Software Design Description Report

SAFECLOUD

01/03/2015

Samet AYTAÇ
Daren TUZİ
Aylin MERT

PREFACE

This document contains the software design description for SafeCloud.

This document is based on IEEE Standards. For this purpose “IEEE Standard for Information Technology – Systems Design – Software Design Descriptions – IEEE Std 1016™-2009” document was taken as reference.

This software design document involves detailed and complete definitions and specifications that can guide developers throughout implementation of SafeCloud.

First section is an overview for the document and the second section includes all the necessary definitions for the system.In third chapter conceptual model for software design descriptions explained. In fourth chapter design descriptions are provided and finally in fifth chapter all necessary viewpoints that are context, composition, logical, information, patterns use, interface, interaction, state dynamics and resource viewpoints to develop this project are explained.

TABLE OF CONTENTS

PREFACE 1
1 Overview 5
1.1 Scope 5
1.2 Purpose 5
1.3 Intended audience 6
1.4 References 6
2 Definition 6
3 Conceptual Model for Software Design Descriptions 8
3.1 Software Design in Context 8
3.2 Software Design Descriptions within the Life Cycle 8
3.2.1 Influences on SDD Preparation 8
3.2.2 Influences on Software Life Cycle Products 8
3.2.3 Design Verification and Design Role in Validation 9
4 Design description information content 10
4.1 Introduction 10
4.2. SDD Identification 10
4.3 Design Stakeholders and Their Concerns 10
4.4 Design Views 10
4.5 Design Viewpoints 11
4.5.1 Context Viewpoint 11
4.5.2 Composition Viewpoint 11
4.5.3 Logical Viewpoint 12
4.5.4 Information Viewpoint 12
4.5.5 Interface Viewpoint 12
4.5.6 Interaction Viewpoint 12
4.5.7 State Dynamics Viewpoint 12
4.6 Design Elements 12

4.7 Design Overlays 12
4.8 Design Rationale 12
4.9 Design languages 13
5 Design Viewpoints 13
5.1 Introduction 13
5.2 Context Viewpoints 14
5.2.1 Design Concerns 15
5.2.2 Design Elements 15
5.3 Composition Viewpoint 22
5.3.1 Design Concerns 22
5.3.2 Design Elements 22
5.4 Logical Viewpoint 23
5.4.1 Design Concerns 23
5.4.2 Design elements 24
5.5 Dependency viewpoint 28
5.5.1 Design concerns 28
5.5.2 Design elements 28
5.6 Information viewpoint 29
5.7 Interface viewpoint 30
5.7.1 Design concerns 30
5.7.2 Design elements 30
5.8 Structure viewpoint 32
5.9 Interaction viewpoint 33
5.9.1 Design concerns 33
5.9.2 Design elements 33
5.10 State dynamics viewpoint 39
5.10.1 Design concerns 39
5.10.2 Design elements 40

1 OVERVIEW

This document describes the conceptual design of the SafeCloud Project according to the document guidelines presented in the IEEE 1016-2009 Recommended Practice for Software Design Descriptions (SDD). The software application is a web-based software application project which is cloud storage system with high level security.

1.1 SCOPE

SafeCloud is a product built having in mind open source code and high security encryption. The system is intended to have minimalistic and hassle free user interfaces at the same time provide state of the art file encryption when a file is uploaded. The data containers will be build in a tree structure to allow different levels of access and groups to match real life hierarchy of the university staff or business employees.

1.2 PURPOSE

This document is aiming to specify the features and requirements of the product SafeCloud. It will also supply an interface for the program, which will give an idea about what the end product will look like.

Moreover, processes during the implementation will be explained meticulously. Thus, reader of this SDD report will at least gain an overall description of the project. Designers and developers will take advantage of this detailed explanation because, in this way,

solutions for the specific problems will be revealed throughout the document.

1.3 INTENDED AUDIENCE

The Software Description document is intended for:

- Developers: by using the SDD, developers can easily understand the software and it helps developers to improve the current project features or add new features to existing system.
- Testers: by using the SDD, testers can find some bugs for their testing strategy. Instead of searching whole software program, testers can examine SDD.
- Users: by using the SDD, users can easily understand how to use the software. In this project our users can be anyone who uses the SafeCloud.

1.4 REFERENCES

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2 DEFINITION

Term	Definition
Database	Collection of all the information monitored by this system.
UML Diagram	Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of object-oriented software engineering.
Software Requirements Specification (SRS)	A complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software.
SDD	Software Design Description which is the complete description of the design of system.
IEEE	Institute of Electrical and Electronics Engineers
Use Case Diagram	It represents user's interaction with the system.

State Diagram	It is used to define the behavior of the system
Activity Diagram	Activity diagrams are graphical representations of workflows of stepwise activities and actions.
Class Diagram	It describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among objects.

3 CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTIONS

3.1 SOFTWARE DESIGN IN CONTEXT

In SafeCloud project, object oriented approach will be used as a design method. Hence, it will be easier to implement the project and add possible future features. Furthermore multi-layered system architecture will be used. Layers will help modularity and adaptability of the software. With object oriented design and multi-layered architecture, portability and integrality between components will be improved.

3.2 SOFTWARE DESIGN DESCRIPTIONS WITHIN THE LIFE CYCLE

3.2.1 INFLUENCES ON SDD PREPARATION

The key software life cycle product that drives this software design is the SRS we have prepared. All the details and requirements are taken from there in order to prepare this document.

3.2.2 INFLUENCES ON SOFTWARE LIFE CYCLE PRODUCTS

This SDD influences the content of SRS of this project. It also has influences on the whole implementation phase of this system. More than that, the test documentation and test plans of the system are also influenced by the SDD. In addition, the contents of SDD is taken into consideration by the developers in order to develop test cases and test procedures.

3.2.3 DESIGN VERIFICATION AND DESIGN ROLE IN VALIDATION

Test cases will be prepared after document phase. Verification of the software will be tested with these test cases and all parts will be evaluated. Success of the software system will be determined with test cases. After the results, validation of the software will be checked if requirements of the system are fulfilled or not.

.

4 DESIGN DESCRIPTION INFORMATION CONTENT

4.1 INTRODUCTION

Software Design Description of SafeCloud project identifies how this system will be implemented and designed. SafeCloud project has a modular object-oriented structure. Furthermore a qualified interface will be designed in a way that system will represent events of the simulation successfully.

The contents are also going to be explained in this section are as follows:

- Identification of the SDD,
- Identified design stakeholders,
- Identified design concerns,
- Selected design viewpoints, each with type definitions of its allowed design elements and design languages,
- Design views,
- Design overlays,
- Design rationale

4.2. SDD IDENTIFICATION

This document is a first version of System Design Description for this project. This SDD report prepared based on IEEE 1016-2009 standards. Visual Paradigm 12.0 is used for drawing diagrams. Project authorship, organization of the project team and date of the report are given in cover page of SDD. In the first section an overview of SDD is given. Scope of the SDD report refers to the section 1.1, Purpose of the SDD report refers to section 1.2 and Intended Audience of this document refers to section 1.3. For design conceptual model for software design descriptions refer to the section 3. Lastly, for the design viewpoints including context, composition, logical, information, patterns use, interface, interaction, state dynamics and resource viewpoints refer to the section 5.

4.3 DESIGN STAKEHOLDERS AND THEIR CONCERNS

Design stakeholders of SafeCloud are the developer team of the project.

Their major concerns are:

- The implementation should be safe, secure, maintainability and open to future changes.
- The interface shall be easy to read and use.
- The desired results should be obtained from the developed system.

4.4 DESIGN VIEWS

Representing the diagrams of view, UML is used. Design views of this SDD are design rational, contextual, composition, interface, logical and interaction views. These design views are governed by design viewpoints that are explained in chapter 5.

4.5 DESIGN VIEWPOINTS

This section will be used to give brief outline on the design viewpoints which are used in section 5. A design viewpoint addresses a different perspective to be focused on to effectively encompass requirements that have been previously created and to identify the users as to which these requirements are relevant.

4.5.1 CONTEXT VIEWPOINT

The context viewpoint depicts services provided by a design subject with reference to an explicit context. That context is defined by reference to actors that include users and other stakeholders, which interact with the design subject in its environment.

4.5.2 COMPOSITION VIEWPOINT

The Composition viewpoint describes the way the design subject is (recursively) structured into constituent parts and establishes the roles of those parts.

