line or config file for server IP/port.
- GUI: Provide a user-friendly interface (could be minimal but clear).
- Demo & Code Review: Show working system and evaluate code coverage + documentation.
- Turn In: Submit code repo, engineering notebook, and review forms on due date.

4. Implementation Process(Detailed Design)

- Detailed design of windows(crafted on 02/10/2025):



Based off of the design above, 4 windows with the description of each window as indicated, Oleg will work on the implementation, custom implementation first and Dagim will modify the custom implementation to make it json-friendly.



5. Actions

02/11/2025 – Oleg finished the implementation for custom protocol

Challenge: The custom protocol was not working with two computers, problems with IP addresses and wifis.

Meeting with Oleg: Tried different approach, approach 1 was to use a hotspot from Oleg's Wi-Fi and connect to the hotspot from two different devices, the approach didn't work and troubleshooting.

Approach 2: Used a hotspot from Harvard Secure Wi-Fi on my pc, and have Oleg connect to my hotspot, and have a MacBook serve as a server, which worked.

Json Implementation

Dagim completed the Json implementation, only minor changes needed and test files included, unit test implemented and the following command line can be run:

```
python -m coverage run -m unittest testcustomclient.py  
python -m coverage report -m
```

Dagim Gebrie, [12/02/2025 13:31]

```
python -m coverage run -m unittest testcustomserver.py  
python -m coverage run -m unittest testcustomclient.py  
python -m coverage report -m
```

Demo

Demo went great, what was left was a config file to not hard code IP Addresses and cleaning up the test files. Final check-up went great and ready to submit on February 14, 2025.

LLM Use:

During our implementation and troubleshooting, we used OpenAI models(co-pilot) to aid in code syntax, search for shortcomings and brainstorm ideas for implementation and robustness