

ABSTRACT

Cloud servers are used to utilize the distributed transactional database systems, entities are collaborated to form testament of authorization as that are justified by collections of certified credentials. These proofs and credentials may be evaluated and collected over extended time under the risk of having the underlying authorization protocol or the user credentials being in inconsistent states. Therefore, it is possible for Policy-Based Authorization Systems which results in unsafe decisions, that might threaten sensitive resources. The criticality of the problem is highlighted, we define the trusted transactions while dealing with proofs of authorization. Accordingly, we propose several increasingly valid levels of policy consistency constraints, and present different enforcement approaches to guarantee the trust worthiness of transactions executing on cloud servers. Two-Phase Validation Commit protocol is proposed as a solution. We finally analyze the different approaches presented using both analytical (systematic) evaluation (estimation) of the overheads and match to guide the decision maker to which approach to use.

Keywords

Cloud Servers, Database systems, Authorization, Protocols, Inconsistent states, Accuracy.

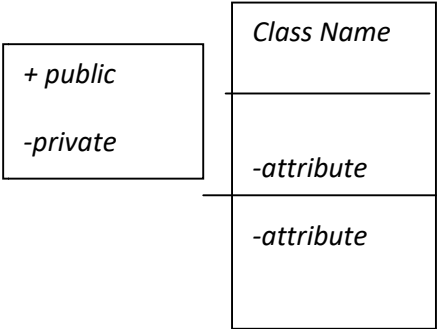
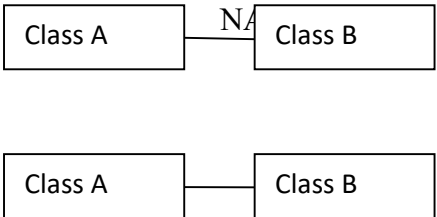
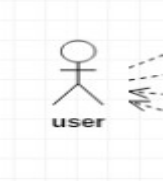
LIST OF FIGURES

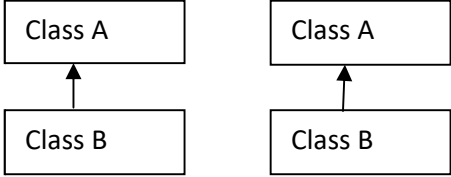
S.NO	FIGURE NO	NAME OF THE FIGURE	PAGE NO
1	3.1	System Architecture	13
2	3.2	User Interface Design	14
3	3.3	Quality of Service	14
4	3.4	Authorization Policies	15
5	3.5	Distributed Transactions	15
6	3.6	Certificate Authorities	16
7	4.1	Activity Diagram	19
8	4.2	Use Case Diagram	20
9	4.3(a)	Data Flow Diagrams(level-0)	21
10	4.3(b)	Data Flow Diagrams(level-1)	21
11	4.4	Sequence Diagram	22
12	4.5	Collaboration Diagram	23
13	4.6	Class Diagram	24
14	4.7	Entity-Relationship Diagram	25
15	4.8	Gantt chart	26
16	6.1	Login page-ScreenShot	45
17	6.2	Registration page-ScreenShot	45
18	6.3	Admin login page-Screenshot	46
19	6.4	User file search page-Screenshot	46
20	6.5	login database-Screenshot	46
21	6.6	Admin database-Screenshot	47
22	6.7	Life database-Screenshot	47
23	6.8	Home database-Screenshot	47
24	6.9	Medical database-Screenshot	48

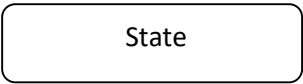
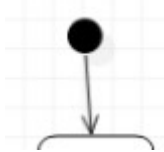

LIST OF TABLES

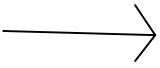
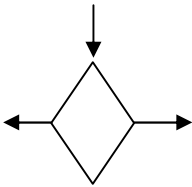
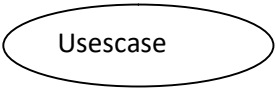
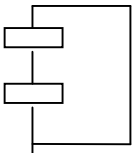
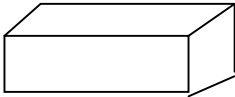
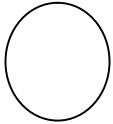
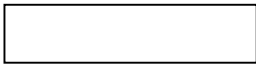

S.NO	TABLE NO	NAME OF THE TABLE	PAGE NO
1	5.1	Test Case for Login	42
2	5.2	Test Case for Registration	43
3	5.3	Test Case for Admin	44

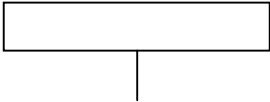
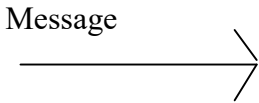
LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	 <p>The diagram shows a UML class box. On the left side, there are two compartments: the top one contains <i>+ public</i> and the bottom one contains <i>-private</i>. The main body of the box is divided into three horizontal sections. The top section is labeled <i>Class Name</i>. The bottom two sections each contain <i>-attribute</i>.</p>	Represents a collection of similar entities grouped together.
2.	Association	 <p>The diagram shows two UML class boxes, 'Class A' and 'Class B', connected by a solid line. The top diagram has the label 'NA' on the association line. The bottom diagram shows a simple association line without a label.</p>	Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor	 <p>The diagram shows a UML actor, represented by a stick figure. Below the figure is the label 'user'.</p>	It aggregates several classes into a single classes.

4	Aggregation		Interaction between the system and external environment

5.	Relation (uses)	uses	Used for additional process communication.
6.	Relation (extends)	<u>extends</u> →	Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication	_____	Communication between various use cases.
8.	State		State of the processs.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object

11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Use case		Interact ion between the system and external environment.
14.	Component		Represents physical modules which is a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or acion.
17.	External entity		Represents external entities such as keyboard,sensors, etc.
18.	Transition		Represents communication

			that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message		Represents the message exchanged.

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	EXPANSION
1.	DB	DataBase
2.	JVM	Java Virtual Machine
3.	JSP	Java Server Page
4.	CB	Collective Behavior
5.	SD	Social Dimension
6.	JRE	Java Runtime Environment
7.	SSD	Sparse Social Dimension
8.	LGP	Line Graph Partition

TABLE OF CONTENTS

TITLE

COLLEGE CERTIFICATE

ACKNOWLEDGEMENT

DECLARATION

ABSTRACT v

LIST OF FIGURES..... vi

LIST OF TABLES..... vii

LIST OF SYMBOLS..... vii

LIST OF ABBREVIATIONS..... x

1.INTRODUCTION 1-2

1.1 General1

1.2 Objective1

1.3 Existing System1

1.4 Proposed System2

2.LITERATURE REVIEW 3-5

3.DESIGN 6-16

3.1 System Requirements6

3.2 Software Description7

3.2 System Design13

3.4 Modules14

4. ANALYSIS 17-23

4.1 Introduction.....17

4.2 UML Diagram18

4.3 Conclusion26

5.IMPLEMENTATION AND TESTING 27-44

5.1 General	27
5.2 Coding	27
5.3 Testing	40
6.RESULTS	45-48
6.1 General.....	45
6.2 Various snapshots	45
7. CONCLUSION	49
7.1 General	49
7.2 Application	49
7.3 Future Enhancements	49
8.REFERENCES	50-51