

SOFTWARE DESIGN DOCUMENT

Eth-Photo

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Document Conventions	1
1.3	Intended Audience and Reading Suggestions	1
1.4	Product Scope	1
1.5	References	2
2	Description	3
2.1	Product Perspective	3
2.2	Product Functions	3
2.3	User Classes and Characteristics	3
2.4	Operating Environment	4
2.5	Design and Implementation Constraints	4
2.6	Dependencies	4
3	External Interface Requirements	5
3.1	User Interface	5
3.2	Software Interface	5
3.3	Hardware Interface	5
3.4	Communication Interfaces	5
4	System Features	6
4.1	Map-based Frontend	6
4.1.1	Description and Priority	6
4.1.2	Stimulus/Response Sequences	6
4.1.3	Functional Requirements	6
4.2	Geo-Tagging/Topic based Tagging	6
4.2.1	Description and Priority	6
4.2.2	Stimulus/Response Sequences	6
4.2.3	Functional Requirement	6
5	Other Requirements	7
5.1	Performance Requirements	7
5.2	Safety Requirements	7
5.3	Security Requirements	7
5.4	Software Quality Attributes	7
5.4.1	Availability	7
5.4.2	Flexibility	7
5.4.3	Interoperability	7
5.4.4	Maintainability	8
5.4.5	Security	8
5.4.6	Reliability	8
5.4.7	Usability	8

6 Analysis Models	9
6.1 Control Flow Diagrams (CFD)	9
6.2 Data Flow Diagrams (DFD)	10
7 Test Cases	17
7.1 Show Thumbnails	17
7.2 Search Tags	18
7.3 Add Photo	18
7.3.1 Place Marker	18
7.3.2 Upload Image	19
7.3.3 After Uploading	19
7.4 Delete Image	20
A Glossary	21

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to provide the Support Information for the Software EthPhoto. It explains the required functionality of the Software and its features.

It demonstrates the need for developing Photo Sharing Application using Decentralized Technology.

1.2 Document Conventions

It follows the conventions of Standard IEEE template. The Acronyms used in this document are abbreviated in the Glossary Section of this document.

1.3 Intended Audience and Reading Suggestions

- Developers who can review project's capabilities for facilitating them to understand the requirements easily so that they can realize where their efforts should be targeted to design and code the application.
- Project Testers can use this document as a base for their testing strategy as some bugs are easier to be found using this.
- End Users of this software who wish to read about what this software can do.

1.4 Product Scope

- This software serves as a replacement for the present PhotoSharing Applications which uses a centralized server for fulfilling User requests.
- This software should be built on Ethereum Platform which uses BlockChain Technology which makes this DApp decentralized allowing each and every user to identify other's activities.

1.5 References

- This Document is prepared mainly by referring to the provided problem statement document.
- For software reference, one can use the screen shots and test case results provided along with the Application.

Chapter 2

Description

2.1 Product Perspective

The present applications, websites are based on web2 technology which uses central Server to fulfill User Requests. Using Central Server may give rise to risk of Spamming, Fraud and Virus attack. Hence there is a need to develop applications on Decentralized Technology.

EthPhoto is a Dapp to fulfill the above requirements which is built using Ethereum. Ethereum is a blockchain based platform that helps in building decentralized applications, where there is no centralized server and multiple users participate in the process through the blockchain platform.

This application instead of having central database to store data, the data is replicated to all the participants in the form of Blockchain, that ensures validity of data in spite of having malicious users among the participants

The files are stored using a unique fingerprint (the cryptographic hash of their content) which also solves the problem of data duplication across the network.

2.2 Product Functions

- The main function of the Software is to enable its Users to share their Photos in a Social Community and to view the photos shared by others in the community.
- The Users will be able to share their Photos by mentioning the location where Photo has been taken and a Topic describing (related to photo) the Photo.
- The Users will also be able to browse for already uploaded photos by searching for their topic of interest in the provided search bar.
- Deletion of photos from the Software can only be done by the User who have uploaded it.

2.3 User Classes and Characteristics

- This application is suitable for everyone who is interested in sharing their photos with the community and get photos based on their location and interest.

2.4 Operating Environment

- This Dapp will run on all operating Systems that have the following dependencies installed:
 - IPFS
 - Ethereum Virtual Machine
- The instructions for setting up the network are separately given in the **README.md** file of the code repository.

2.5 Design and Implementation Constraints

- **Front-End Constraints**
 - The front-end of the application uses Google Map API on which every users can see the photos that have been uploaded to Eth-Photo.
- **Back-End Constraints**
 - Backend uses IPFS,a distributed web technology,to store and retrieve images.Uses solidity, as programming language, to create contracts for the blockchain.

2.6 Dependencies

- **Python - Pillow Module**
 - It is a Python Imaging Library

Chapter 3

External Interface Requirements

3.1 User Interface

The user interface will be designed with simplicity and functionality in mind.

