Reddit Clone Implementation Report

Project Overview

This report details the implementation of a Reddit clone system utilizing a REST API architecture for client-server communication. The project implements core Reddit functionality including user registration, forum management, posting, commenting, voting, and direct messaging.

Team Members

1. Arpita Patnaik
2. Bala Tripura Kumari Bodapati

Demo Link

https://www.youtube.com/watch?v=RSbL_fuPvZ8&feature=youtu.be

System Architecture

Components

REST API Server (server.go)

- Handles HTTP requests using the Gorilla Mux router
- Implements CORS and logging middleware
- Routes requests to appropriate engine handlers
- Response standardization using common Response struct

Social Engine (social_engine.go)

- Core business logic implementation
- Actor-based concurrency using Proto Actor framework
- In-memory data structures for users, forums, posts, and messages
- Thread-safe operations using mutex locks

REST Client (rest_client.go)

- Command-line interface for interacting with the server
- Supports all core Reddit functionalities
- Handles JSON serialization/deserialization
- Configurable timeout settings

Protocol Definitions (messages.proto)

- Protocol Buffer definitions for all message types

- Structured data models for content, feedback, and chat
- Clear separation between request and response messages

Key Design Patterns

Actor Model

- Uses Proto Actor for concurrent message handling
- Ensures thread safety for shared state
- Enables scalable message processing

Repository Pattern

- Centralized data management in SocialEngine
- Clean separation of concerns
- Consistent data access patterns

Core Functionality Implementation

User Management

```
func (s *Server) registerUser(w http.ResponseWriter, r *http.Request) {
    // User registration implementation
}
```

- Simple username-based registration
- Unique username validation
- Activity status tracking

Forum Operations

- Forum creation with name and description
- Join/leave forum functionality
- Member tracking and forum statistics
- Forum content feed management

Content Management

- Post creation with title and body
- Hierarchical comment system
- Content ID generation for unique identification
- Support for reposting/sharing content

Voting System

- Upvote/downvote functionality
- Points calculation
- User karma tracking
- Hot score computation for content ranking

Direct Messaging

- User-to-user messaging
- Message status tracking (seen/unseen)
- Message retrieval with pagination
- Real-time message delivery

Running Multiple Clients

Setup Instructions

1. Start the server:

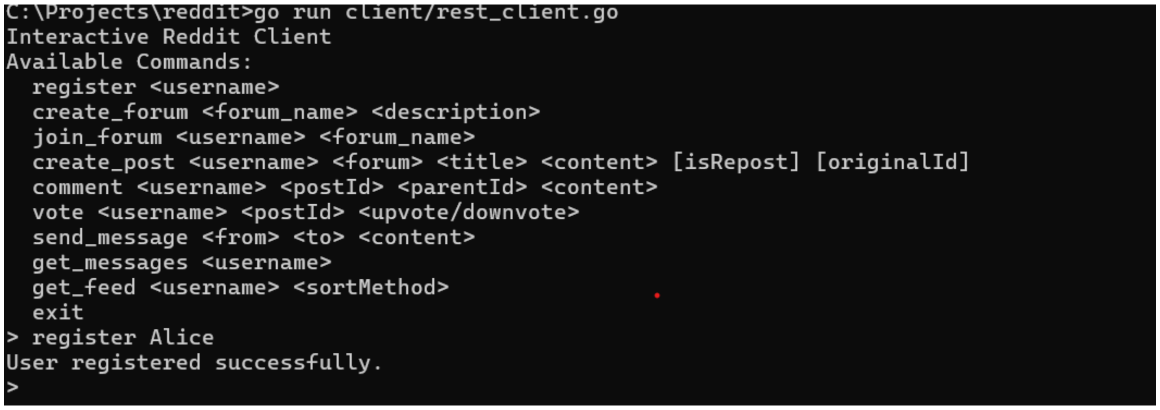
   go run cmd/server/main.go -port 8080 -actor-port 8085

