PingMe

Team Members:

Izabela Camaj, Malik Issani, Christina Carvalho

Process: SCRUM

- An agile methodology for iterative project management
  - Prioritizes collaboration, adaptability, and **HIGH QUALITY** results

1. Sprints: Short phases for focused progress(Requirements, Design, Implementation, Testing)
2. Roles: Scrum leaders & team members
3. Tools: Github Backlog, Google Docs, VSC

- Why? **FLEXIBILITY**
- Approach: Weekly check-ins

Foundational Workflow Highlights

- Set up Jira Backlog to manage tasks and workflow
- Established shared availability and scheduled weekly meetings
  - In person or via Discord
- Defined project phases and rotating Scrum leaders
  - Requirements, Design, Implementation, and Testing
- Tracked progress and maintained detailed notes
  - With Visual Studio Code and shared Google Docs
- Conducted extensive research on necessary tools
- Embraced changes along the way by utilizing Scrum's agile framework
  - Prioritized product quality

Requirements Phase

- **SCRUM Leader:** Izabela Camaj
  - Reviewed requirements
  - Communicated with team to meet once a week
  - Initially assigned 2 requirements per member
  - Modified and finalized requirements.

- Example Backlog Tasks:
  - Create 2 Functional Requirements(Assigned to each member)
  - Modify Requirements
  - Finalize Requirements

Main Project Requirements

Authentication Server Services:

- User registration and login system.
- Password recovery mechanism.
- Token-based authentication for secure access.

1. **Programming Language**
2. **Networking Protocol**
3. **Features**
4. **Concurrency**
5. **Error Handling**
6. **Security**

Functional Requirements

1. The app shall allow new users to register a new account.
   a. The user shall create a unique username and password.
   b. The user shall answer two security questions.
   c. The user shall be given a 2 Factor Authentication code.
   d. The app shall store user information in a secure database.

2. The app shall authenticate users by verifying their username, password, and 2FA code before allowing the user to have access to the application.

3. The app shall issue a unique 2 Factor Authentication code to users upon successful login.
   a. The user shall utilize this same 2FA code when logging into the app.

4. The app shall allow users to recover their account password.
   a. The user shall share their username and answer 2 security questions in order to recover their 2FA code and password.

5. The app shall issue a session token after each successful login.

6. The app shall allow users to send chat messages to other concurrent users.
   a. The app shall only allow logged in users to send chat messages.

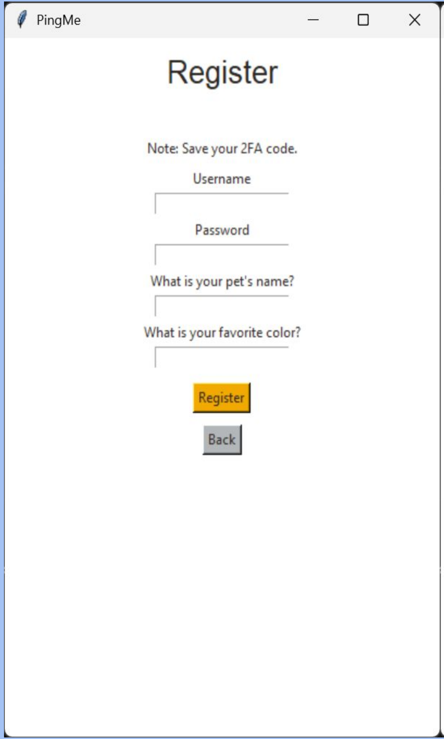
Design Phase

- Why TCP?
  - Guarantees the delivery of packets by checking for any data loss, duplication, or reorganization
  - Can handle multiple clients at the same time using threading
- Server Build:
  - Creates socket using IPV4 and TCP
  - Binds socket
  - Listens for requests from clients
- Handling Client Connections:
  - New thread for each connected client
  - Session tokens

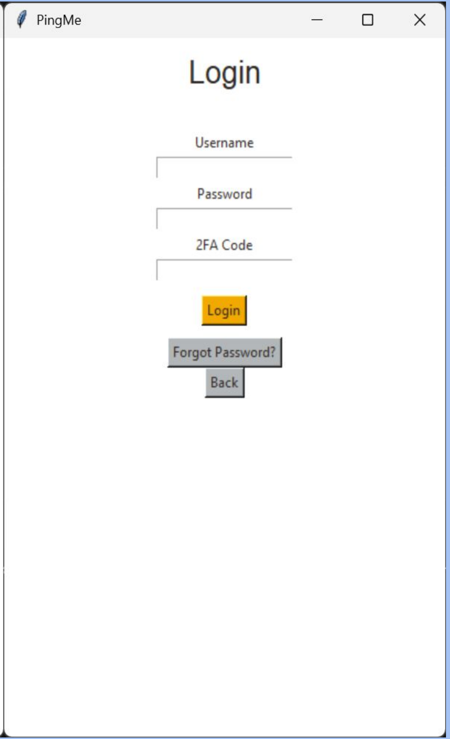
- Overview of Client Application:
  - Opens with home page
  - User creates an account
  - Login
  - Forgot Password
  - Chat Page
- Security Features:
  - Safe port binding
  - Checks for matching login info
  - Password hashing
  - Session tokens
  - 2-factor authentication
  - Security questions