The front-end of the application uses Google Maps API, on which every users can see the photos that have been uploaded to the Software. If user goes to a specific location, he could see all the photos tagged with that location.

Apart from the location based photo tagging, a user can also browse the photo based on topics. Once a user clicks on a photo, the DApp also shows the location information tagged with that photo.

3.2 Software Interface

The system require properly configured version of an operating system (Ubuntu, Windows or Mac OS X) a browser installed (Chrome, Firefox). These computers must have the requirements installed as mentioned in **README.md** file.

3.3 Hardware Interface

The application can run on any hardware which supports any modern operating system and provides proper network access. The application requires 1.5GB-2GB memory to load. The program runs in a browser and writes information to the user's computer in non-root directories. The user's computer transfers and receives data packets from peers using basic networking protocols. All the information is stored in a distributed way across all peers using the application.

3.4 Communication Interfaces

The users are connected in a peer-to-peer network through wired or wireless network.

Chapter 4

System Features

EthPhoto will utilize the features of the blockchain based platform Ethereum. The following are the features of the software -

4.1 Map-based Frontend

4.1.1 Description and Priority

The frontend of the app will be using Google Maps API on which users can see the photos that have been uploaded to EthPhoto. If one goes to a specific location, user is able to see all the photos tagged with that location. It also contains a search bar in order to allow users to browse photos by their interested topic.

4.1.2 Stimulus/Response Sequences

User action : Browse through map

System response : Show thumbnail of the images uploaded from that place.

4.1.3 Functional Requirements

The user should be able to click on the thumbnail in order to view the originally uploaded image.

4.2 Geo-Tagging/Topic based Tagging

4.2.1 Description and Priority

Users can browse the pictures uploaded on the EthPhoto via its tagname. Tags should be given, when the photo is uploaded by the user.

4.2.2 Stimulus/Response Sequences

User action : Upload picture

System response : Ask for the topic

4.2.3 Functional Requirement

The user should see the list of entire images available on the network based on the tag he / she searches.

Chapter 5

Other Requirements

5.1 Performance Requirements

As Functions on the contracts are executed on all the connected nodes, the contract should be as less resource-intensive as possible. Apart from that, the synchronization problem is to be solved. Also, since the application is to be available to a large scale, the proper storing and accessing mechanism has to be used to minimize the time delays / latency.

5.2 Safety Requirements

The information of a an individual who uploaded the picture can't be accessed by another without his/her content.

5.3 Security Requirements

A person shouldn't be able Spam the network with the same image. Also he shouldn't be able to delete any photo which he has not uploaded.

5.4 Software Quality Attributes

5.4.1 Availability

Any user running the application will be automatically connected to an Ethereum network, then user gets the access to upload,delete,view the images.

5.4.2 Flexibility

The app can work over operating systems such as Linux, Windows, and Mac OS, with the only requirement being that the user has installed the required software for running Ethereum Network.

5.4.3 Interoperability

- The users of the apps can communicate with each other even if their systems consists of different platforms as long as they are connected to the ethereum network.
- System differences between the two app users should be hidden from each other.

5.4.4 Maintainability

As there is no central server involved, the system should efficiently discard the trash and keep the required information.

5.4.5 Security

All the details of a particular As the technology used in this Software is still in development phase, the one who are connected to the network must ensure that they are using the same version. user will be safe as long as he has the private key with him, which he/she can use to perform any transaction on any other platforms as well.

5.4.6 Reliability

The users will be able to perform all the functions such as browse, upload and delete images at any time.

5.4.7 Usability

The service should be easy enough to learn quickly. The user interface should be intuitive with minimal user setup.