4.5.3 LOGICAL VIEWPOINT

The purpose of the Logical viewpoint is to elaborate existing and designed types and their implementations as classes and interfaces with their structural static relationships. This view point also uses examples of instances of types in outlining design ideas.

4.5.4 INFORMATION VIEWPOINT

The Information viewpoint is applicable when there is a substantial persistent data content expected with the design subject.

4.5.5 INTERFACE VIEWPOINT

The Interface viewpoint provides information designers, programmers, and testers the means to know how to correctly use the services provided by a design subject. This description includes the details of external and internal interfaces not provided in the SRS. This viewpoint consists of a set of interface specifications for each entity.

4.5.6 INTERACTION VIEWPOINT

The Interaction viewpoint defines strategies for interaction among entities, regarding why,where, how, and at what level actions occur.

4.5.7 STATE DYNAMICS VIEWPOINT

Reactive systems and systems whose internal behavior is of interest use this viewpoint.

4.6 DESIGN ELEMENTS

The main design elements are entities, attributes and some other member associated with communication and relations between modules and user of our project. These main design elements are defined inside the related viewpoints in detail in chapter 5.

4.7 DESIGN OVERLAYS

Design Overlays usually used to add information to the existing views. This concept will be explained clearly when necessary in the design viewpoints section.

4.8 DESIGN RATIONALE

In this project, design choices are made according to performance concerns and integrality of the system. System has to be designed in a way that future models and features can be added and current models can be changed and updated independently. Stakeholders may have and request further requirements, therefore system parts have to be modular. Developers of the system has to document development process and use comments in their code frequently, so that in the future other developers may understand code and the structure of the system.

We use HTML and Javascript for client side. In server side, to upload and download files we use PHP. We use Firebase for state syncronization. In this way, if user upload a file to system,file will be appear in container without reflesh website. We use Zend framework for encryption/decryption. We choose AES as encryption algorithm.

4.9 DESIGN LANGUAGES

In this project, Unified Modeling Language (UML) is selected as a part of design viewpoint and it will be used for clarifying design viewpoints.

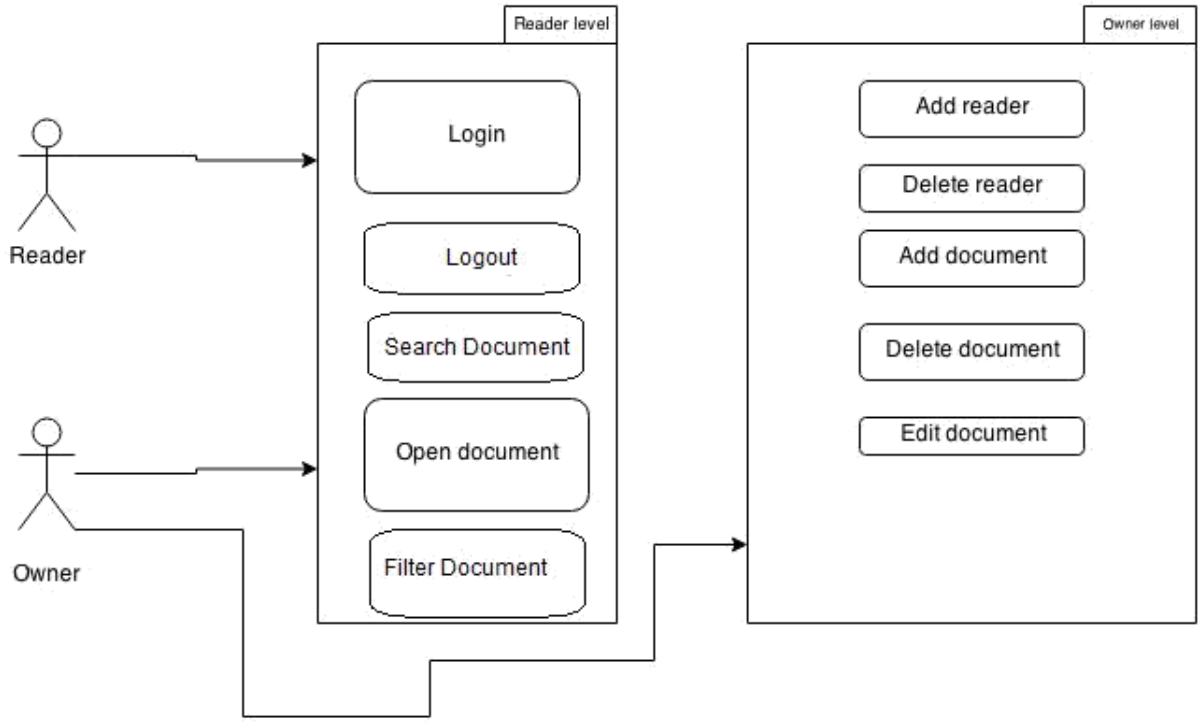
5 DESIGN VIEWPOINTS

5.1 INTRODUCTION

In this chapter, the viewpoints of the SafeCloud are explained in detail. During this section, UML diagrams will be used to increase understandability.

5.2 CONTEXT VIEWPOINTS

SafeCloud software context viewpoint shows the functions provided by a design. There are two major parts which are reader level and owner level features in the system. Each part has sub-parts too. The context is defined by the elements that interact with the software like reader and owner.



5.2.1 DESIGN CONCERNS

There are two main categories concerning to SafeCloud. Each of these categories are related with their actors. The main service categories are reader and owner actions.

Readers are people who has permission from owner of the container to use the SafeCloud. They are going to use reader level features of the final product. Owners are controllers of the SafeCloud who can use reader and owner level features.

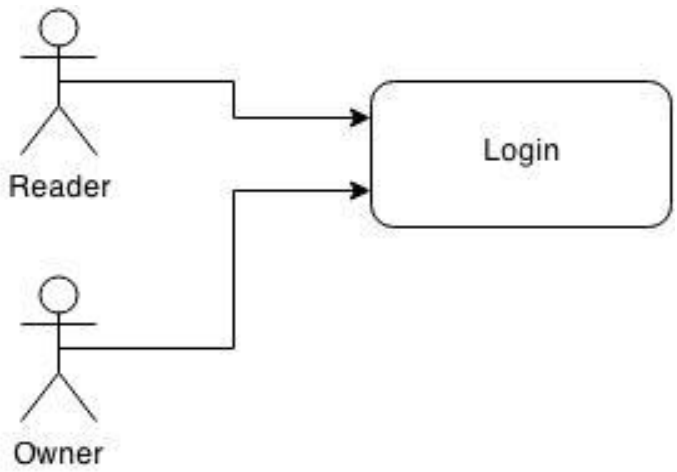
5.2.2 DESIGN ELEMENTS

5.2.2.1 READER LEVEL FEATURES

5.2.2.1.1 LOGIN

Allows readers and owner to login to the system.

Diagram:



Normal Flow of Events:

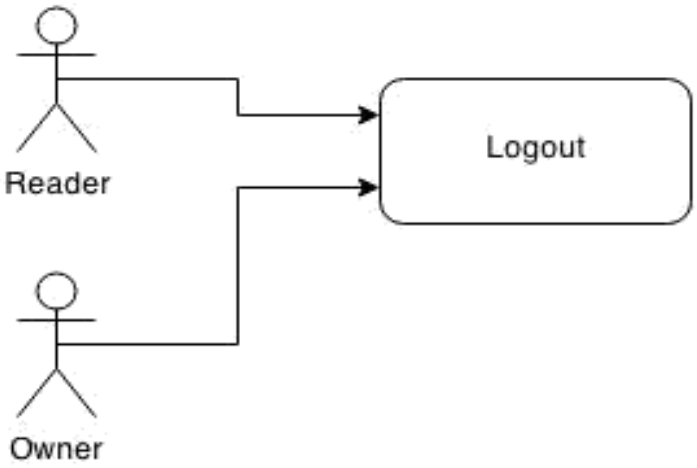
1. User enters the SafeCloud's page.

2. In the first page,by entering username and password, user can enter container.

5.2.2.1.2 LOGOUT

Allows readers and owner to logout from the system.

