

BU College of Engineering
Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

Final Testing Plan



RE3: Reproducibility, Reusability, Readability

Submitted to

Ana Trisovic
anatrisovic@g.harvard.edu

by

Team #5
RE3

Team Members

Andreas Francisco De Melo Oliveira andoliv@bu.edu
Ethan Hung ehung@bu.edu
Jyotsna Penumaka jyotsnap@bu.edu
Layan Bahaidarah layanb@bu.edu
Lukas Rosario lukasr@bu.edu

Required Materials

Hardware:

- Computer/Laptop - *Thats is it!*

Software:

- If you are visiting the website: any browser
- If you are running locally:
 - Front End
 - React JS web application
 - Tailwind CSS
 - Firebase JavaScript SDK
 - Backend:
 - Google Firebase
 - Firestore
 - Storage
 - Authentication
 - Python Scripts/Notebooks
 - Flask
 - Miniconda3
 - Docker

Set Up:

Since our project is just software based, all of the set up required is to simply access the re3.ai website to access our platform.

Local Set-up:

In order to run the app locally, run npm start to load our web app. Also to run the server backend code for the readability page you can execute the following commands:

- To open the project on a web browser, follow the following steps:
 - cd 21-05-Re3/re3-client
 - npm install
 - npm start
- To run the flask server that calls our machine learning model:
 - cd Re3-readability/flask_api/app
 - python3 main.py
- To run the flask server that lets a user execute the docker file:
 - cd 21-05-Re3/docker-app
 - python3 app.py

Testing Procedure :

1. Go to the <https://re3.ai> or <http://localhost:3000> (if you are running locally)
2. Login with a google email
3. Transfer User to the main page that will allow the user to choose one of the offered services.
4. Navigate to the Rating Page where the user code can rate 100 snippets of R code to improve the ML model.
 - a. Click on any of the numbers below the text box to rate the snippet
 - b. Test next or previous buttons.
 - c. Close tab and verify ratings are saved in firestore.
5. Navigate to the the Readability page where the user can use the ML model page
 - a. Upload a .R file in the readability page and click the get rating button to obtain:
 - i. Readability score of file
 - ii. Any suggestions (if any)
6. Navigate to the Reproducibility page where users can test their files
 - a. Store the R code files in firestore storage
 - b. Send R files to server
 - c. Run sample R code.

Measurable Criteria:

The criteria for successful running and output is as follows:

To test the code rating functionality:

1. Rate 10 snippets while using the previous and back buttons and verify ratings are saved correctly in firestore.

To test the readability machine learning model:

1. Upload a file into the choose file box and get an output rating.
2. Obtain a red, yellow, and green score.
3. Get the appropriate suggestions from the suggestion box, which include:
 - a. Breaking up your lines
 - b. Diminish the periods in your lines
 - c. Diminishing the parentheses in your lines
 - d. Commenting your code if no comments are found
 - e. Diminishing the number of assignments in your lines

To test the reproducibility aspect of the project:

1. Users should be able to specify the files they would like to upload.
 - a. Should be successfully uploaded to the firebase storage.
2. Users should be able to decide the order in which files are run in the docker.
3. Users should be able to specify the R version of their choice
4. The specified R version should be used to run the docker file.
5. The following information should be collected and saved to firestore:
 - a. Author name
 - b. Title

- c. Key Words
- 6. Readability scores should be generated upon uploading and saved to firestore.

What should you expect?

The screenshot shows a web interface titled "Code and Dataset Information". It features several input fields: "Author Name" (with placeholder "ex: John Doe, Jane Doe"), "Title" (with placeholder "ex: A Study in Reproducibility"), "R Version Used" (a dropdown menu with "Select Version"), "Files to Upload" (with an "Add Files" button and a blue checkmark icon), and "Information" (with an "Enter Information" button). To the right of these fields are two blue boxes: "Unordered" containing three snippets labeled "snippet1.R", "snippet2.R", and "snippet3.R", each with a close icon; and "Ordered" which is currently empty. At the bottom center is a black "Run Code" button. The top right of the interface has links for "My Account" and "Sign out".

Figure 1 : Collect information about the research project dependencies to reproduce code. The user should be able to upload their files and order the files depending on how the user wants to run the files in the container.

This screenshot shows a modal window titled "Licenses and associated keywords" overlaid on the same form as Figure 1. The modal contains three input fields: "Code License:" (with placeholder "ex: Apache License 2.0"), "Data License:" (with placeholder "ex: PDDL"), and "Key Words:" (with placeholder "ex: R code, Reproducibility"). A blue "Done" button is at the bottom of the modal. The "Run Code" button from the background form is still visible at the bottom of the screen.

