Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

Second Prototype Testing Plan

Reproducibility, Reusability, Readability (RE3)

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:

- Front End
    - React JS web application
    - Tailwind CSS
    - Firebase JavaScript SDK
- Backend:
    - Google Firebase
        - Firestore
        - Storage
        - Authentication
    - Python Scripts/Notebooks
    - Flask
    - Miniconda3
    - Docker
    - SocketIO

Set Up:

Since our project is just software based all of the set up required is to 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:

- Go to lb/readability_reproducibility_UI
- 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 lets a user execute the docker file :

- To start the flask server
    - cd 21-05-Re3/docker-app
    - python3 main.py

Testing Procedure:

1. Go to the http://localhost:3000
2. Login with 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 get the readability score for the uploaded file.
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 and dependency information 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.
    a. Obtain a red, yellow, and green score.

To test the reproducibility aspect of the project:

1. Users should be able to specify the files they would like to upload.
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. These files should be successfully uploaded to the firebase storage.

What should you expect?

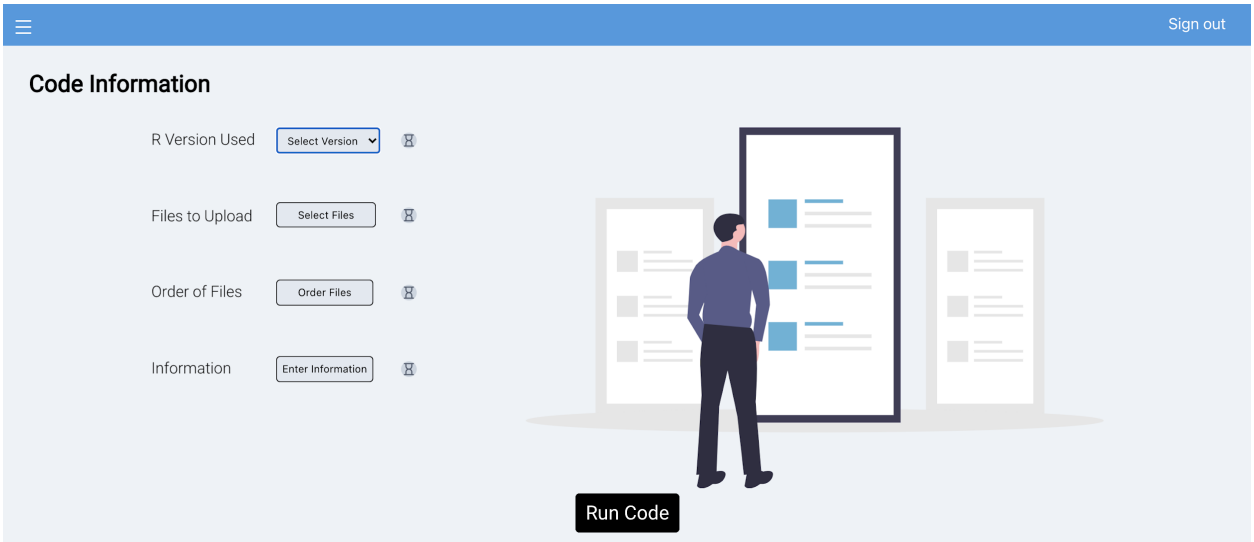

Figure 1 : Collect information about the research project dependencies to reproduce code

Figure 2 : Run the docker file using the specified R version and stream the output logs