Diagram:



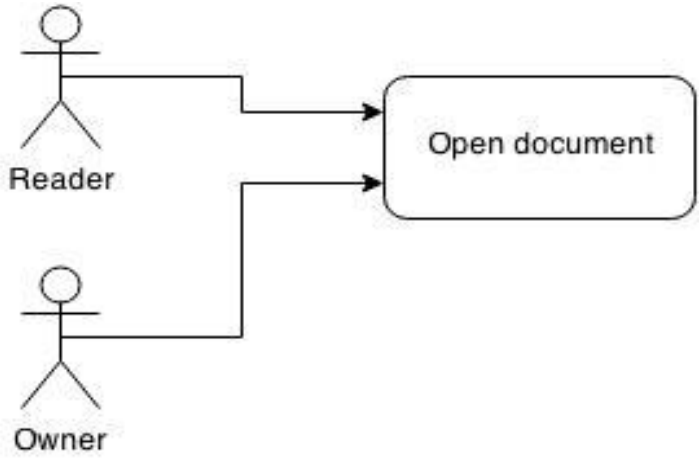
Normal Flow of Events:

1. User can click logout button to logout from the system.

5.2.2.1.3 OPEN DOCUMENT

Allows users to decrypt file with their keys and open document.

Diagram:



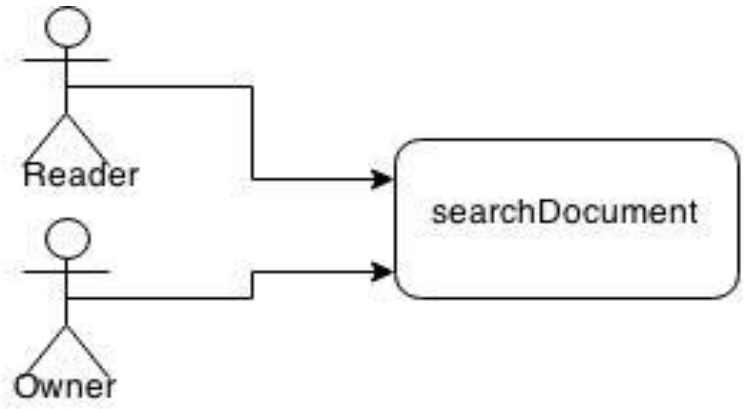
Normal Flow of Events:

1. User should select any document on the container.
2. Document will be decrypted and opened by user.

5.2.2.1.4 SEARCH DOCUMENT

Allows all types of user to search documents after entering keywords.

Diagram:



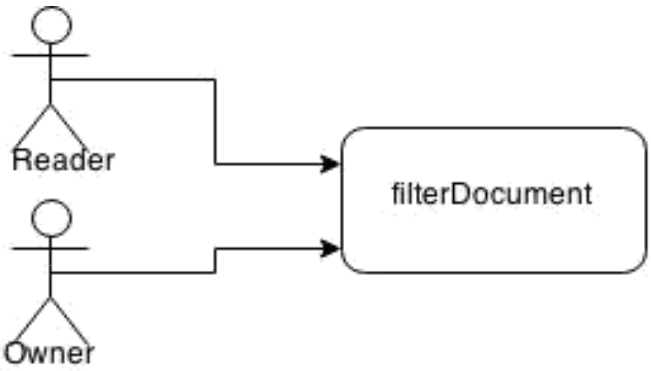
Normal Flow of Events:

1. User should enter keywords.
2. User should press search button after setting keywords.
3. System lists documents which have selected features.

5.2.2.1.5 FILTER DOCUMENT

Allows all types of user to filter documents after selecting date.

Diagram:



Normal Flow of Events:

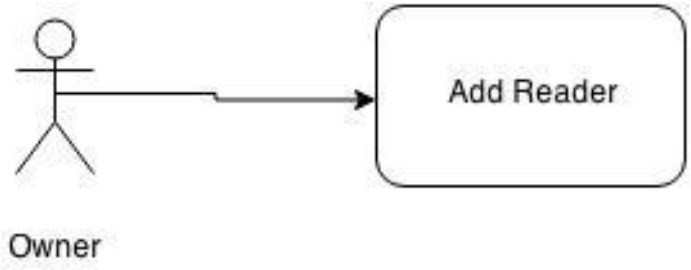
1: User should enter a date.
2: Documents will be listed according to filter.

5.2.2.2 OWNER LEVEL FEATURES

5.2.2.2.1 ADD READER

Allows owner to add reader.

Diagram:



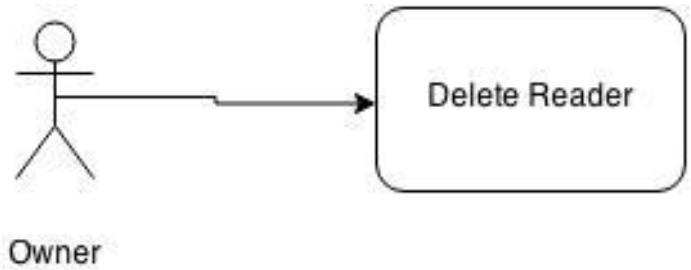
Normal Flow of Events:

1. Owner should add a reader who can read documents.
2. System will generate a key for user.

5.2.2.2.2 DELETE READER

Allows owner to delete reader.

Diagram:



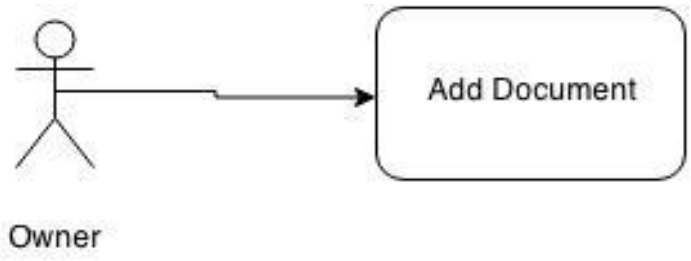
Normal Flow of Events:

1. Owner should select a reader who will be removed.
2. System will remove user.

5.2.2.2.3 ADD DOCUMENT

Allow owner to add new document.

Diagram:



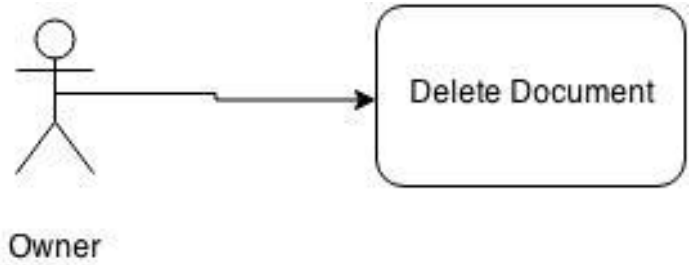
Normal Flow of Events:

1. Owner should add a document to container.
2. System will encrypt document and upload to container.

5.2.2.2.4 DELETE DOCUMENT

Allow owner to delete any document from container.

Diagram:



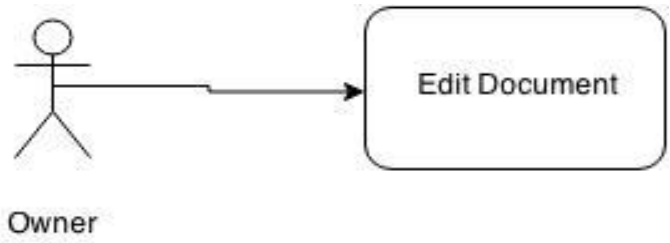
Normal Flow of Events:

1. Owner should select a document which will be removed from container.
2. System will remove this document from container.

5.2.2.2.5 EDIT DOCUMENT

Allow owner to edit any document from container.