2. Start multiple client instances:

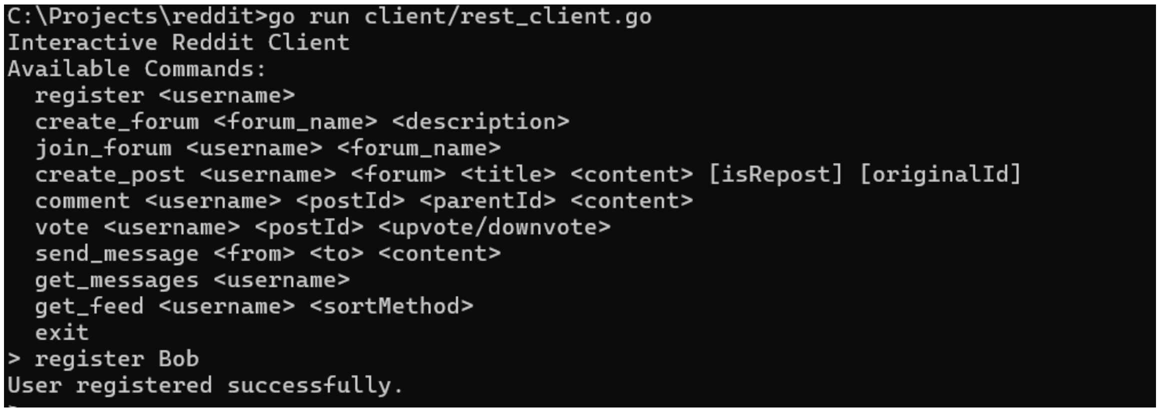
   go run client/rest_client.go

Testing Multiple Client Interactions

1. User Registration

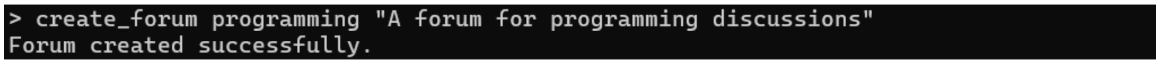


```
C:\Projects\reddit>go run client/rest_client.go
Interactive Reddit Client
Available Commands:
  register <username>
  create_forum <forum_name> <description>
  join_forum <username> <forum_name>
  create_post <username> <forum> <title> <content> [isRepost] [originalId]
  comment <username> <postId> <parentId> <content>
  vote <username> <postId> <upvote/downvote>
  send_message <from> <to> <content>
  get_messages <username>
  get_feed <username> <sortMethod>
  exit
> register Alice
User registered successfully.
>
```

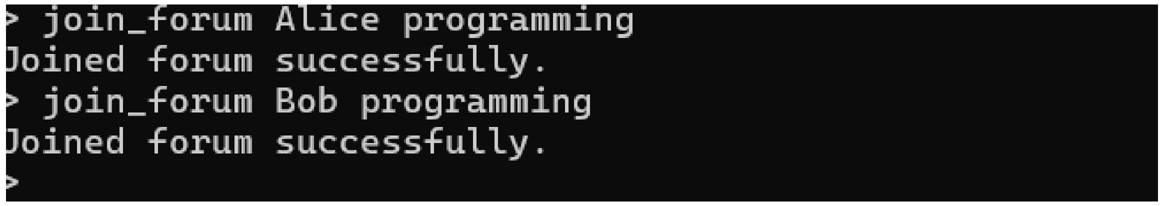


```
C:\Projects\reddit>go run client/rest_client.go
Interactive Reddit Client
Available Commands:
  register <username>
  create_forum <forum_name> <description>
  join_forum <username> <forum_name>
  create_post <username> <forum> <title> <content> [isRepost] [originalId]
  comment <username> <postId> <parentId> <content>
  vote <username> <postId> <upvote/downvote>
  send_message <from> <to> <content>
  get_messages <username>
  get_feed <username> <sortMethod>
  exit
> register Bob
User registered successfully.
>
```

2. Forum Creation and Joining

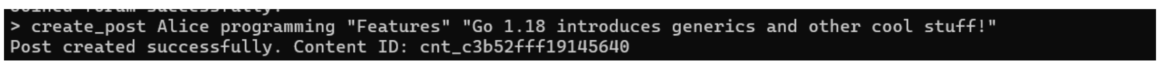


```
> create_forum programming "A forum for programming discussions"
Forum created successfully.
```



