Demo Day 1 Code Review: Aayush Karan & Michael Hla

ARTEMAS RADIK[1] AND SWATI GOEL[2]

[1]Harvard College, A.B./S.M. Computer Science & Statistcs, artemas@college.harvard.edu
[2]Harvard College, A.B. Computer Science & Physics, sgoel@college.harvard.edu

Compiled February 23, 2023

JOCN article style and format is being updated to conform to Optica journal style and format. This new template is now required for preparing a research article for submission to the *Journal of Optical Communications and Networking*. Consult the Author Style Guide for general information about manuscript preparation. Authors may also submit articles prepared using this template to the Optica Publishing Group preprint server, Optica Open. However, doing so is optional. Please refer to the submission guidelines found there. Note that copyright and licensing information should no longer be added to your Journal or Optica Open manuscript.

1. INTRODUCTION

This code review was conducted on commit `573fc69` of the repository submitted to us by Aayush and Michael, available publicly on GitHub. Aayush and Michael are reachable via email at `karan@college.harvard.edu` and `michaelhla@college.harvard.edu`, respectively. The code review aims to answer the following questions with regards to the submission:

1. Does it function as specified?
2. Is the code rational, correct, and clear?
3. Is the code and installation documentation adequate?

2. FUNCTIONALITY

We tested both the gRPC and non-GRPC versions of the program for the following functionaity.

1. Creating an account with a unique username.
2. List all accounts or a subset of the accounts by text wildcard.
3. Send a message to a recipient. If the recipient is logged in, deliver immediately; otherwise queue the message and deliver on demand. If the message is sent to someone who isn't a user, return an error message.
4. Deliver undelivered messages to a particular user.
5. Delete an account. You will need to specify the semantics of what happens if you attempt to delete an account that contains undelivered message.

Testing

We began by following the provided installation instructions in the engineering notebook, starting the non-gRPC server via `python3 socket_server.py 10.250.113.127 5000`. Our first test involved creating an account with the client as seen below.

Listing 1. Account Creation Test

```
python3 socket_client.py 10.250.113.127 5000
Welcome to Messenger! Please login or create an account:
Create Account
a
Account created. Welcome a!
```

We wanted to ensure that unique usernames were enforced, so we spawned another client and tried to create another account with the existing username.

Listing 2. Unique Username Test

```
python3 socket_client.py 10.250.113.127 5000
Welcome to Messenger! Please login or create an account:
Create Account
a
The account a already exists. Please try again.
```

We were satisfied with these two tests as evidence of account functionaity, so we began testing the username listing feature. As a prerequisite, we created additional accounts with usernames `blueberry`, `cranberry` and `elderberry`. Per the engineering notebook, the username listing feature follows Python RegEx. Thus we began by listing usernames following the `.*` expression, which we expected to return all usernames. We then listed usernamed following the `^.*berry.*$` expression, expecting the last three usernames. We were highly impressed with the RegEx matching functionaity.

Listing 3. Account List and Filter Test

```
python3 socket_client.py 10.250.113.127 5000
Welcome to Messenger! Please login or create an account:
List Accounts
.*
Users matching .*:
a blueberry cranberry elderberry
List Accounts
^.*berry.*$
Users matching ^.*berry.*$:
blueberry cranberry elderberry
```

The next test involved sending a message from the logged in `blueberry` client to the logged in `cranberry` client.

Listing 4. Send Live Message Test

```
python3 socket_client.py 10.250.113.127 5000
Welcome to Messenger! Please login or create an account:
Create Account
blueberry
Account created. Welcome blueberry!
Send
To:
cranberry
Message:
hello there, cranberry.

Message successfully sent.
```

The `cranberry` client successfully received the message.

Listing 5. Receive Live Message Test

```
python3 socket_client.py 10.250.113.127 5000
Welcome to Messenger! Please login or create an account:
Create Account
cranberry
Account created. Welcome cranberry!
<blueberry>: hello there, cranberry.
```

Next we logged out `blueberry` and sent an outbound message from `cranberry`. Upon re-logging in `blueberry` we requested our undelivered messages via `Open Undelivered Messages` as described in the engineering notebook.

Listing 6. Receive Undelivered Message Test

```
python3 socket_client.py 10.250.113.127 5000
Welcome to Messenger! Please login or create an account:
Login
blueberry
Welcome back blueberry!
Open Undelivered Messages
<cranberry>: Hi, blueberry!
```

The blueberry client successfully received the undelivered message. Sending a message to a non-existent user also correctly returns an error as seen below.

Listing 7. Send Message to Nonexistent User Test

```
python3 socket_client.py 10.250.113.127 5000
Welcome to Messenger! Please login or create an account:
Login
blueberry
Welcome back blueberry!
Send
To:
gooseberry
Message:
do you exist?
Sorry, message recipient not found. Please try again.
```

The last remaining functionaity we tested for was deletion, as seen below on the `blueberry` account.