Design Phase

- **Registering**
  - Enters username, password, secret questions
  - Info sent through socket to SQLite database
  - Checks for username availability
  - Hashes password
  - Stores in database
  - User receives 2-factor authentication code
- **Logging In**
  - Enters username, password, 2-factor code
  - Info sent through socket to SQLite database
  - Checks if info matches
  - Generates session token

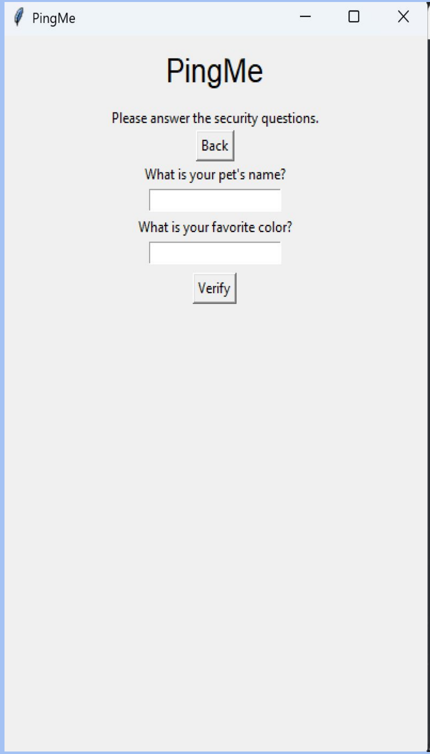


Note: Save your 2FA code.



Design Phase

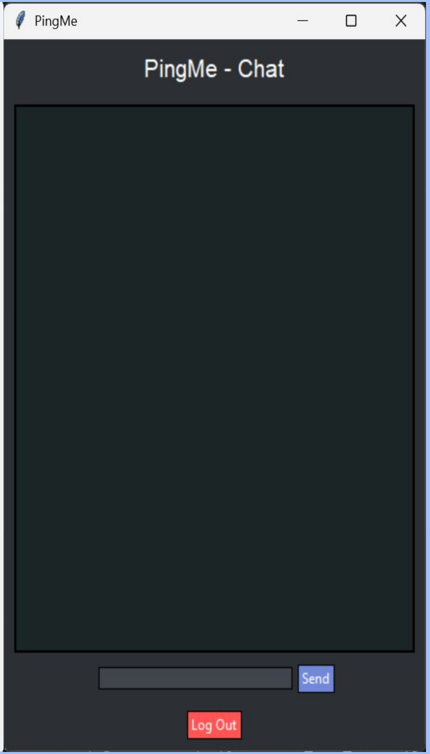
- **Password Recovery**
  - Enters username and secret question answers
  - Provides password and 2-factor code
- **Chat Functionality**
  - User logs in, has session token
  - Displays username and message to all concurrent users
- **Logging Out**
  - Session token deleted



Please answer the security questions.

What is your pet's name?

What is your favorite color?



Implementation Phase

➢ **Transition from Web Interface to Tkinter Desktop Application**

- Shift from web-based to desktop app based on user feedback
- Tkinter as a suitable choice for Python-based desktop app development

➢ **Race Conditions and Multi-Client Handling**

- Challenges of managing concurrent client connections and race conditions
- Implemented thread synchronization using locks to ensure data integrity

Implementation Phase

➢ **2-Factor Authentication Implementation**

- Initial plan: Implement 2FA with email/SMS
- Challenges with third-party integration, reliability issues
- Adapted to use security questions for 2FA

➢ **Password Recovery Mechanism**

- Recovery via security questions for account access recovery
- Ensured extra security with user-friendly experience

Testing Phase

➢ **Manual Testing and Simulated Clients**

- Extensive manual testing to simulate multiple client connections
- Verified message transmission, UI responsiveness, and user interactions

➢ **Successful Connections and Test Outcomes**

- Confirmed stable connections between clients and server
- Real-time updates and smooth user experience
- Simulated multiple clients connecting simultaneously
- Observed some performance degradation as client numbers increased
- Identified areas for optimization to handle larger loads

DEMO