Diagram:



Normal Flow of Events:

1. Owner should select a document which will be edited.
2. Owner can edit document on container.

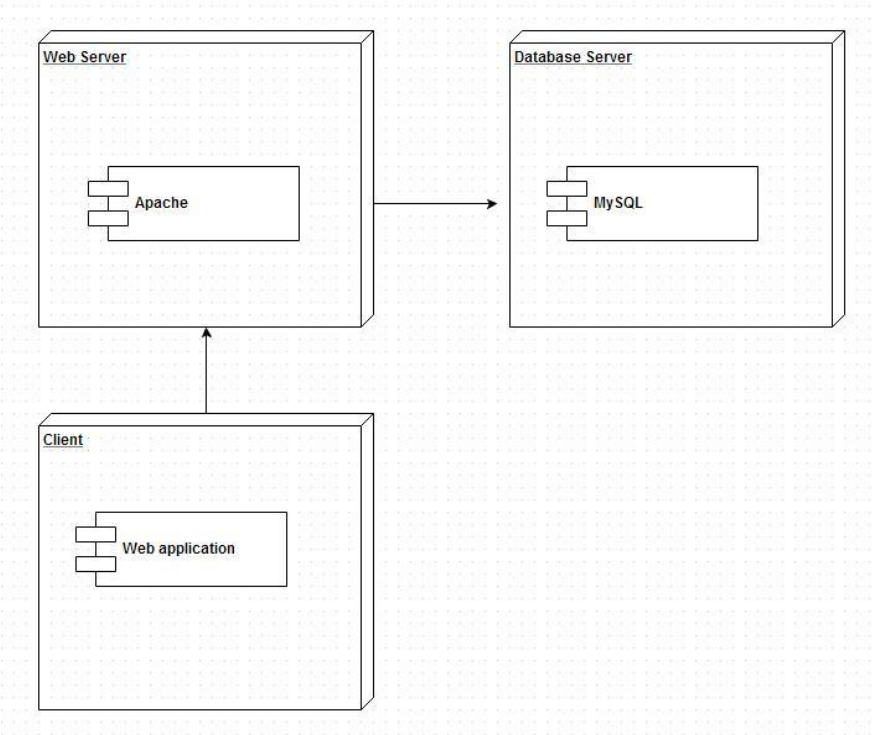
5.3 COMPOSITION VIEWPOINT

5.3.1 DESIGN CONCERNS

Composition viewpoint will help to manage software process. In this part, main work components and components inside those will be identified.

There will be tree main work components in this software. Namely: Database server, web server and client.

5.3.2 DESIGN ELEMENTS



- Design Entities: There are three main design components in our project which are namely client, database server and webserver.

- Webserver establishes a web interface between the client and the database.
- Design Attributes: Design attributes are discussed in the following two chapters.
- Database Server is placed at lower level of system

5.3.2.1 FUNCTION ATTRIBUTE

Database server, web server and client are the main components of the SafeCloud Project. Webserver is responsible for providing an interaction between client and the Database through a browser. The database will store documents and files which is in the container.

5.3.2.2 SUBORDINATES ATTRIBUTE

All of the components mentioned in the previous subsection are composed together in order to construct the main SafeCloud.

5.4 LOGICAL VIEWPOINT

In this section, the classes used in the project and their relations are explained in details. Firstly, a complete class diagram containing all classes and their relations are given. Then, each of the classes and their methods and fields are explained in details.

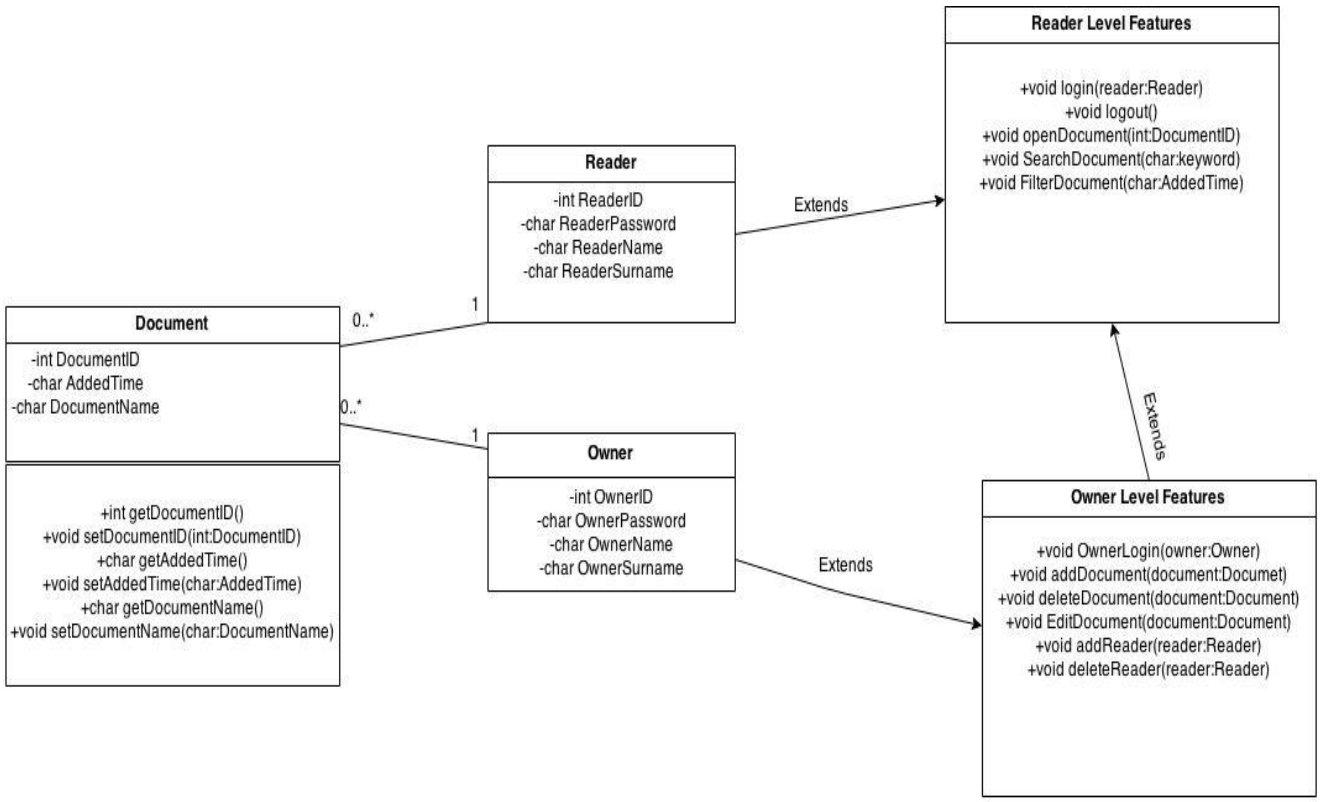
5.4.1 DESIGN CONCERNS

Logical viewpoint will identify all classes and relations between classes in SafeCloud. Aim of logical viewpoint is to clarify and simplify the system design and lead development team.

5.4.2 DESIGN ELEMENTS

5.4.2.1 CLASS RELATIONS

The class diagram below shows the class relations in SafeCloud. There are five classes related each other, namely ownerLevelFeatures, readerLevelFeatures, reader, owner and documents.



5.4.2.2 DOCUMENT CLASS

Document class is represents files which are keeped in container.

Method/Field	Definition
int documentID	The ID of document.
char documentName	The name of the document.
char addedTime	Date when document is upload.

void getDocumentID()	Returns documentID of the document.
void setDocumentID(int ID)	Sets value to documentID.
char getDocumentName()	Returns name of the document.
void setDocumentName(char Name)	Sets name to document.
char getAddedTime()	Returns upload time of document.
void setAddedTime(char Time)	Sets time to uploadtime.

5.4.2.3 READER CLASS

Reader class is represents users who have permission to open documents.

Method/Field	Definition
int readerID	The ID of reader.
char readerPassword	The password of reader
char readerName	The name of reader.
char readerSurname	The surname of reader.

,

5.4.2.4 OWNER CLASS

Owner class is represents user who has owner level features.

Method/Field	Definition
int ownerID	The ID of owner.
char ownerPassword	The password of owner.
char ownerName	The name of owner.
char ownerSurname	The surname of owner.

5.4.2.5 READER LEVEL FEATURES CLASS

Reader level features such as searchdocument,opendocument etc. are kept in this class.

Method/Field	Definition
void login(reader Reader)	Executes login activity.
void logout()	Executes logout activity.
void openDocument(int DocumentID)	Decrypt and open selected document.
void searchDocument(char keyword)	Searchs by keyword in all documents.
void filterDocument(char addedTime)	Filters all documents by given date.

5.5.2.6 OWNER LEVEL FEATURES CLASS

Owner level features such as addDocument, deleteDocument etc. are kept in this class.

Method/Field	Definition
void ownerLogin(owner Owner)	Executes login activity.
void logout()	Executes logout activity.
void addDocument(document Document)	Encrypt and upload selected document.
void deleteDocument(document Document)	Delete selected document from container.
void editDocument(document Document)	Edit selected document.
void AddReader(reader Reader)	Add given reader to container.
void DeleteReader(reader Reader)	Delete selected reader from container.

5.5 DEPENDENCY VIEWPOINT

The dependency viewpoint specifies the relationships and dependencies between the design components of the system.

5.5.1 DESIGN CONCERNS

Identifying the dependencies of the SafeCloud system and determining which subsystems are depends on other subsystems helps deciding the priorities in developing design entities.