Listing 8. Account Deletion Test

```
python3 socket_client.py 10.250.113.127 5000
Welcome to Messenger! Please login or create an account:
Login
blueberry
Welcome back blueberry!
Delete Account
Account blueberry successfully deleted.
Welcome to Messenger! Please login or create an account:
```

Thankfully, the gRPC submission command structure follows extremely closely to the non-gRPC one. The only distinction is that no prompt is required to automatically deliver queued messages on the gRPC implementation. We ran similarly extensive testing on the gRPC implementation and verify that all functionaity is present, but have omitted those tests from this report as they repeat much of what is already present.

3. CODE RATIONALITY, CORRECTNESS, AND CLARITY

TBD!

4. DOCUMENTATION

It is not necessary to place figures and tables at the back of the manuscript. Figures and tables should be sized as they are to appear in the final article. Do not include a separate list of figure captions and table titles.

Figures and Tables should be labelled and referenced in the standard way using the `\label{}` and `\ref{}` commands.

A. Sample Figure

Figure **??** shows an example figure.

B. Author Photographs

Author photographs. The final printed size of an author photograph is exactly 1 inch wide by 1 1/4 inches long (6 picas × 7 1/2 picas). Please ensure that the author photographs you submit are proportioned similarly.

C. Sample Table

Table 1 shows an example table.

Table 1. Shape Functions for Quadratic Line Elements

local node	$\{N\}_m$	$\{\Phi_i\}_m$ $(i = x, y, z)$
$m = 1$	$L_1(2L_1 - 1)$	Φ_{i1}
$m = 2$	$L_2(2L_2 - 1)$	Φ_{i2}
$m = 3$	$L_3 = 4L_1L_2$	Φ_{i3}

5. SAMPLE EQUATION

Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables with $\mathrm{E}[X_i] = \mu$ and $\mathrm{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n}\sum_{i}^{n} X_i \qquad (1)$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

6. SAMPLE ALGORITHM

Algorithms can be included using the commands as shown in algorithm 1.

7. SUPPLEMENTAL MATERIAL

Consult the Author Guidelines for Supplementary Materials in Optica's Journals for details on accepted types of materials and instructions on how to cite them. For preprints submitted to Optica Open a link to supplemental material should be included

Algorithm 1. Euclid's algorithm

```
1: procedure EUCLID($a, b$)                    ▷ The g.c.d. of a and b
2:     $r \leftarrow a \bmod b$
3:     while $r \neq 0$ do                     ▷ We have the answer if r is 0
4:         $a \leftarrow b$
5:         $b \leftarrow r$
6:         $r \leftarrow a \bmod b$
7:     return $b$                              ▷ The gcd is b
```

in the submission, but do not upload the material. All materials must be associated with a figure, table, or equation or be referenced in the results section of the manuscript. (1) 2D and 3D image files and video must be labeled "Visualization," not "Movie," "Video," "Figure," etc. (2) Machine-readable data (for example, csv files) must be labeled "Data File." Number data files and visualizations consecutively, e.g., "Visualization 1, Visualization 2. . . ." (3) Large datasets or code files must be placed in an open, archival database. Such items should be mentioned in the text as either "Dataset" or "Code," as appropriate, and also be cited in the references list. For example, "see Dataset 1 (Ref. [1]) and Code 1 (Ref [2])." Here are examples of the references:

A. Sample Dataset Citation

1. M. Partridge, "Spectra evolution during coating," figshare (2014) [retrieved 13 May 2015], http://dx.doi.org/10.6084/m9.figshare.1004612.

B. Sample Code Citation

2. C. Rivers, "Epipy: Python tools for epidemiology," (figshare, 2014) [retrieved 13 May 2015], http://dx.doi.org/10.6084/m9.figshare.1005064.

8. FUNDING AND ACKNOWLEDGMENTS

Formal funding sources should be listed in a separate paragraph block before any other acknowledgment information. Funding sources and any associated grant numbers should match the information entered into the Prism manuscript system. Funders should be listed without any introductory language or use of labels (do not use labels such as "grant no."). The acknowledgments may contain any information that is not related to funding. Here is an example:

FUNDING

National Science Foundation (NSF) (1263236, 0968895, 1102301); The 863 Program (2013AA014402).

ACKNOWLEDGMENTS

The authors thank H. Haase, C. Wiede, and J. Gabler for technical support.

9. REFERENCES

Full references (to aid the editor and reviewers) must be included. This will be produced automatically if you are using a .bib file.

Add citations manually or use BibTeX. See [**? ?**].

AUTHOR BIOGRAPHIES

First A. Author (M'76–SM'81–F'87) and the other authors may include biographies at the end of regular papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author's educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state, and country, and year degree was earned. The author's major field of study should be lower-cased.

The second paragraph uses the pronoun of the person (he or she) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests end the paragraph.

The third paragraph begins with the author's title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies. Finally, list any awards and work for committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of the biography. Personal hobbies will be deleted from the biography.

Alice Smith received her BSc (Mathematics) in 2000 from The University of Maryland. Her research interests also include lasers and optics.