Chapter 6

Analysis Models

6.1 Control Flow Diagrams (CFD)

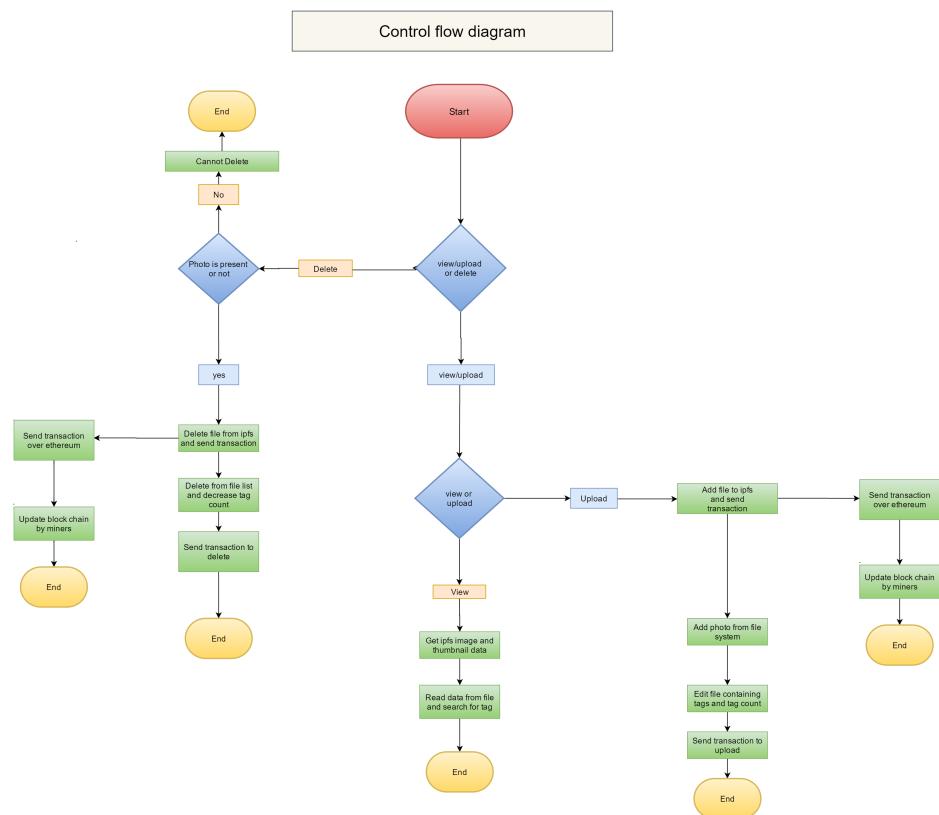


FIGURE 6.1: Control Flow for whole process

6.2 Data Flow Diagrams (DFD)

Level 0 Diagram for EthPhoto

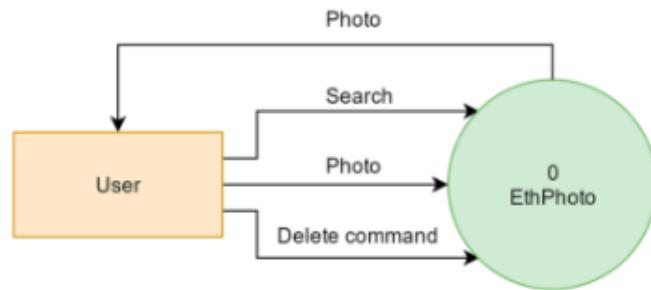


FIGURE 6.2: Level 0/Context Level diagram

Level 1 Diagram for EthPhoto