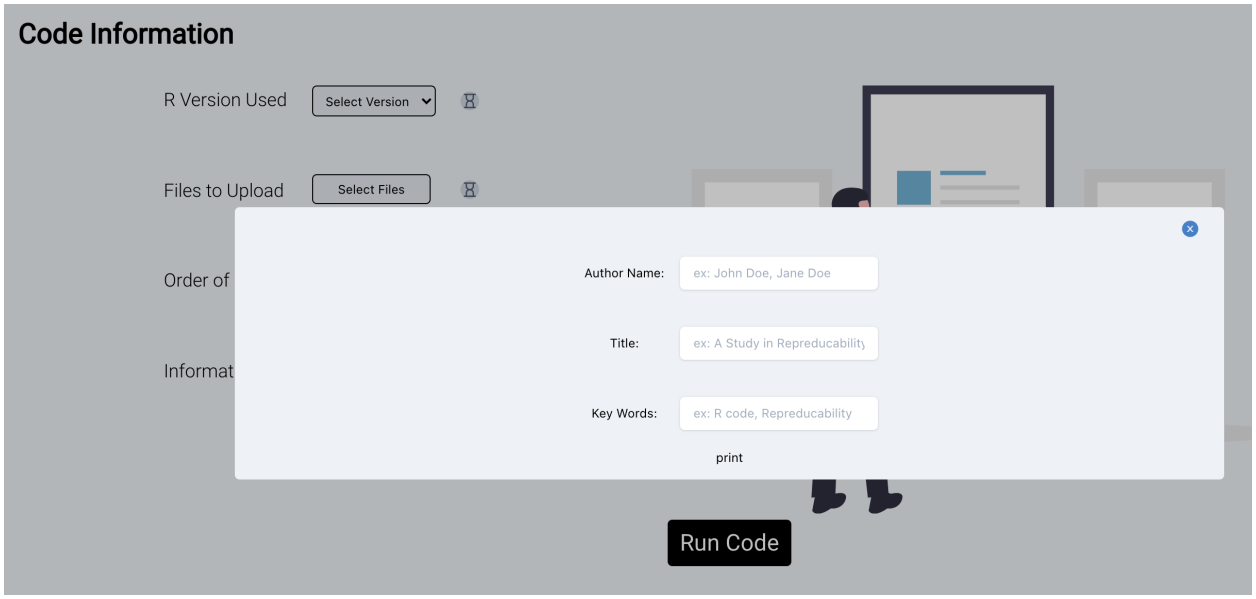

Figure 3: Collect information about the research project, the user plans to upload

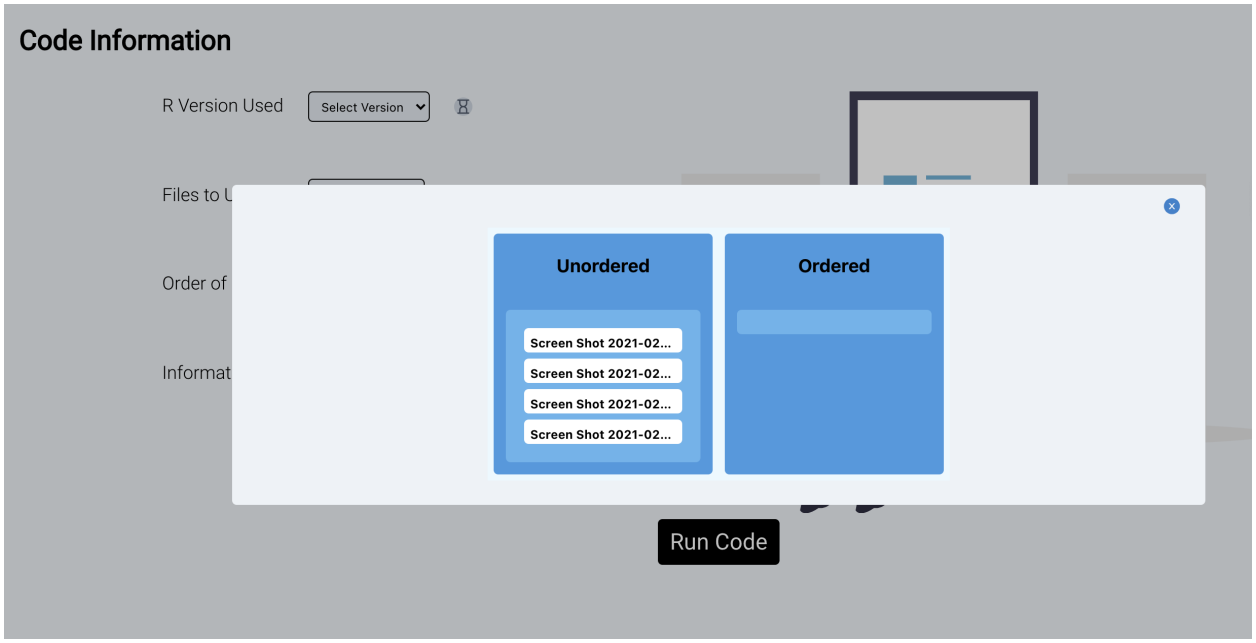

Figure 4: The user should be able to upload their files and order the files depending on how the user wants to run the files in the container.

Figure 5: Upload file on the readability page and the code readability rating of such file