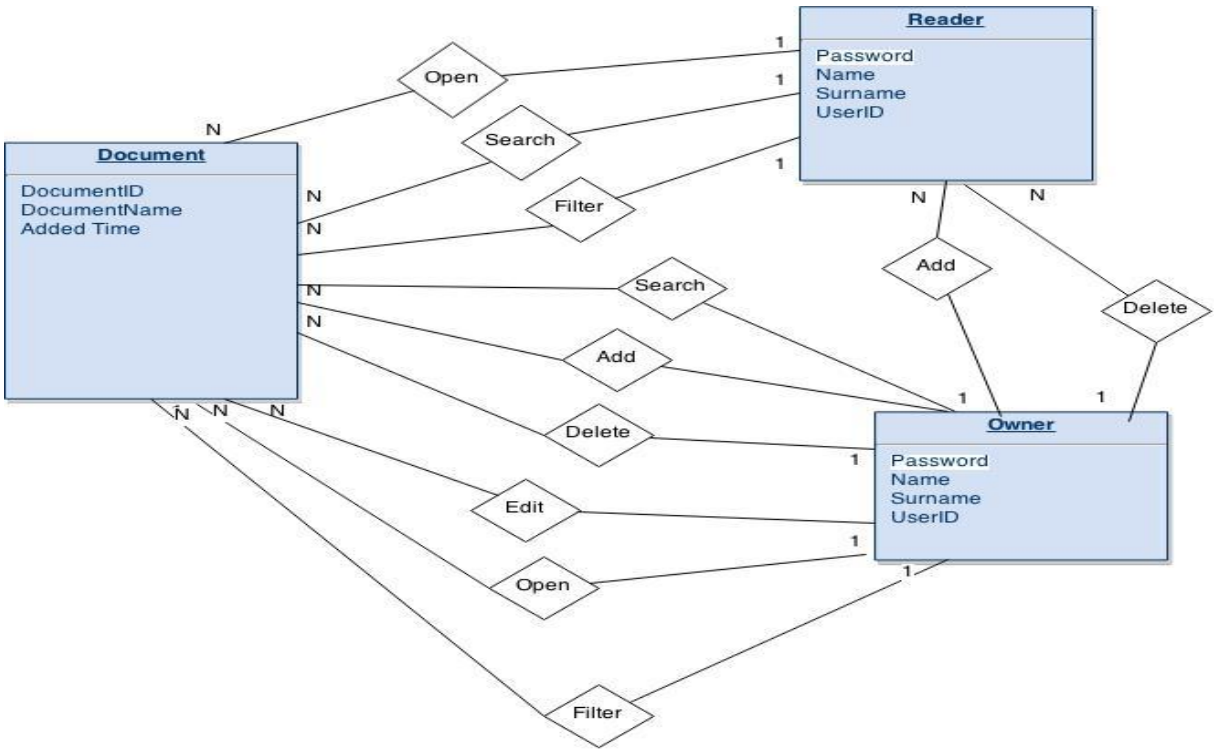
5.5.2 DESIGN ELEMENTS

Design Entities: Webserver (Apache), database server (MySQL) and the client.

Design Relationships: Each entities are related each other. Database Server is placed at lower level of system. Webserver is between the client and database.

5.6 INFORMATION VIEWPOINT

In this section, relations of the different classes are explained. Main purpose of the following ER diagram is to explain class diagram in a different way. One can understand relationships between classes without being confused by attributes easily by this diagram.



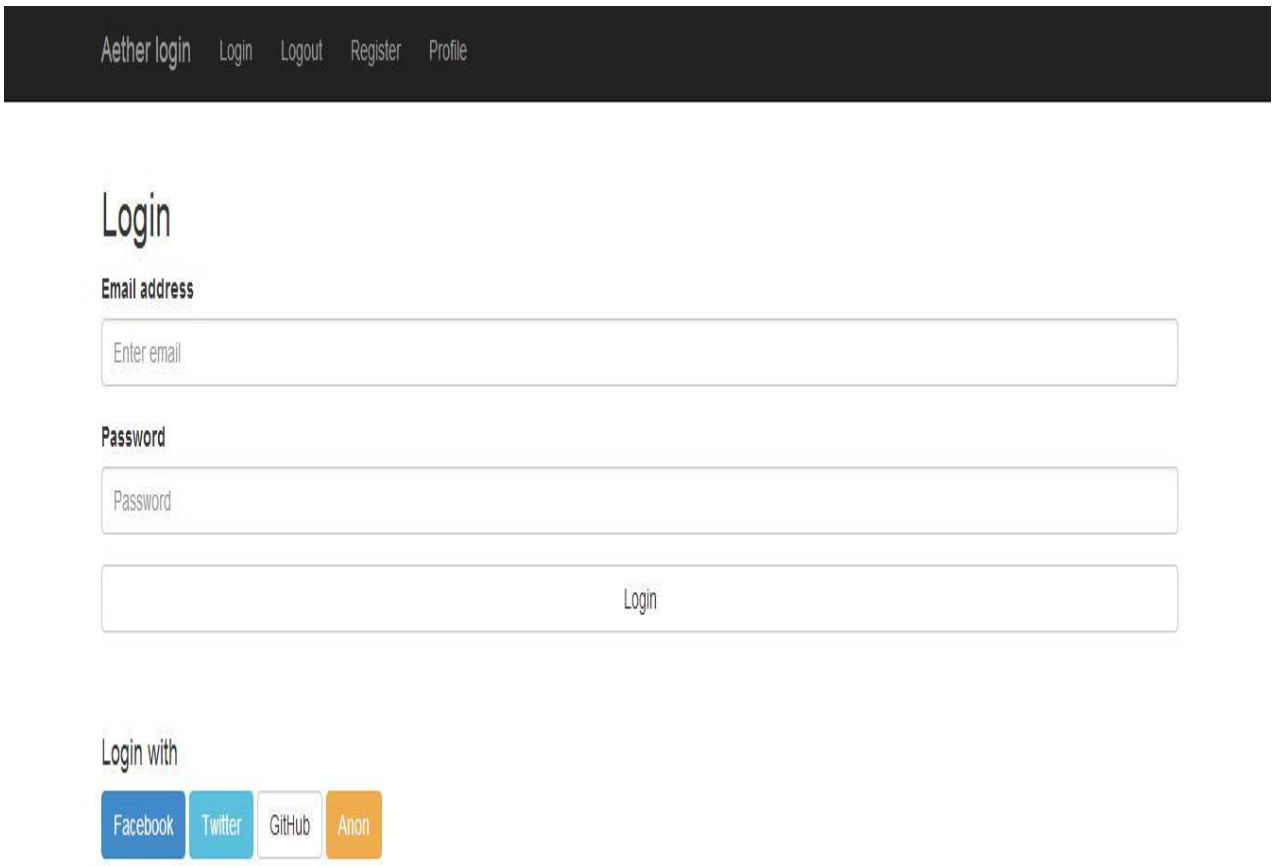
5.7 INTERFACE VIEWPOINT

5.7.1 DESIGN CONCERNS

The interface viewpoint provides information for designers, programmers, testers and customers to acquire knowledge about how to use SafeCloud correctly. In every subsection, related interfaces will be described in details.