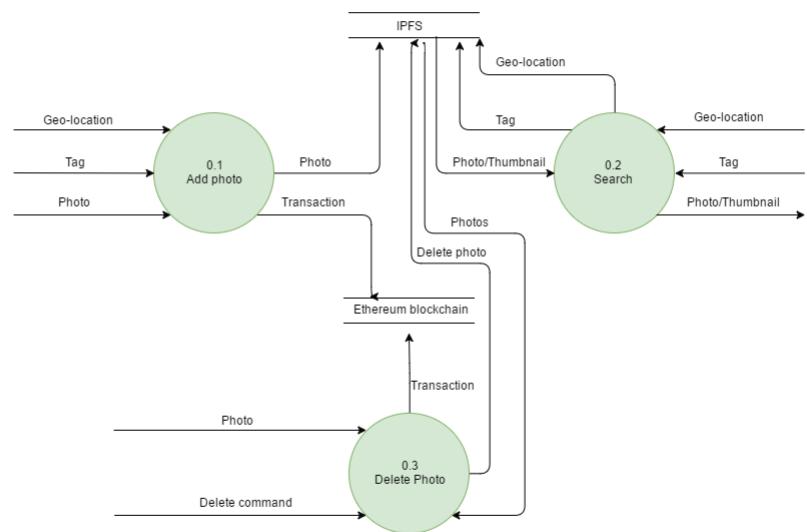


FIGURE 6.3: Level 1 diagram

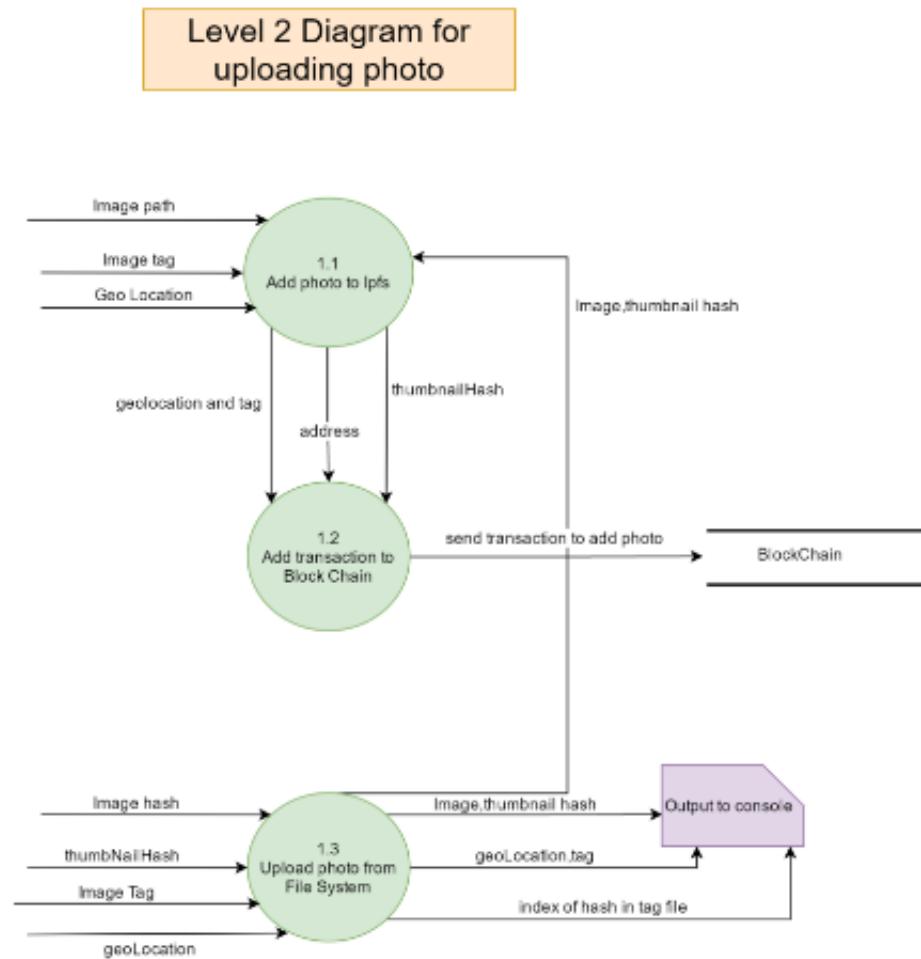


FIGURE 6.4: Level 2 Diagram for adding image

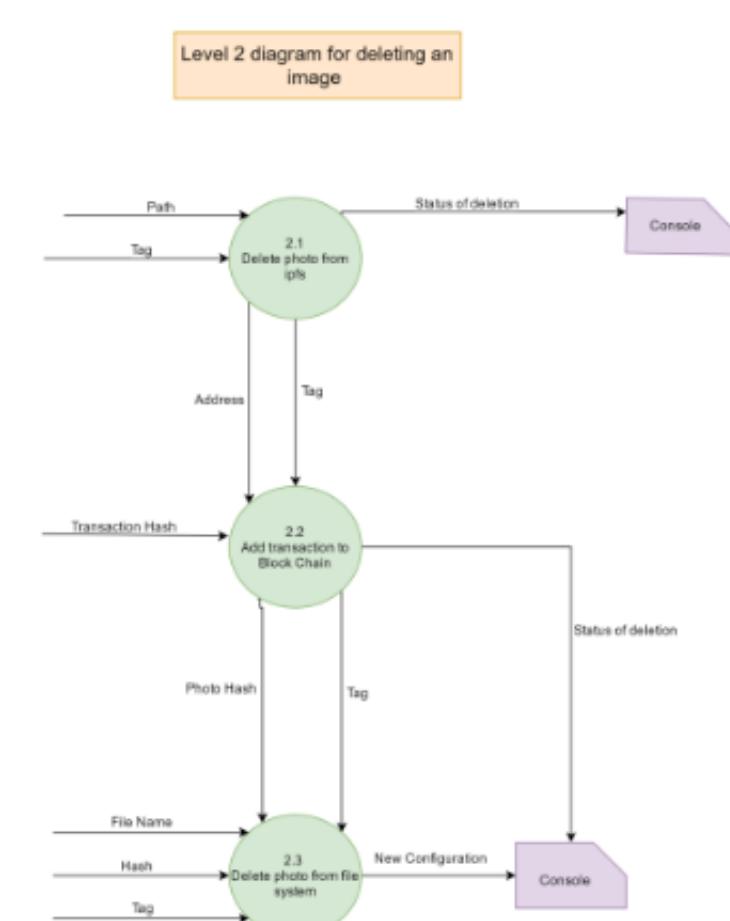


FIGURE 6.5: Level 2 diagram for deleting image

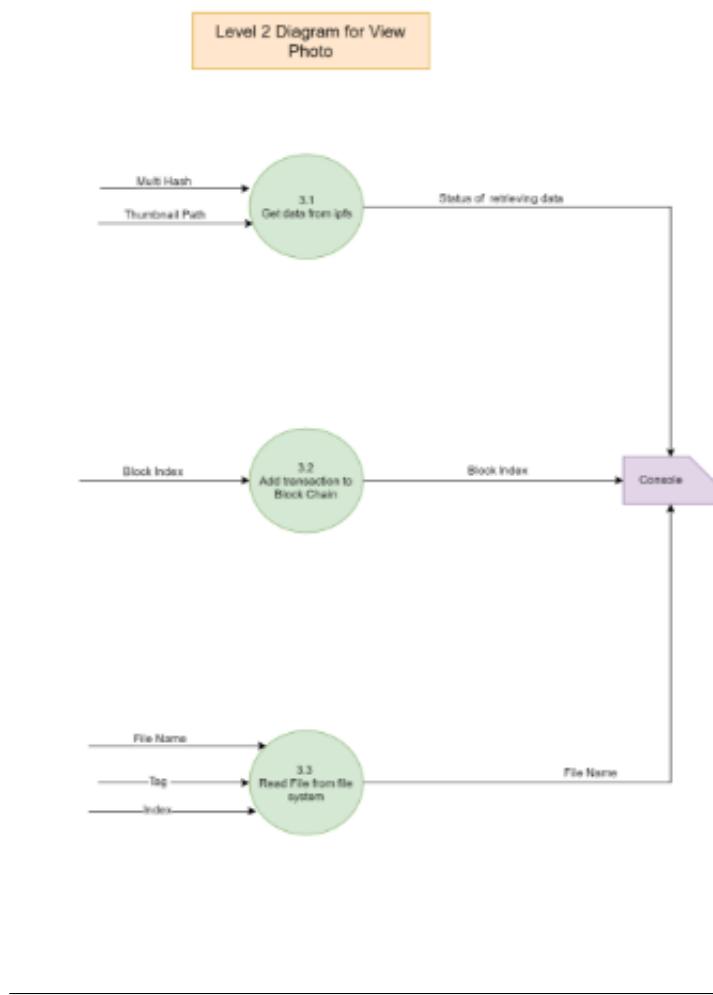


FIGURE 6.6: Level 2 diagram for viewing image