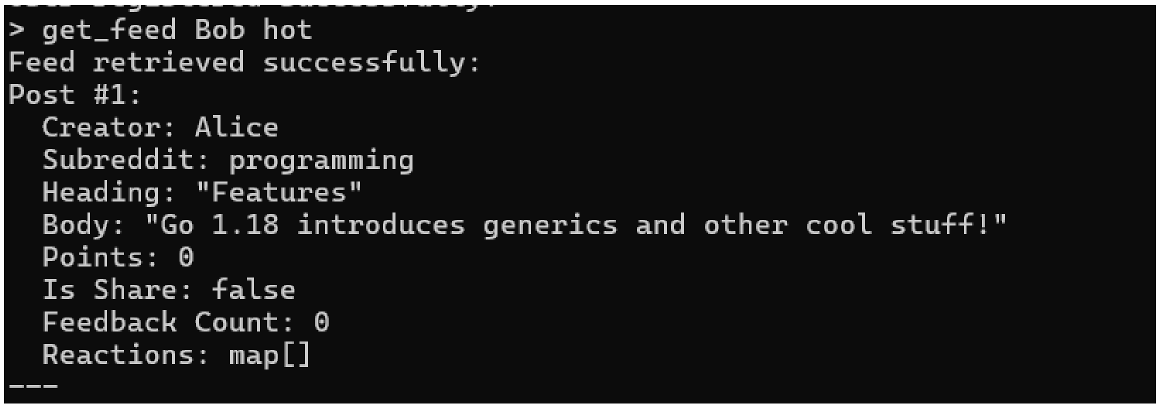
```
> join_forum Alice programming
Joined forum successfully.
> join_forum Bob programming
Joined forum successfully.
>
```

3. Create Post



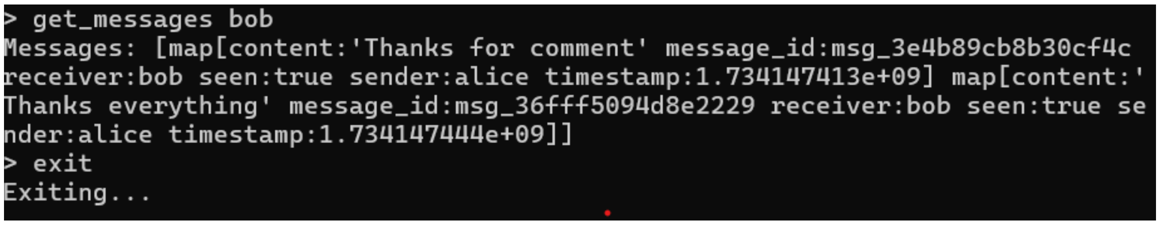
```
> create_post Alice programming "Features" "Go 1.18 introduces generics and other cool stuff!"
Post created successfully. Content ID: cnt_c3b52fff19145640
```

4. Get Feed



```
> get_feed Bob hot
Feed retrieved successfully:
Post #1:
  Creator: Alice
  Subreddit: programming
  Heading: "Features"
  Body: "Go 1.18 introduces generics and other cool stuff!"
  Points: 0
  Is Share: false
  Feedback Count: 0
  Reactions: map[]
---
```

5. Exit



```
> get_messages bob
Messages: [map[content:'Thanks for comment' message_id:msg_3e4b89cb8b30cf4c
receiver:bob seen:true sender:alice timestamp:1.734147413e+09] map[content:'
Thanks everything' message_id:msg_36fff5094d8e2229 receiver:bob seen:true se
nder:alice timestamp:1.734147444e+09]]
> exit
Exiting...
```

Conclusion

The implemented Reddit clone successfully demonstrates core functionality with a clean separation between client and server components. The REST API provides a standardized interface for client-server communication, while the actor-based engine ensures thread-safe operations. The system supports multiple simultaneous clients and implements all required features including user management, content creation, voting, and direct messaging.