Configuration Manual

MSc Research Project
Cloud Computing

Tamunobelema Miebaka-Ogan
Student ID: 20231890

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Tamunobelema Miebaka-Ogan

Student ID: 20231890

Programme: Cloud Computing **Year:** 2022

Module: MSc Research Project

Supervisor: Sean Heeney

Submission Due Date: 15/08/2022

Project Title: Configuration Manual

Word Count: 837 **Page Count** 6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……………………………………………………………………………………………

Date: 13/08/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Tamunobelema Miebaka-Ogan
20231890
MSc in Cloud Computing

14th August 2022

1 Introduction

In the research paper, a blockchain-based incentive mechanism was developed to enable the rewarding and verification of participants in task offloading to volunteered mobile devices without compromising their privacy. This configuration manual provides the information and steps required to replicate the results obtained in the evaluation. The structure of this document is; section 2 discusses the technologies required for the prototype that was built. Section 3 discusses the system configuration used to build the prototype. Section 4 discusses how to open the blockchain-based incentive code. Section 5 discusses how to run the mobile application component of the prototype. Section 6 discusses how to execute the task offloading using the mobile application. And Section 7 discusses how to evaluate the blockchain-based incentive mechanism.

2 Technologies Required

This section discusses the various technologies used in the development of the prototype in this project.

- Solidity: The smart contract for the blockchain-based incentive mechanism was written in solidity.
- Remix IDE: The smart contract was written, compiled, and deployed using the Remix IDE.
- Metamask: Metamask was used as the web3 provider to deploy the smart contract to the blockchain.
- Infura: The Infura platform was used to generate the API key to connect to the Ropsten network in order to interact with the deployed smart contract.
- Etherscan: This was used to verify the deployed smart contract token and view the transactions performed with the token.
- NodeJs: This was used to write the functions that interacted with the smart contract to initiate transfers to the volunteer. It was also used to write the function that created a blockchain address for the volunteer.

- Visual Studio Code: This IDE was used to write the NodeJs functions that interacted with the blockchain-based incentive mechanism and the cloud controller.
- Firebase: This was used to provide the cloud infrastructure for the project. The ML Kit was also used to execute the Optical Character Recognition and Face detection task on the mobile application. Furthermore, the monitoring tool provided by Firebase was used to monitor the execution of the various modules in the blockchain-based incentive mechanism.
- Android studio: The android studio IDE was used to build the android mobile application that registered devices, offloaded tasks, and executed the offloaded tasks.
- Kotlin: The android application was written in Kotlin.

3 System configuration

This section shows the configuration of the system in terms of hardware and software used to build the application.

3.1 Hardware requirements

- Model: Apple MacBook Pro M1 2020
- Processor: Apple M1 Chip
- Memory: 8GB RAM
- Storage: 256GB SSD

3.2 Software requirements

- macOS Monterey 12.5
- Android Studio IDE
- Visual Studio IDE

4 Blockchain-based incentive mechanism code

The blockchain-based incentive mechanism was implemented using Firebase cloud functions. To view these functions open the code_offloading_incentive folder with Visual studio code. These functions were deployed to Firebase using the Firebase CLI.

5 Running the Mobile Application

To run the android application, you need to open the CodeOffloadingIncentive folder with Android Studio IDE. Then connect an android device to the Android Studio IDE through cable or WiFi. To install the mobile application on the connected device, click the run button in Android Studio IDE. Figure 1 shows the run button on Android studio IDE that is represented by a play symbol.