5.7.2 DESIGN ELEMENTS

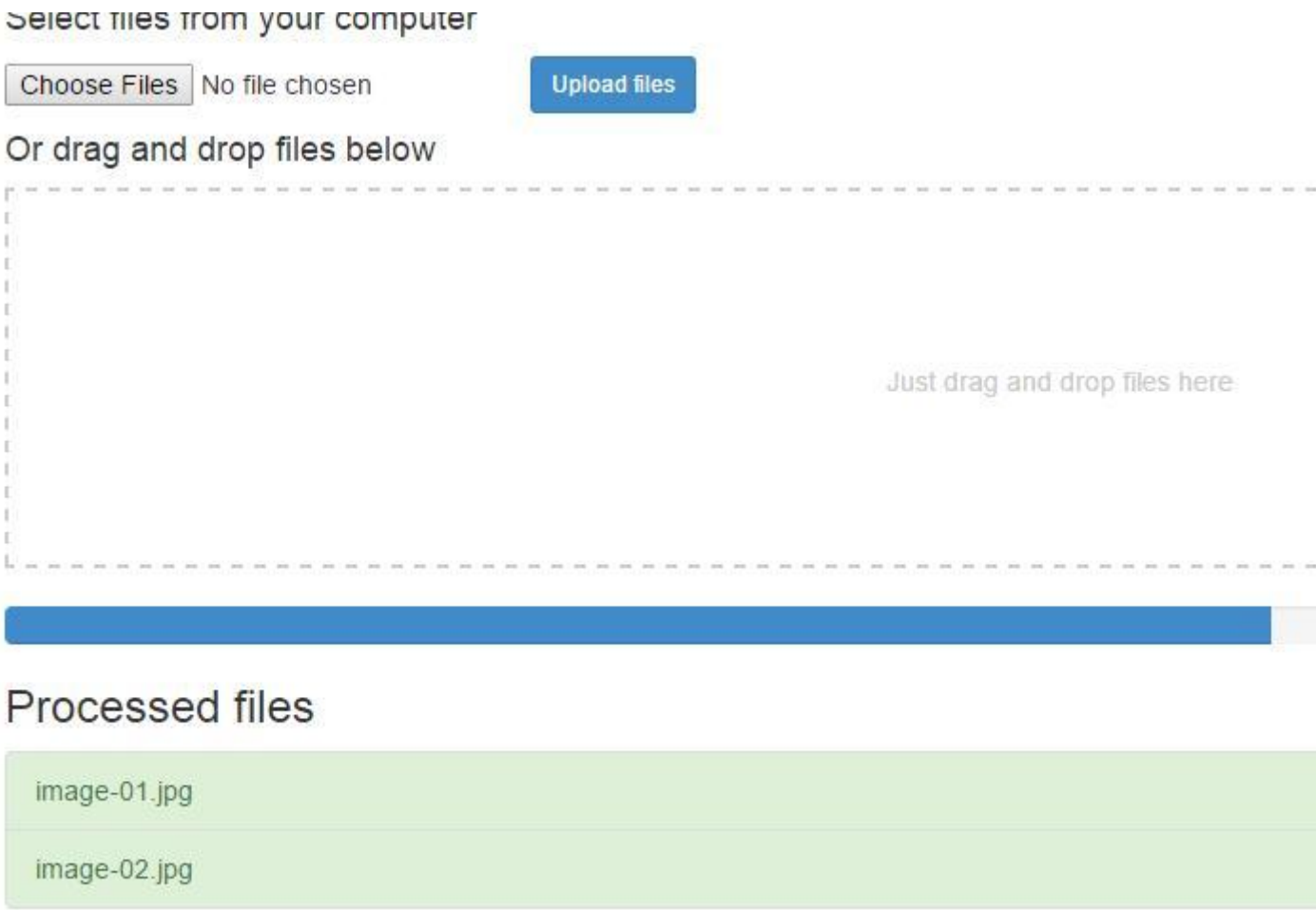
5.7.2.1 LOGIN PAGE



Login page allows users to enter the container page. In order to perform operations, user should be first login to the SafeCloud system. After filling the related fields, user can login to the system by

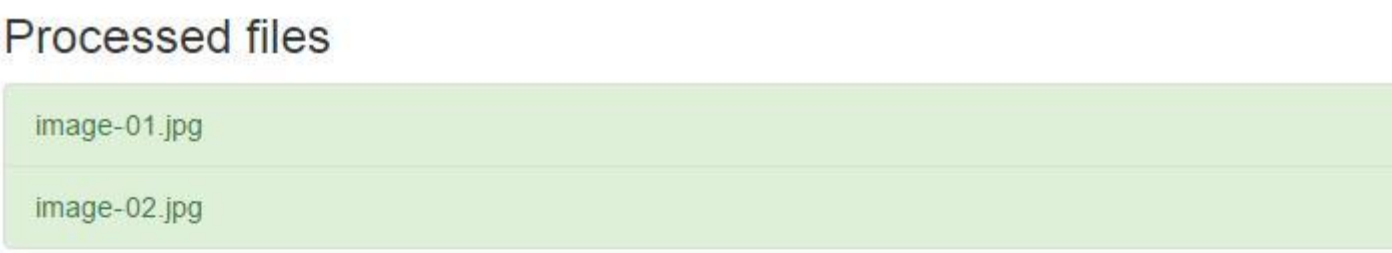
clicking login button.Moreover,user can login with Facebook,Twitter,GitHub and Anon accounts. The figure below shows the login page interface.

5.7.2.2 OWNER MAIN PAGE



Owner main page allows owner to perform owner class operations.Owner can upload new document by choosing from computer files or just drag and drop.If owner wants to delete or edit document,s/he must choose one of the documents which is in the container and perform operation.

5.7.2.3 READER MAIN PAGE



In reader main page,reader only has permission to see documents that are in the container.To open a document,reader must choose one of the documents.

5.8 STRUCTURE VIEWPOINT

This design view is governed in chapter 5.5 Dependency viewpoint and 5.3 Composition viewpoint.

5.9 INTERACTION VIEWPOINT

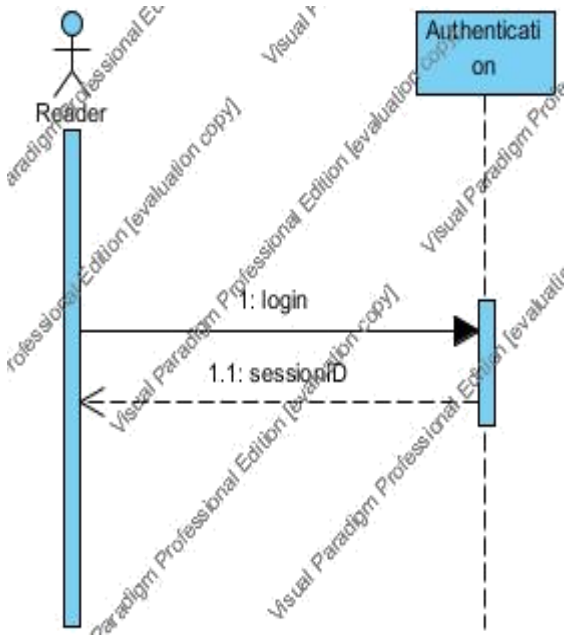
5.9.1 DESIGN CONCERNS

The main concerns of the interaction viewpoint and this viewpoint is important when designing and adopting patterns. Sequence Diagram is going to be used to show interaction between objects. Since flow of event is shown sequentially in the diagram, order of the events, satisfied and unsatisfied conditions can easily be understood.In this viewpoint we explained the relations of modules and functions to each other and the flow of events. This viewpoint clarifies the communication and messaging between user and modules. The flow of events is shown sequentially which makes understanding the occasion times of events easy. We separate this viewpoint according to modules and every module includes sequence diagrams showing its functions and fields which provide communication between user and other modules.

5.9.2 DESIGN ELEMENTS

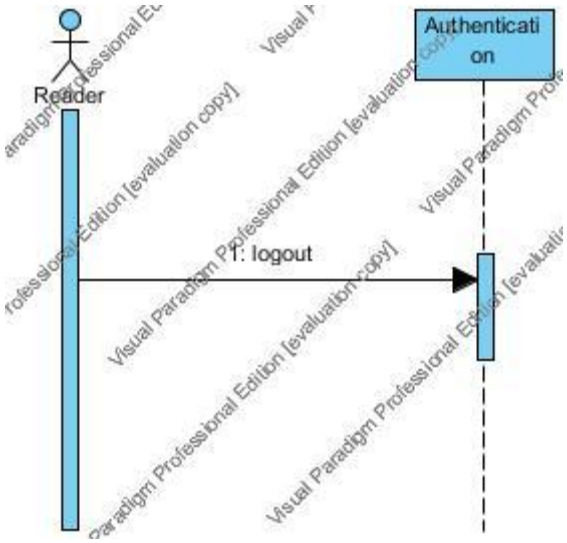
5.9.2.1 READER LEVEL FEATURES

5.9.2.1.1 LOGIN



In the diagram reader sends login request to Authentication and receives sessionID.

