Identifying the Bugs in LokOS App and Suggesting a Few Remedies

Report

Team:

Singh Ayush Kumar Satish | CSE | IIITD
Vibhor Agarwal | CSAM | IIITD

Advisors:

Dr. Mukulika Maity | CSE - Department | IIITD
Dr. Arani Bhattacharya | CSE, ECE - Department | IIITD

Introduction:

This report outlines the testing of the LokOS application. The purpose of the testing was to identify any issues with the application and provide recommendations for improvement. The focus of the testing was on the issue of data not getting successfully uploaded to the server.

Methodology:

To test the LokOS application, the testing team wrote an **Appium script** to automate the application's actions. The script was designed to create an SHG of a desired size, mainly focusing on SHGs with **5 members.** Each SHG profile included **21 photos** and **1 PDF-Resolution Copy** to be uploaded. Specifically, the SHG profile had **4 pictures** for each member, including the **member's image, consent form for Aadhaar, Aadhaar front, and Aadhaar rear, and one picture for the SHG's bank passbook front page.**

After successfully creating an SHG profile, the testing team attempted to upload the SHG data at different internet speeds and observed the results. The purpose of this test was to verify the application's ability to upload large amounts of data reliably and efficiently, under varying network conditions.

Testing:

The testing team carried out several tests to evaluate the performance of the LokOS application, with a focus on its ability to upload large amounts of data to the server. Two types of SHGs were created, both consisting of five members, with different characteristics.

- **Type 1:** SHGs included **21 black images** in the data, and the size of the SHG data was **3.18 MB.**
- **Type 2:** SHGs included **21 color images** in the data, and the size of the SHG data was **21 MB.**

Note: A SHG of size 5 members have at least 21 Pictures + 1 SHG Resolution Copy.

To simulate different network conditions, the testing team conducted the data upload tests at **3 different upload speeds: 256Kbps, 512Kbps, and 1Mbps.** The primary objective of this test was to identify any issues with the application's ability to upload large amounts of data under varying network conditions.
Also, an additional test was conducted in which the team reuploaded an SHG, which was earlier rejected by the **Block Program Manager** by just changing the **Resolution Copy(PDF)** of the SHG.

Results:

Type 1:
During the data upload tests, the testing team observed that **Type 1 SHG data**, consisting of **21 black images** and a size of **3.18MB**, was successfully uploaded to the server at all network conditions, including **256Kbps, 512Kbps, and 1Mbps.**
Note: Upload Duration at 256Kbps - 2Mins 21Secs.

Type 2:
For **Type 2 SHG data**, consisting of **21 color images** and a size of **21MB**, the upload was not successful at 256Kbps and 512Kbps. During the testing, the upload attempts were made **twice at 256Kbps** and **once at 512Kbps**, and all attempts **failed.** Only partial data was uploaded during the attempts, uploading **4.7MB and 10.92MB** respectively at **256Kbps** with an upload duration of **3 Mins 56 Secs** in the first case(4.7MB). The second attempt at **512Kbps** uploaded **10.85 MB** with an upload duration of **4 Mins 01 Secs**. However, at **1Mbps**, the data was successfully uploaded to the server without any issues.

Regarding the **additional test** that involved the re-uploading of SHG data with an updated resolution copy, only the **Resolution Copy Data** was sent to the server, and not the entire Data.

Observation:

Positives:

1. One of the strengths of the application is its ability to identify and upload only the updated data, without re-uploading the entire set of information. This feature was observed during our testing, where we specifically updated only the Resolution Copy of an SHG, and the application successfully uploaded only the updated file.

Issues:

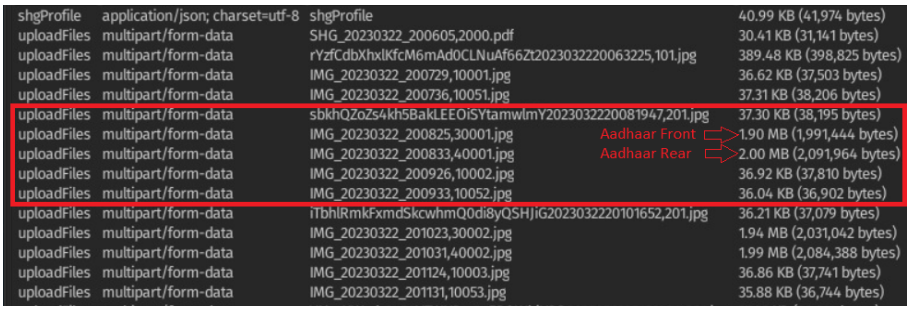

Fig 1.1 - SHG Data with Colour Images

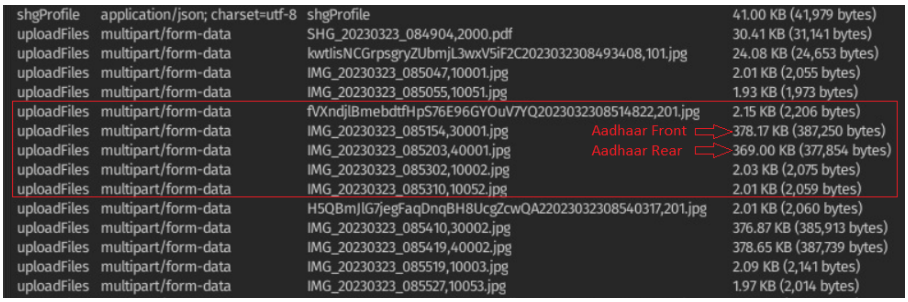

Fig 1.2 - SHG Data with Black Images

1. One issue observed during the testing was that an SHG of 5 members requires at least **21 images**, with **4 images for each member**, including their **personal image, Aadhaar consent form image, and Aadhaar front and rear images.** However, in both **Fig 1.1** (SHG data with colorful images) and **Fig 1.2** (SHG data with black images), the data under the **red box** are **"jpg" / "image" files.** Among these, the last 4 images correspond to the images mentioned above. As is visible from the above images the Aadhaar Front and Back Images are around **2MB** while the rest of the images are **significantly smaller of**

size 36KB for colored images. **It is evident that Aadhaar front and rear images are not compressed as much as the other images, resulting in larger file sizes.**

2. The application does not have the capability to **resume uploads from where they failed**, forcing the user to restart the entire process in case of a failed upload. The user is not shown a progress bar of the status of the upload, though he is notified when the upload is completed successfully. However, if the data transfer failed, a **failed message is not shown explicitly**.
3. The data is not **fragmented into smaller chunks** based on the network conditions, which could have increased the chances of successful data transfer.

Conclusion:

1. The application could **improve the compression** of Aadhaar front and rear images to reduce the overall size of the data, which could enhance the upload process.
2. Implementing the capability to **resume uploads** from where they failed could save users from the inconvenience of having to restart the entire process, which could also save time.

Future Work:

1. We can **fragment the data into smaller-sized packets** based on the network conditions. In case of low bandwidth, sending data one by one as several such smaller fragments can lead to a better data transfer success rate. This can also help in selectively sending the failed data fragments instead of the complete data again.

Resources:

Lok-OS Testing Automation Script
LokOS-Testing Data