**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 | 15 |
| **2** | 3.2 | User Interface Design | 16 |
| **3** | 3.3 | Quality of Service | 17 |
| **4** | 3.4 | Authorization Policies | 17 |
| **5** | 3.5 | Distributed Transactions | 18 |
| **6** | 3.6 | Certificate Authorities | 18 |
| **7** | 4.1 | Activity Diagram | 22 |
| **8** | 4.2 | Use Case Diagram | 23 |
| **9** | 4.3(a) | Data Flow Diagrams(level-0) | 24 |
| **10** | 4.3(b) | Data Flow Diagrams(level-1) | 24 |
| **11** | 4.4 | Sequence Diagram | 25 |
| **12** | 4.5 | Collaboration Diagram | 26 |
| **13** | 4.6 | Class Diagram | 27 |
| **14** | 4.7 | Entity-Relationship Diagram | 28 |
| **15** | 4.8 | Gantt chart | 29 |
| **16** | 6.1 | Login page-ScreenShot | 49 |
| **17** | 6.2 | Registration page-ScreenShot | 49 |
| **18** | 6.3 | Admin login page-Screenshot | 50 |
| **19** | 6.4 | User file search page-Screenshot | 50 |
| **20** | 6.5 | login database-Screenshot | 50 |
| **21** | 6.6 | Admin database-Screenshot | 51 |
| **22** | 6.7 | Life database-Screenshot | 51 |
| **23** | 6.8 | Home database-Screenshot | 51 |
| **24** | 6.9 | Medical database-Screenshot | 52 |

**LIST OF TABLEs**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **TABLE NO** | **NAME OF THE TABLE** | **PAGE NO** |
| **1** | 5.1 | Test Case for Login | 46 |
| **2** | 5.2 | Test Case for Registration | 47 |
| **3** | 5.3 | Test Case for Admin | 48 |

**LIST OF SymbolS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO**  Class Name  -attribute  -attribute  *+operation*  *+operation*  *+operation* | **NOTATION**  **NAME** | **NOTATION** | **DESCRIPTION** |
| **1** | Class | + public  -private  *# protected* | Represents a collection of similar entities grouped together. |
| **2** | Association | Class B  Class A  name  Class A  Class B | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| **3** | Actor | Capture70.PNG | It aggregates several classes into a single classes. |
| **4** | Aggregation | Class B  Class B  Class A  Class A | Interaction between the system and external environment |
| **5** | Relation  (uses) | uses | Used for additional process communication. |
| **6** | Relation  (extends) | extends | 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 |  | State of the process. |
| **9** | Initial State | Capture60.PNG | Initial state of the object |
| **10** | Final state | Capture50.PNG | 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 ABBERIVATIONS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVATION** | **EXPANSION** |
| **1.** | DB | Data Base |
| **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**

Contents Page.no

TITLE

COLLEGE CERTIFICATE

ACKNOWLEDGEMENT

DECLARATION

ABSTRACTv

LIST OF FIGURESvi

LIST OF TABLESvii

LIST OF SYMBOLSviii

LIST OF ABBERIVATIONSxi

1.INTRODUCTION1

1.1 General1

1.2 Objective1

1.3 Existing System1

1.4 Proposed System2

2.LITERATURE REVIEW 3

3.DESIGN 7

3.1 System Requirements 7

3.2 Software Description 8

3.3 System Design 14

3.4 Modules16

**4. ANALYSIS20**

**4.1 Introduction20**

**4.2 UML Diagram 22**

4.3 Conclusion 29

5.IMPLEMENTATION AND TESTING 30

5.1 General 31

5.2 Coding 31

5.3 Testing 44

6.RESULTS 49

6.1 General49

6.2 Various snapshots49

7. CONCLUSION53

7.1 Conclusion53

7.2 Application 53

7.3 Future Enhancements 53

8.REFERENCES 55