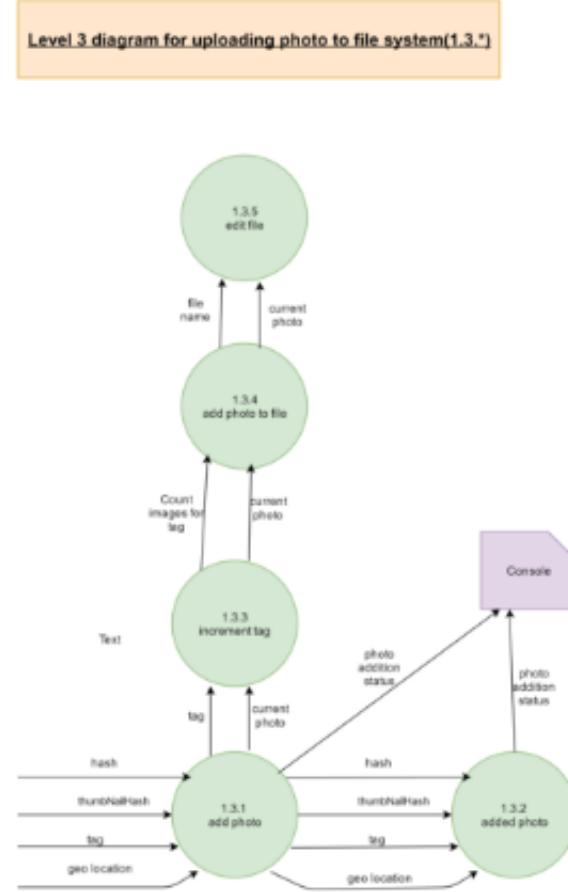


FIGURE 6.7: Level 3 diagram for adding image

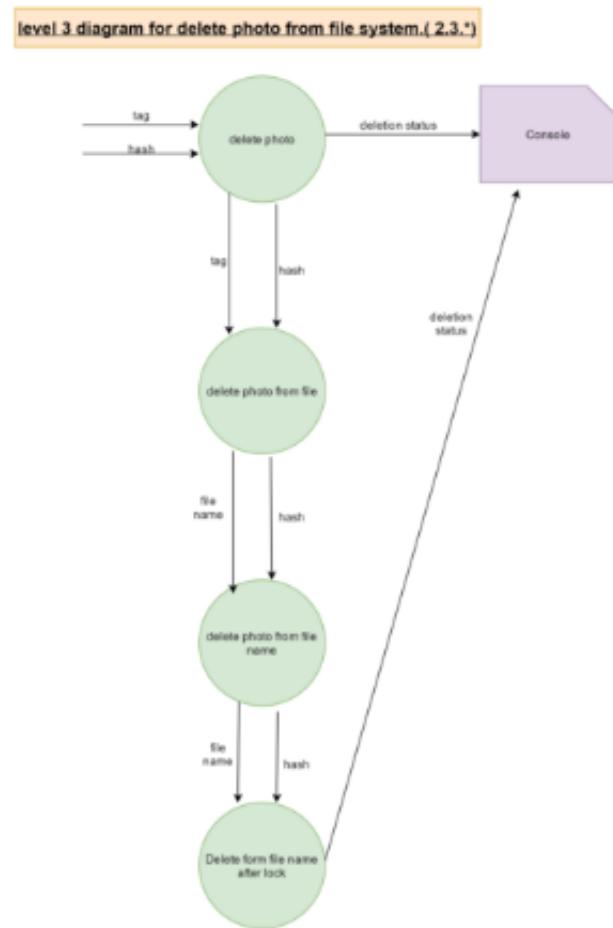


FIGURE 6.8: Level 3 diagram for deleting image

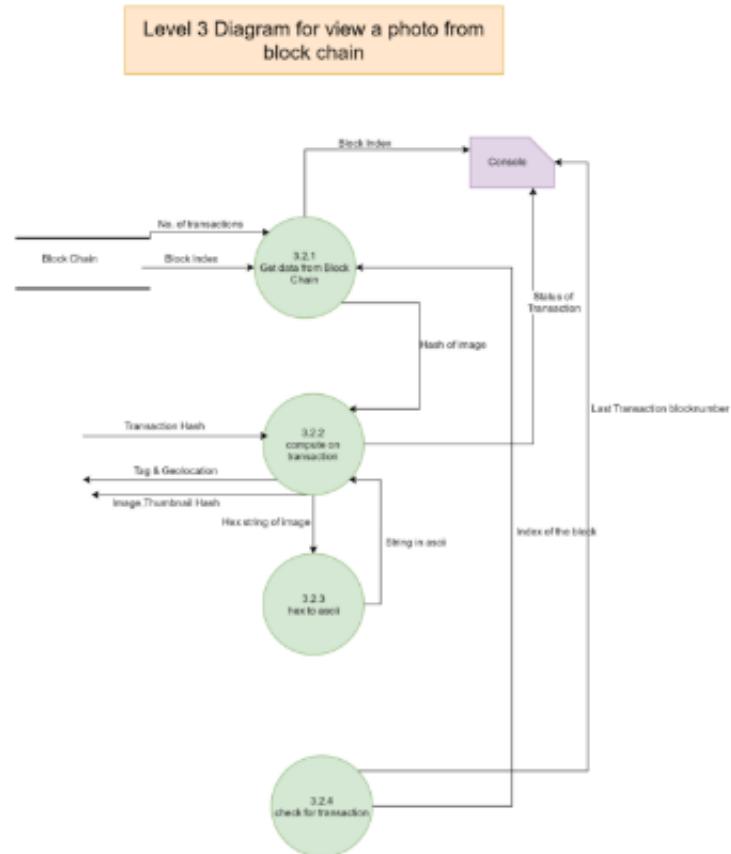


FIGURE 6.9: Level 3 diagram for viewing image

Chapter 7

Test Cases

7.1 Show Thumbnails

