



ABDUL KHALIQ  
01-235152-054  
FAISAL SALEEM  
01-235152-055

# Decentralized Trustless Data Access System

**Bachelor of Science in Information Technology**

Supervisor: Dr. M Muzammil

Department of Computer Science  
Bahria University, Islamabad

April 2019

# Abstract

The aim of our project is to switch the regular data Access systems to the Blockchain systems. The Blockchain system will seek to meet the need of Organizations and people who have some data in the different organizations. This system will secure the data of Data Owner and without his permission this data cannot be added to the Blockchain network. This system is built on the Hyperledger Fabric framework which provides the best private Blockchain. Also by developing this system it will improve the efficiency and effectiveness of the data access systems. On the other hand this system will provide the global access to the data owner. Because many organizations can join this Blockchain network and access the data. So there is no need to keep the manual record with him.

# Contents

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction	1
1.1.1 Why trustless	2
1.2 Objective	2
1.3 Problem Description	3
1.4 Project Scope	3
<b>2 Literature Review</b>	<b>4</b>
2.1 Literature Review	4
2.1.1 Blockchain	4
2.1.2 Types of Blockchain	5
2.2 Hyperledger Fabric	6
<b>3 Requirement Specifications</b>	<b>9</b>
3.1 Functional Requirement	9
3.1.1 FR 01:	9
3.1.2 FR 02:	10
3.1.3 FR 03:	10
3.1.4 FR 04:	10
3.1.5 FR 05:	11
3.1.6 FR 06:	11
3.1.7 FR 07:	12
3.1.8 FR 08:	12
3.1.9 FR 09:	13
3.1.10 FR 10:	13
3.1.11 FR 11:	14
3.2 Non-Functional Requirement:	14
3.2.1 NFR 01:	15
3.2.2 NFR 02:	15
3.2.3 NFR 03:	15
3.2.4 NFR 04:	15
3.2.5 NFR 05:	15
3.3 Use Cases of System:	15

<b>4</b>	<b>System Design</b>	<b>20</b>
4.1	Data Flow Diagram: . . . . .	22
4.1.1	Zero level DFD: . . . . .	22
4.1.2	Level 1 DFD: . . . . .	23
4.2	Architecture diagram: . . . . .	24
4.3	Sequence diagrams: . . . . .	25
4.3.1	Data generator Login sequence . . . . .	27
4.3.2	Data Generator View Record Sequence . . . . .	27
4.3.3	Data Generator record generation sequence . . . . .	28
4.3.4	Transactional data saving procedure sequence . . . . .	28
4.3.5	Data Owner Sign up sequence . . . . .	29
4.3.6	Data Owner Login sequence . . . . .	30
4.3.7	Data Owner view record sequence . . . . .	30
<b>5</b>	<b>System Implementation</b>	<b>31</b>
5.1	System Architecture . . . . .	31
5.2	Tools and techniques: . . . . .	31
5.2.1	Hyperledger Composer: . . . . .	31
5.2.2	Hyperledger Fabric: . . . . .	32
5.2.3	Eclipse: . . . . .	32
5.2.4	Postman: . . . . .	32
5.2.5	Java: . . . . .	32
5.2.6	REST API: . . . . .	33
5.2.7	GOLANG: . . . . .	33
5.2.8	Mechanisms and Consensus . . . . .	33
5.2.9	Spring BOOT: . . . . .	35
5.2.10	Dream Viewer: . . . . .	35
5.2.11	Gradle: . . . . .	35
5.2.12	NODE.js: . . . . .	35
5.2.13	Python: . . . . .	36
5.2.14	Hyperledger Fabric CA: . . . . .	36
5.2.15	CSS3: . . . . .	37
5.2.16	HTML5: . . . . .	37
5.2.17	Bootstrap: . . . . .	37
5.2.18	Java Script: . . . . .	37
5.3	Methodologies: . . . . .	38
5.4	Major Errors: . . . . .	40
5.5	Interface designing: . . . . .	40
5.5.1	Data Generator Registration: . . . . .	41
5.5.2	Data Owner Registration: . . . . .	41
5.5.3	Data Generator Login: . . . . .	42
5.5.4	Data Owner Login: . . . . .	43
5.5.5	Data Generator portal: . . . . .	43
5.5.6	Data Owner portal: . . . . .	43
5.5.7	Data Generator portal: . . . . .	43

<b>6</b>	<b>System Testing and Evaluation</b>	<b>45</b>
6.1	Performance Testing: . . . . .	45
6.2	Alpha Testing: . . . . .	46
6.3	White Box Testing: . . . . .	46
6.3.1	Bug Report: . . . . .	46
6.4	User Interface Testing: . . . . .	47
6.4.1	Data Generator Sign up . . . . .	48
6.4.2	Data Generator Log in . . . . .	48
6.4.3	Data Generator add transaction: . . . . .	49
6.4.4	Data Owner Sign up: . . . . .	49
6.4.5	Data Owner Log in: . . . . .	50
6.4.6	Data Owner Approval: . . . . .	50
<b>7</b>	<b>Conclusions</b>	<b>51</b>
7.1	Major Accomplishment: . . . . .	51
7.2	Future Enhancements: . . . . .	52
	<b>References</b>	<b>53</b>