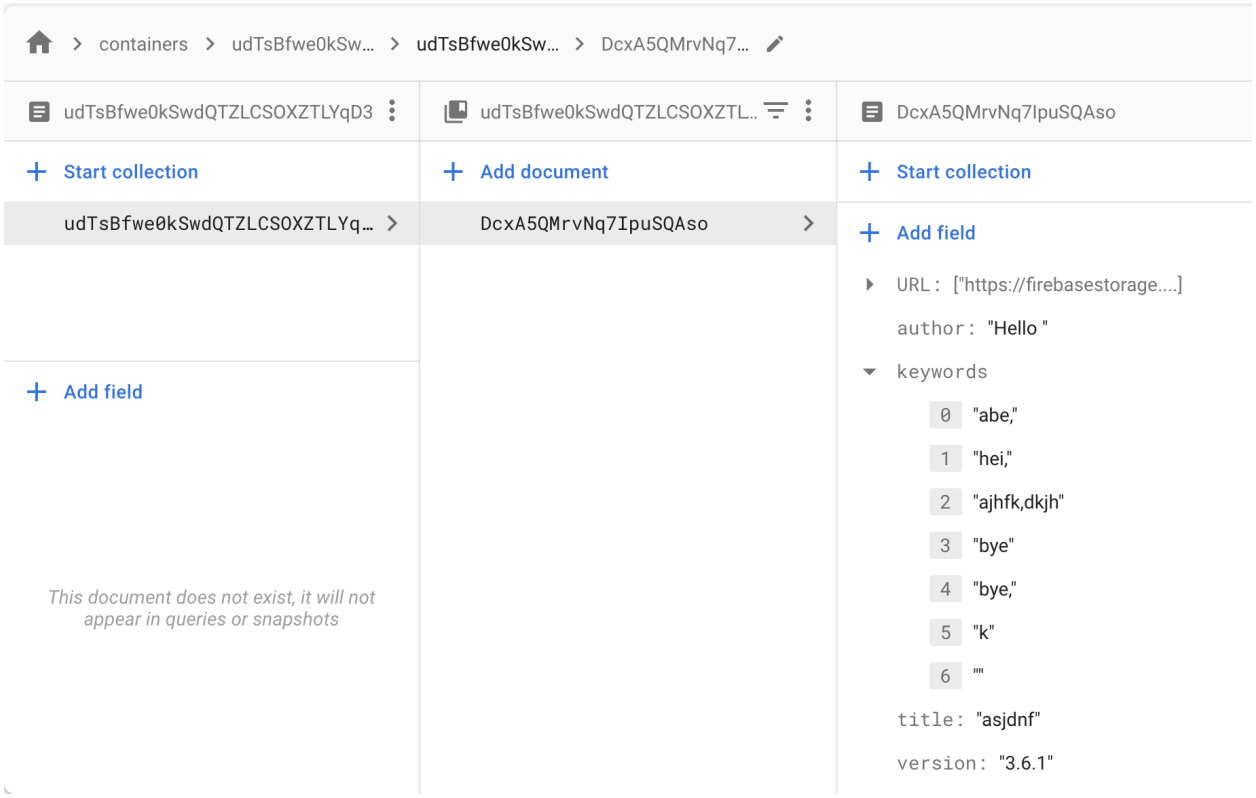

Figure 6: Upload the files details to Firestore, the details include the URL, author, keywords, title of the project, R version

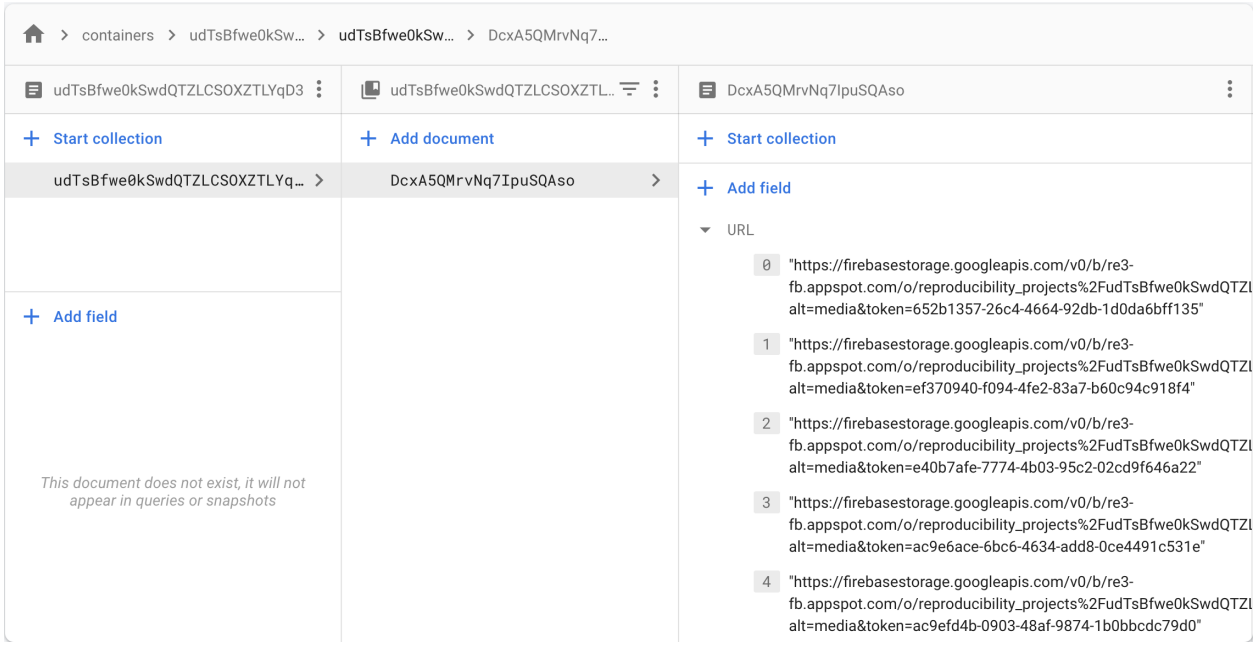

Figure 7: Upload the file projects to Firebase Storage and save the links in the firestore