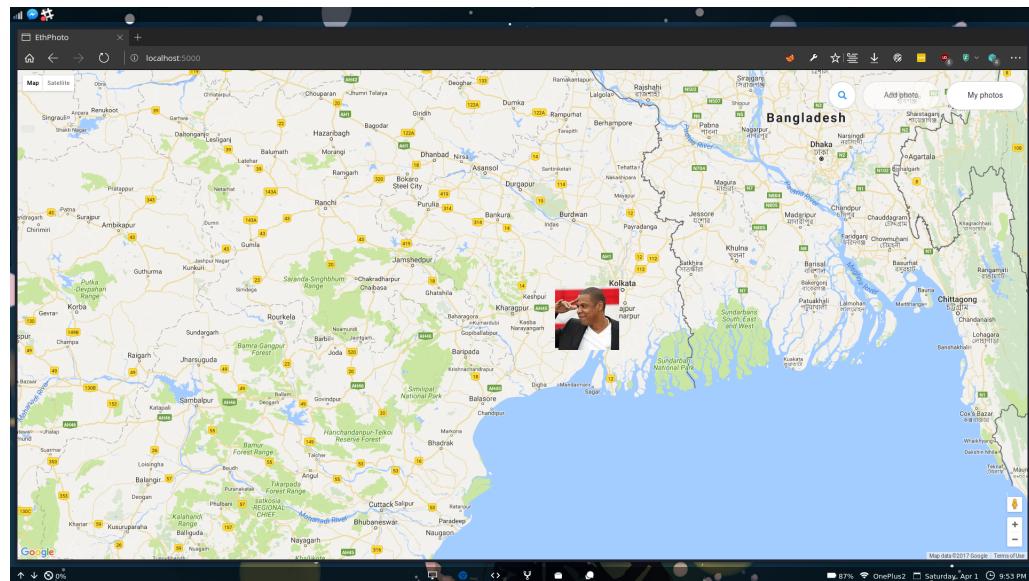


FIGURE 7.1: Image's Thumbnail on Map

7.2 Search Tags

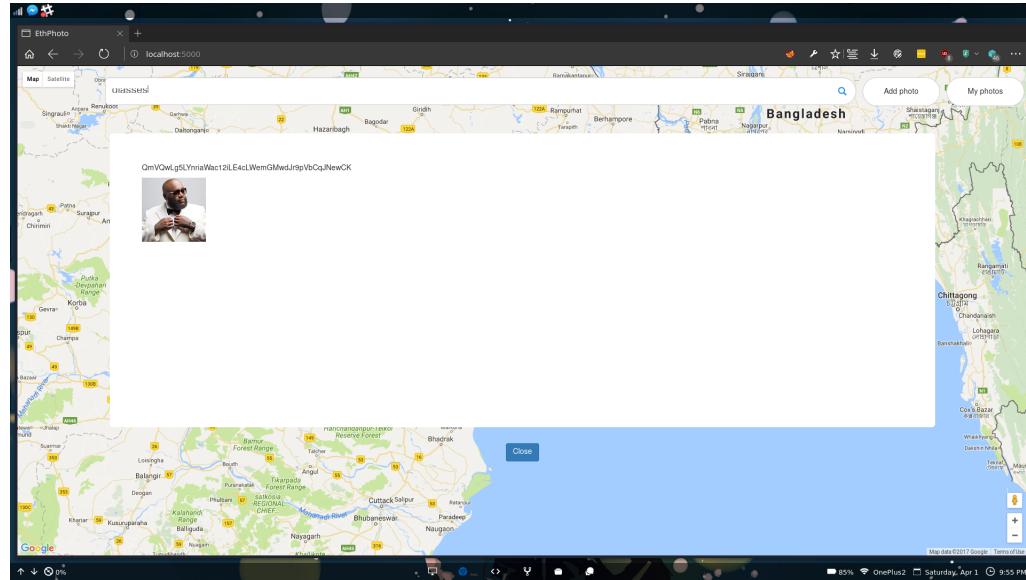


FIGURE 7.2: Search Tag Wise Images

7.3 Add Photo

7.3.1 Place Marker

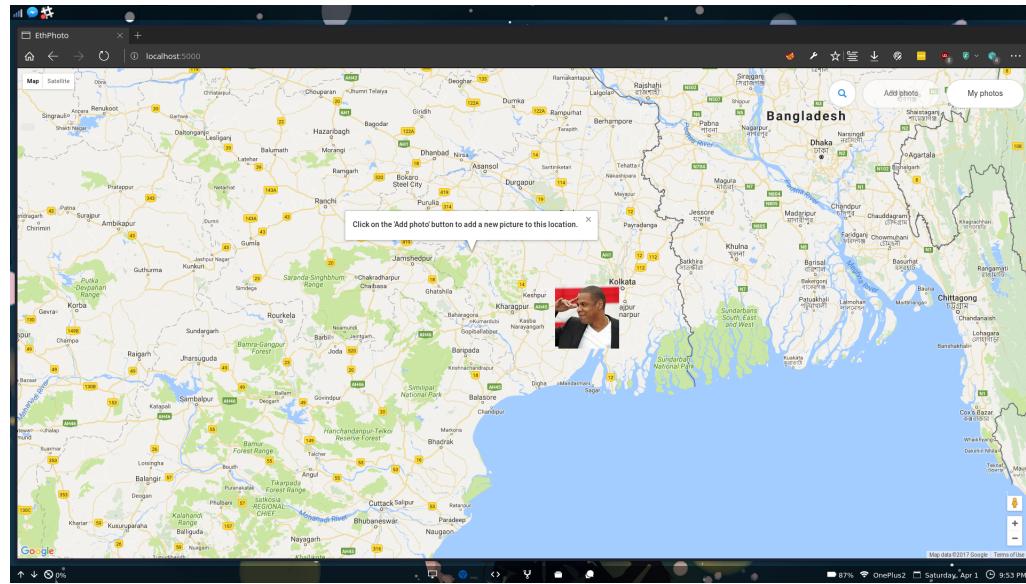


FIGURE 7.3: Place Marker on Location

7.3.2 Upload Image

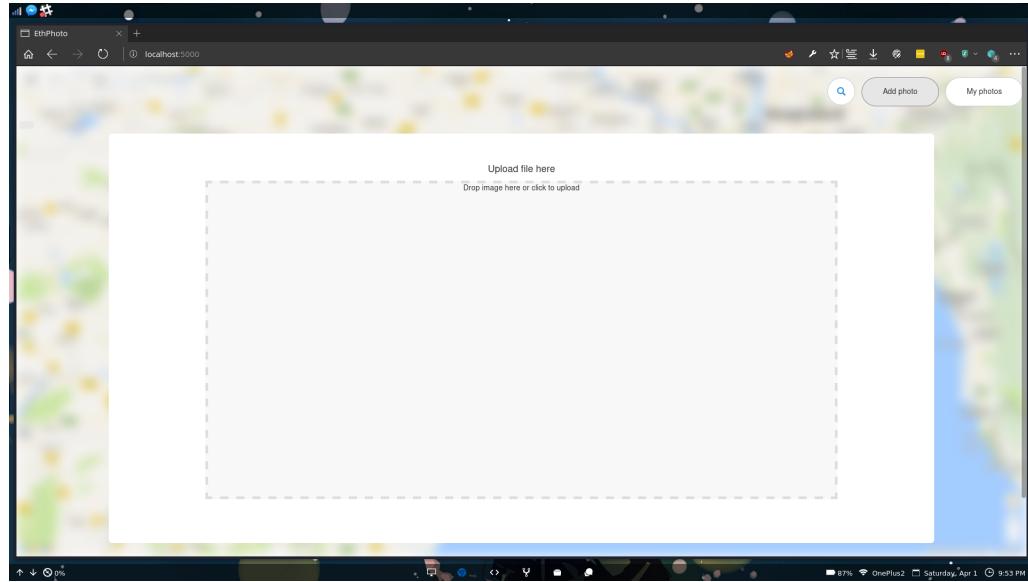