Figure 2: Collect information about the research project, the user plans to upload

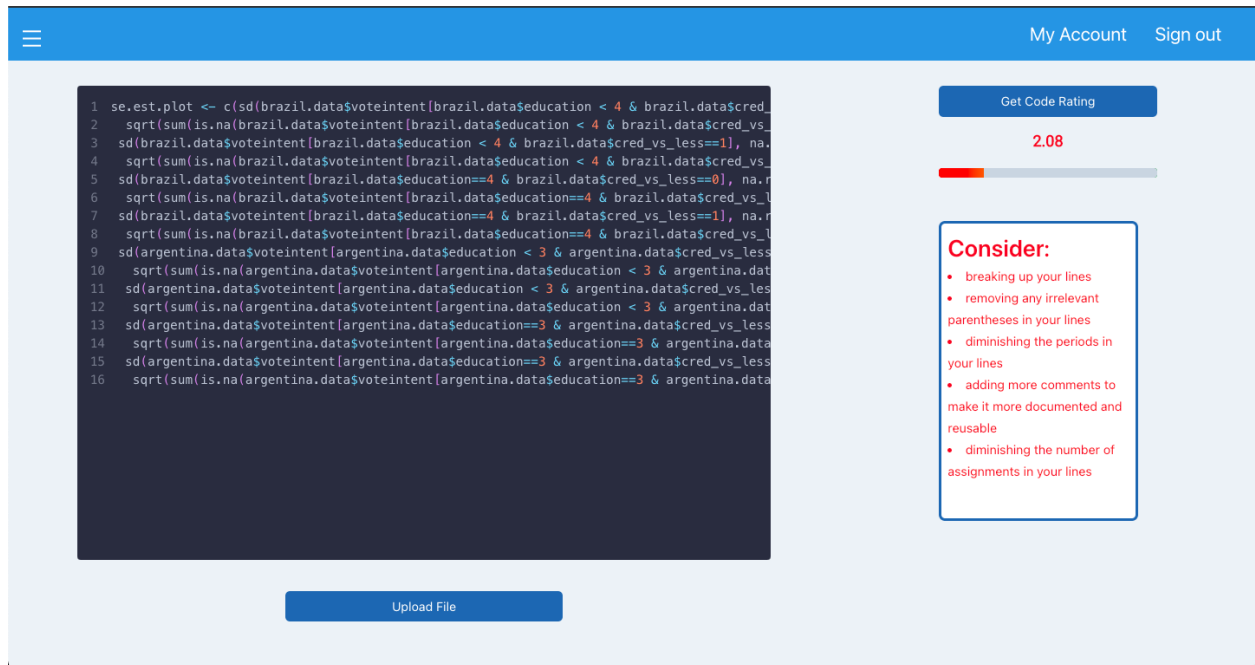


Figure 4: Upload file on the readability page and the code readability rating of such file with suggestions

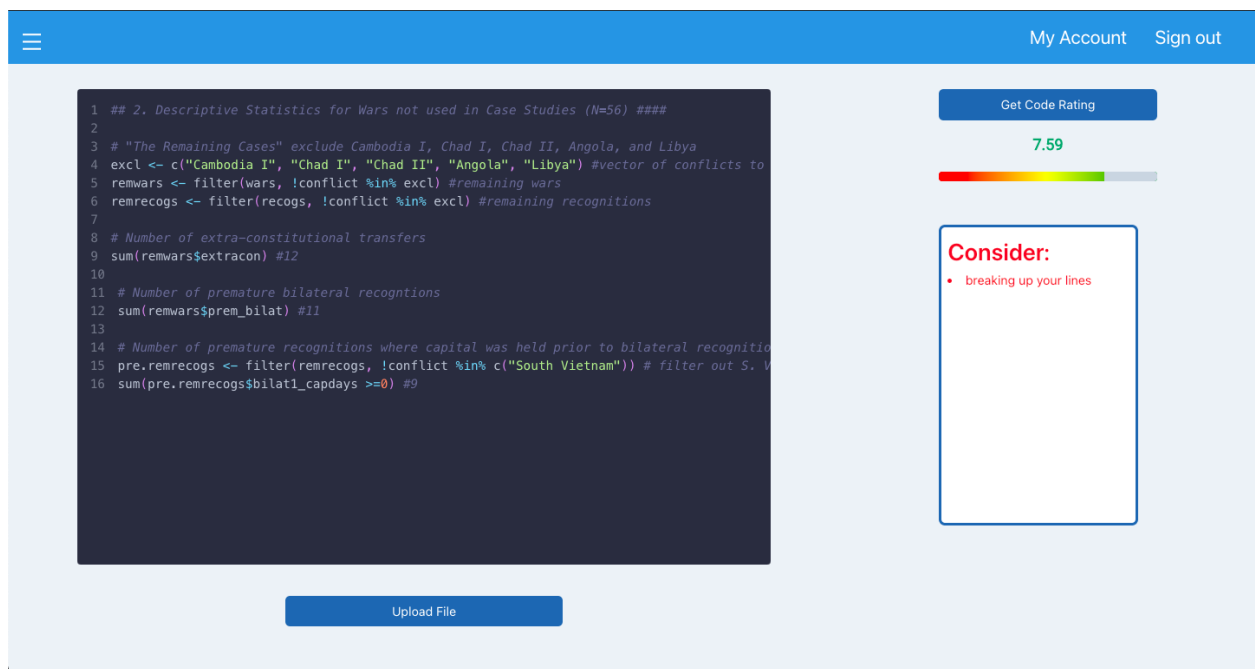


Figure 5: Upload file and get better rating if code is more organized and well commented