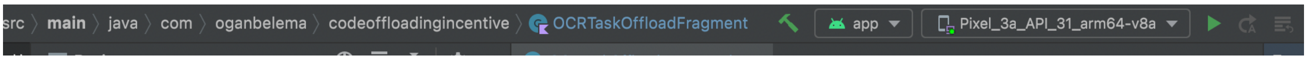


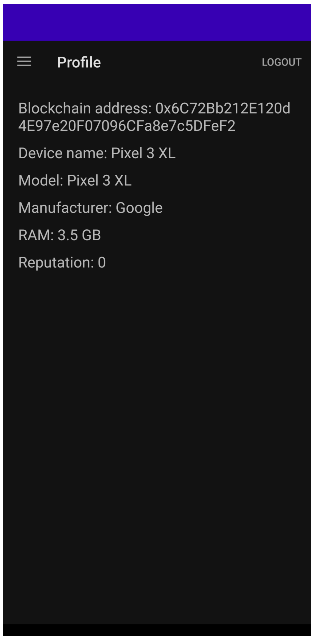
Figure 1: Run button in Android studio IDE

6 Performing task Offloading

To perform task offloading, you need to install the mobile application on an android device as explained in the previous section. Then execute the following steps:

- Register on the developed system providing an email address and password.
- Open the side menu and select the Volunteers option. This opens the Volunteers screen, showing other volunteers on the system.
- Select a volunteer, and a pop-up menu showing the OCR task and face detection task option is displayed.
- Select the task you want to offload. This action takes you to the task's offload screen where you click the upload image button to upload an image to the volunteer.
- Select the image you want the task to be performed on and this is sent to the selected volunteer.

The profile of a volunteer before executing offloaded tasks is shown in Figure 2. Figure 3 shows the Volunteers screen in the mobile application.



Blockchain address: 0x6C72Bb212E120d
4E97e20F07096CFa8e7c5DFeF2

Device name: Pixel 3 XL

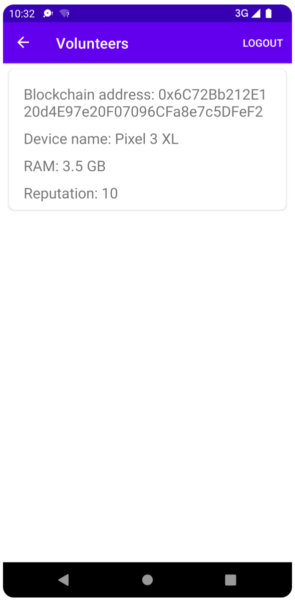
Model: Pixel 3 XL

Manufacturer: Google

RAM: 3.5 GB

Reputation: 0

Figure 2: Volunteers profile screen on mobile application



Blockchain address: 0x6C72Bb212E1 20d4E97e20F07096CFa8e7c5DFeF2

Device name: Pixel 3 XL

RAM: 3.5 GB

Reputation: 10

Figure 3: Volunteers screen with the updated reputation of volunteer

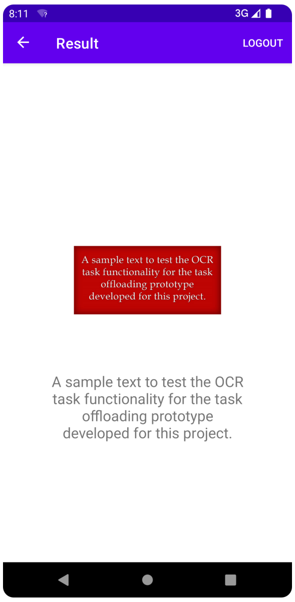
The following steps show how to execute the tasks offloaded on the assigned volunteer's system.

- Open the side menu and select the Tasks option. This shows the list of tasks assigned to the volunteer.
- Select a task that has a pending status and this opens the task execution screen.
- Click the button to execute the task. The task is then executed and the result is sent to the cloud controller.

The following steps show how to view the results of the offloaded tasks.

- Open the side menu and select the Results option. This shows the results of the offloaded tasks.
- Select a result on the list to view the details.

The result detail screen for an OCR task is shown in Figure 4



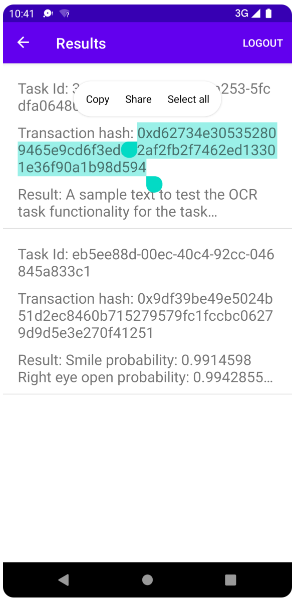
A sample text to test the OCR
task functionality for the task
offloading prototype
developed for this project.