FIGURE 7.4: Upload an Image

7.3.3 After Uploading

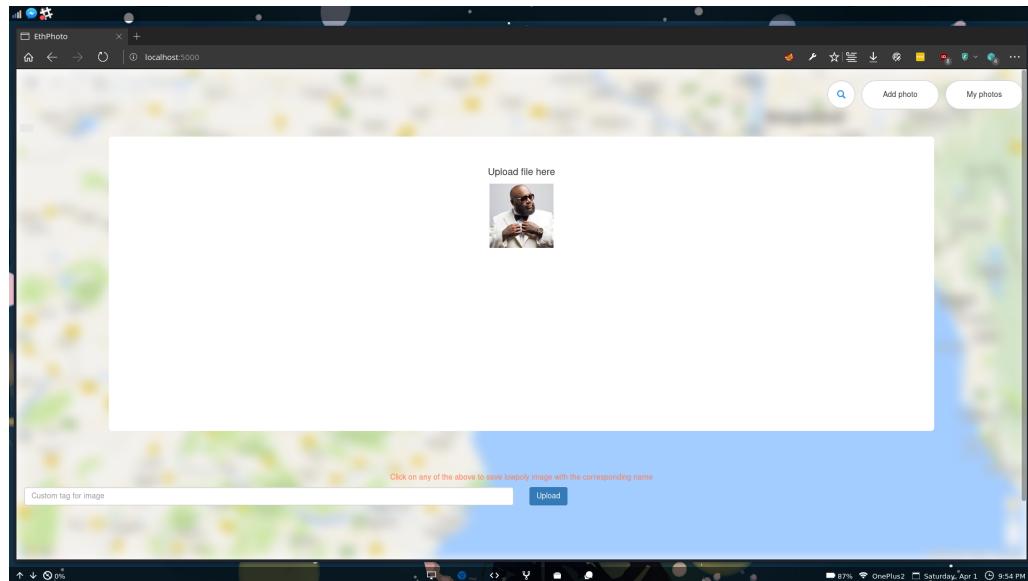


FIGURE 7.5: After Image Uploading

7.4 Delete Image

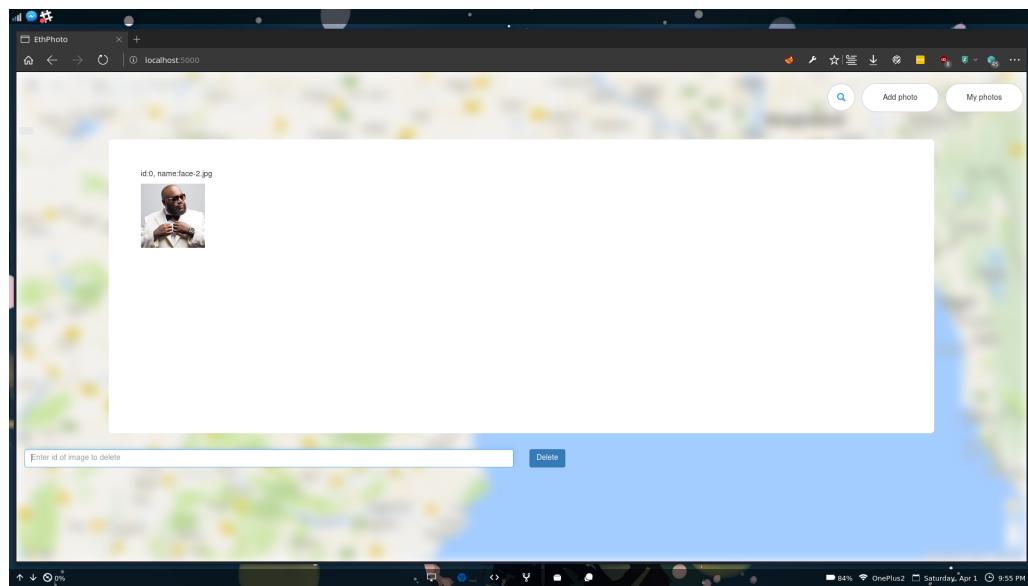


FIGURE 7.6: Delete an Image

Appendix A

Glossary

Ethereum : Ethereum is a blockchain based platform that helps in building decentralized apps , where there is no centralized server and multiple users participate in the process through the blockchain platform.

IPFS : IPFS is a Peer to Peer HyperMedia Protocol to make web faster, safer and more open.

DApp : Decentralised Application built using Ethereum Platform, based on Blockchain Technology.