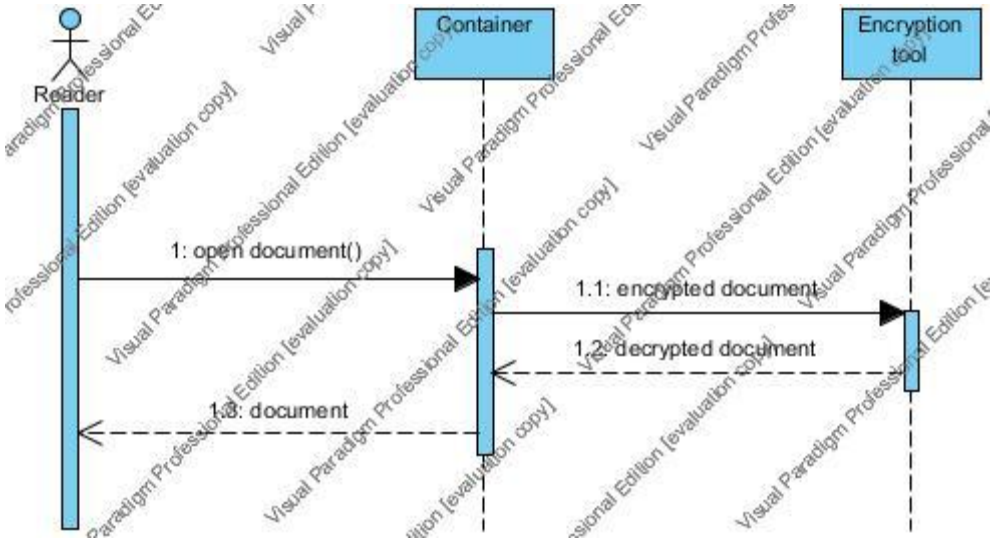
5.9.2.1.2 LOGOUT



Reader should be log out from system by sending logout request to Authentication.

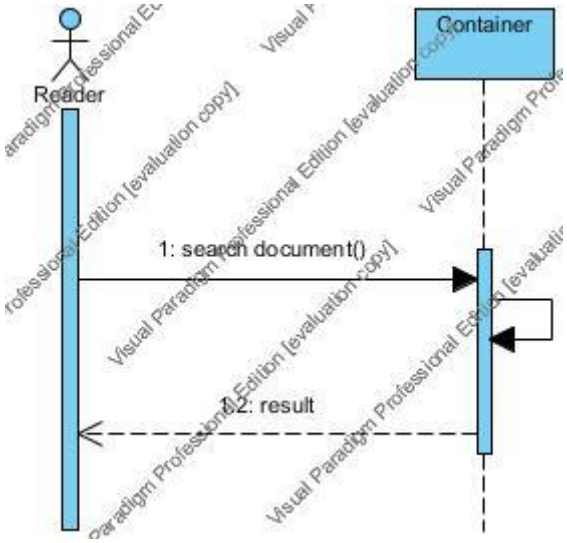
,

5.9.2.1.3 OPEN DOCUMENT



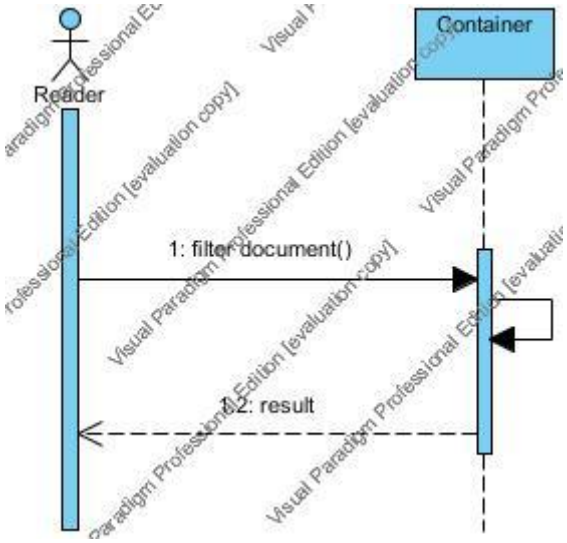
To open document, reader should send request to container.Container send encrypted document to encryption tool, encryption tool send decrypted document to container, finally container send documents to reader.

5.9.2.1.4 SEARCH DOCUMENT



To search document, reader should send request to container.After searching,container sends documents which is match with reader's keywords.

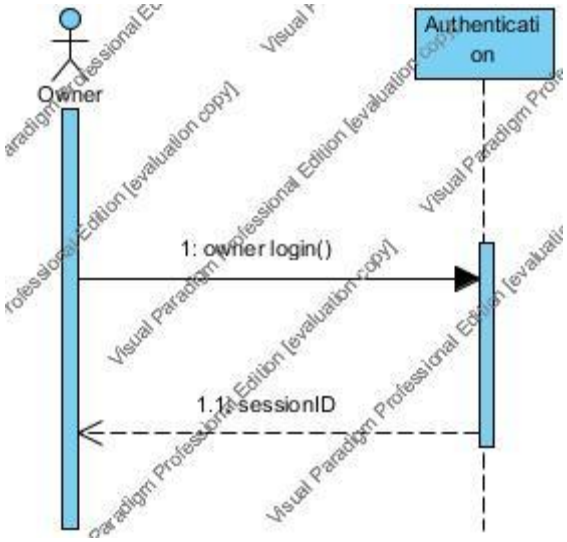
5.9.2.1.5 FILTER DOCUMENT



To filter document,reader should send request to container.After searching,container sends documents which is match with reader's date.

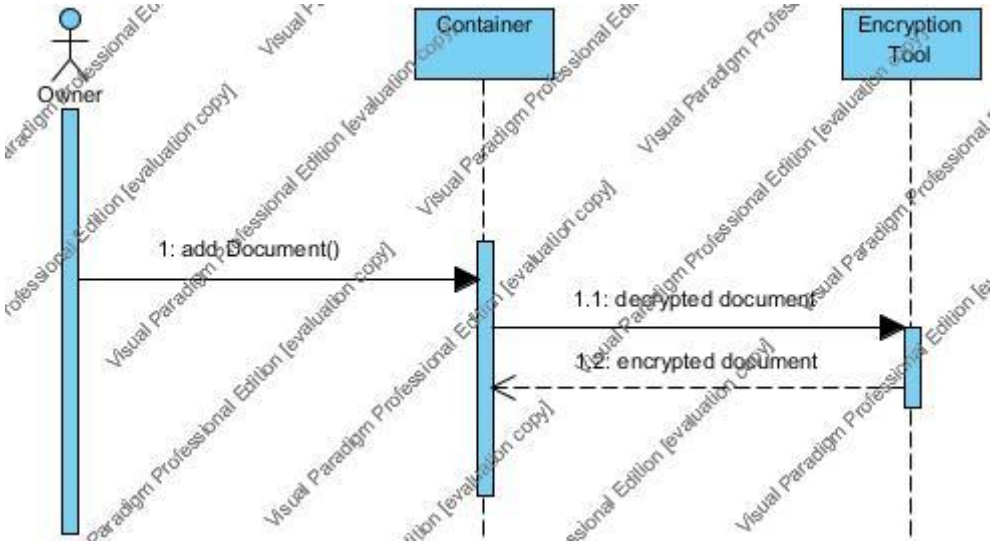
5.9.2.2 OWNER LEVEL FEATURES

5.9.2.2.1 OWNER LOGIN



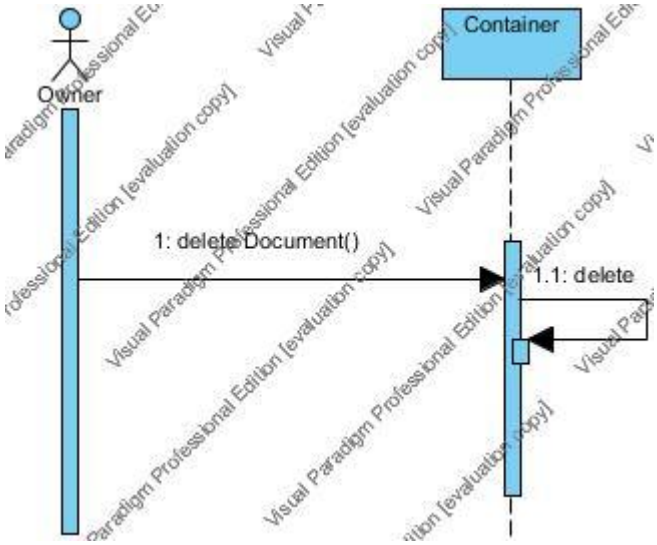
In the diagram owner sends login request to Authentication and receives sessionID.