A sample text to test the OCR
task functionality for the task
offloading prototype
developed for this project.

Figure 4: OCR Task result detail screen

7 Evaluate Blockchain-based incentive

To confirm that the blockchain-based incentive works as expected, copy the transaction hash associated with the result received. This is shown in Figure 5. Query it on `https://ropsten.etherscan.io/`. This is shown in Figure 6. This action verifies the source of the result as the selected volunteer's blockchain address should be in the **to** detail. Furthermore, the reputation of the volunteer that executed the offloaded task should increase.



Task Id: 3 253-5fc
dfa0648

Transaction hash: 0xd62734e30535280
9465e9cd6f3ed 2af2fb2f7462ed1330
1e36f90a1b98d594

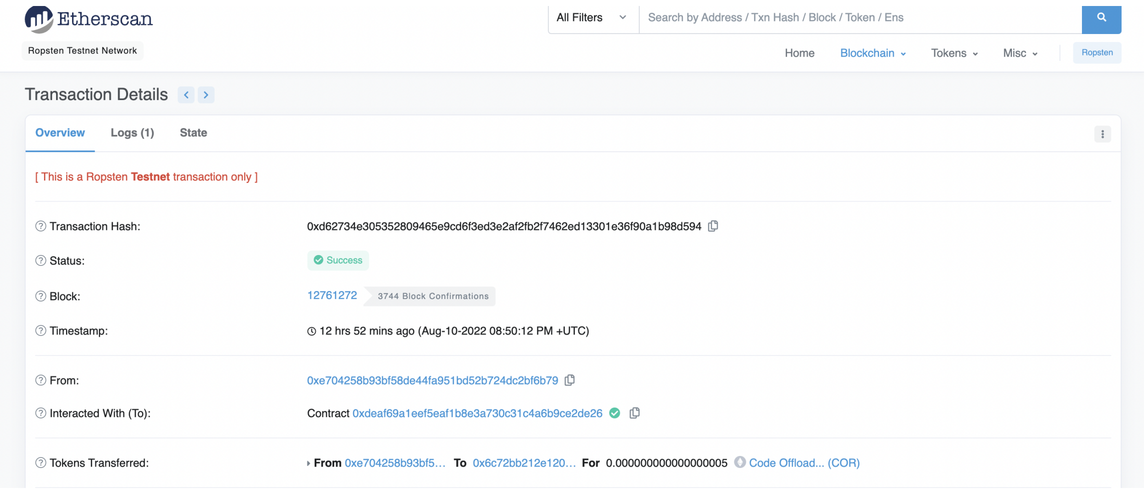
Result: A sample text to test the OCR
task functionality for the task...

Task Id: eb5ee88d-00ec-40c4-92cc-046
845a833c1

Transaction hash: 0x9df39be49e5024b
51d2ec8460b715279579fc1fccbc0627
9d9d5e3e270f41251

Result: Smile probability: 0.9914598
Right eye open probability: 0.9942855...

Figure 5: Results screen on Mobile application



Transaction Details

[This is a Ropsten **Testnet** transaction only]

Transaction Hash:	0xd62734e305352809465e9cd6f3ed3e2af2fb2f7462ed13301e36f90a1b98d594
Status:	Success
Block:	12761272 3744 Block Confirmations
Timestamp:	12 hrs 52 mins ago (Aug-10-2022 08:50:12 PM +UTC)
From:	0xe704258b93bf58de44fa951bd52b724dc2bf6b79
Interacted With (To):	Contract 0xdeaf69a1eef5eaf1b8e3a730c31c4a6b9ce2de26
Tokens Transferred:	**From** 0xe704258b93bf5… **To** 0x6c72bb212e120… **For** 0.000000000000000005 Code Offload... (COR)

Figure 6: Transaction on Etherscan