5.9.2.2.2 ADD DOCUMENT



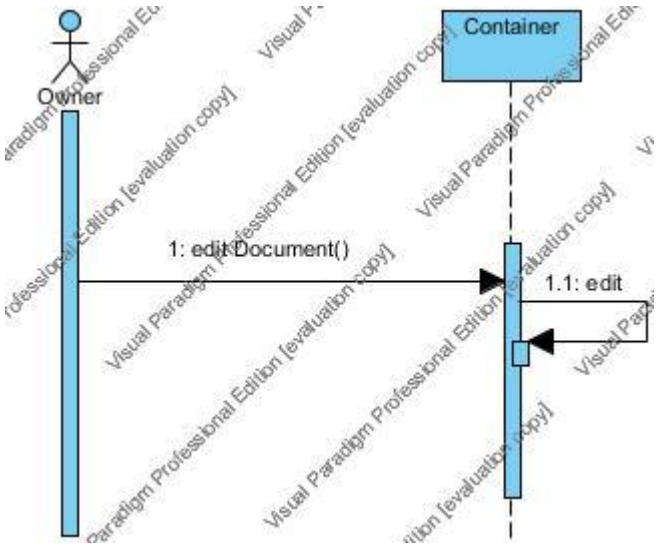
First owner sends document to container.Container sends decrypted document to encryption tool and encryption tool send encrypted document to container.Encrypted document is keeped in container.

5.9.2.2.3 DELETE DOCUMENT



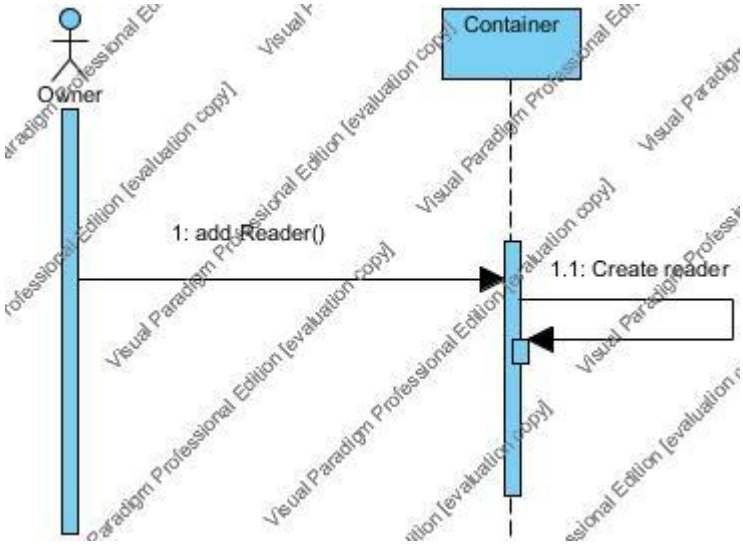
To delete document,owner sends request to container. Container finds and delete document.

5.9.2.2.4 EDIT DOCUMENT



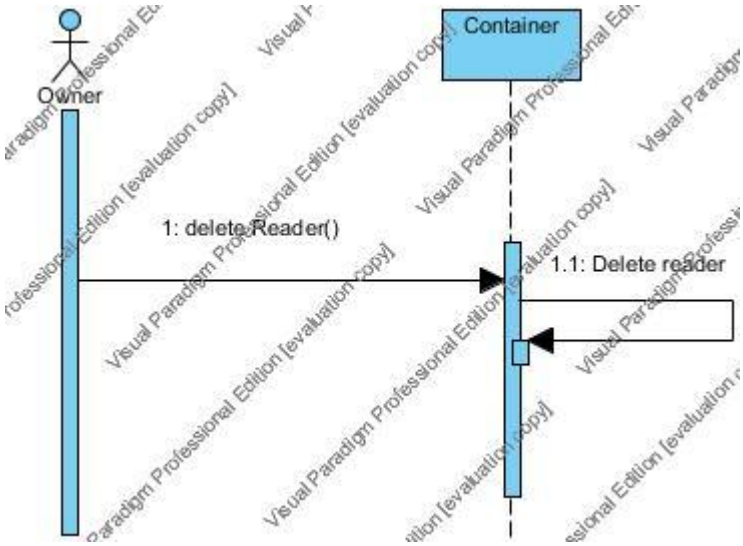
To edit document, owner sends request to container. Container finds and edit document.

5.9.2.2.5 ADD READER



To add reader,owner sends request to container. Container create reader.

5.9.2.2.6 DELETE READER



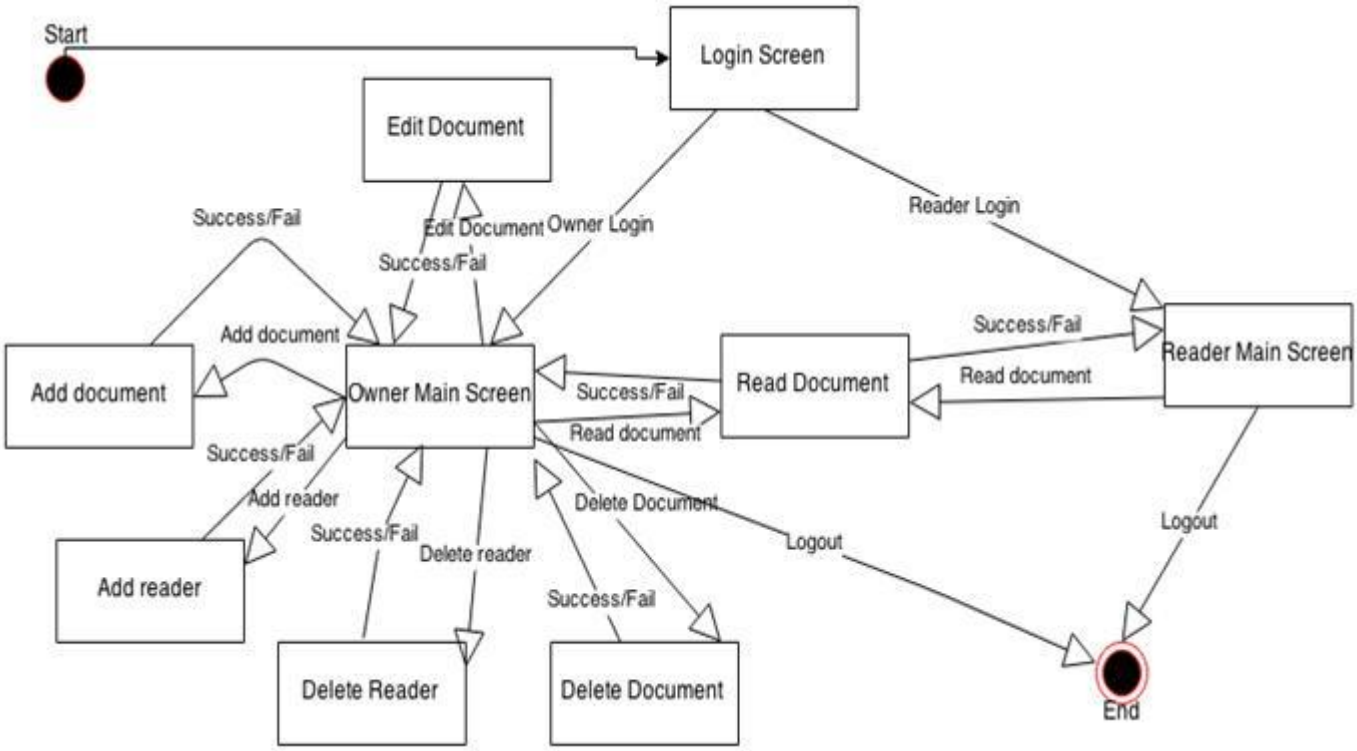
To delete reader, owner sends request to container. Container delete reader.

5.10 STATE DYNAMICS VIEWPOINT

5.10.1 DESIGN CONCERNS

State dynamics viewpoint shows the behavior of the system when some specific events occur. This viewpoint also related to the logical view. Since it explains in which state which condition occurs and how is it handled. Basic design concerns we take care in this viewpoint are the modes, states, transitions among events and reactions given to the events.

5.10.2 DESIGN ELEMENTS



In our project, first, user should be logged in to system.There are two types log-in,reader and owner.If user is reader,s/he will be directed to reader main page.In reader main page,s/he only has permission to read document.After logout,s/he will directed to login page again.If user is owner,s/he will be directed to owner mai page.In owner main page,owner can perform any activity which is in owner level features class like add document,delete document etc.After logout,s/he will directed to login page again.