

# **System and Software Architecture Description (SSAD)**

**E-Lock Box**

**Team 05**

## **Team Members**

Miles Gui  
Azuka Okuleye  
Huaiqi Wang

Weiyi Zhong  
Woon Kim  
Miles Gui  
Cecilia Jou

## **Roles**

Project Manager  
Builder & Feasibility analyst  
Tester & Operational Concept  
Engineer  
Builder & Prototyper  
Builder & System Architect  
Builderr & Life Cycle Planner  
Tester & Quality Focal Point

**UCS-CSSE**

**April 28 2014**

# Version History

Date	Author	Version	Changes made	Rationale
10/15/13	Woon Kim, Dejie Meng	1.0	<ul style="list-style-type: none"> <li>Add Section 1 and Section 2</li> </ul>	<ul style="list-style-type: none"> <li>Initial Draft SSAD</li> </ul>
10/23/13	Woon Kim, Dejie Meng	1.1	<ul style="list-style-type: none"> <li>Modify Section 1 and Section 2</li> </ul>	<ul style="list-style-type: none"> <li>SSAD for FC package</li> </ul>
10/28/13	Woon Kim, Dejie Meng	1.2	<ul style="list-style-type: none"> <li>Modify Section2</li> </ul>	<ul style="list-style-type: none"> <li>Added more use case and changed use case diagram as well as artifacts and information diagram</li> </ul>
11/28/13	Woom Kim, Dejie Meng	2.0	<ul style="list-style-type: none"> <li>Change Template</li> <li>Modify Section 2</li> <li>Adding Section 3</li> </ul>	<ul style="list-style-type: none"> <li>We use VPUML to generate all elements in the SSAD</li> <li>Added more use case and changed some use case description</li> <li>Added software class diagram and hardware class diagram</li> </ul>
12/02/13	Woom Kim, Dejie Meng	2.1	<ul style="list-style-type: none"> <li>Modify Section 3</li> <li>Adding Section 4</li> <li>Adding Section 5</li> </ul>	<ul style="list-style-type: none"> <li>Modified diagrams in section 3</li> <li>Added diagrams in section 4 and rationale in section 5</li> </ul>
12/07/13	Woom Kim, Dejie Meng	2.2	<ul style="list-style-type: none"> <li>Modify ER diagram</li> <li>Modify design class diagram</li> <li>Modify Section 5</li> </ul>	<ul style="list-style-type: none"> <li>Add more detail and right some wrong in the ER diagram</li> <li>Add more detail and relation in design class diagram</li> <li>Compare framework in the section 5</li> </ul>
02/11/14	Woon Kim, Dejie Meng	2.3	<ul style="list-style-type: none"> <li>Modify E-R diagram</li> <li>Modify class diagram</li> </ul>	<ul style="list-style-type: none"> <li>Add more tables</li> <li>Add more attributes for most table</li> <li>Convert class diagram based on E-R diagram</li> </ul>
02/18/14	Woon Kim, Dejie Meng	2.4	<ul style="list-style-type: none"> <li>Modify E-R diagram</li> <li>Modify Use cases</li> </ul>	<ul style="list-style-type: none"> <li>Recaptured the screen shot from VP-UML</li> <li>Change the precondition of each use case.</li> <li>Remove the guideline for each part.</li> </ul>
04/02/14	Woon Kim, Huaiqi Wang	2.5	<ul style="list-style-type: none"> <li>Modify E-R diagram</li> <li>Modify Hardware diagram</li> <li>Modify Software diagram</li> <li>Modify Deployment diagram</li> </ul>	<ul style="list-style-type: none"> <li>Modify diagram based on the instruction from TA and the progress of project.</li> </ul>

# Table of Contents

<b>System and Software Architecture Description (SSAD).....</b>	<b>i</b>
<b>Version History.....</b>	<b>ii</b>
<b>Table of Contents.....</b>	<b>iii</b>
<b>Table of Tables.....</b>	<b>iv</b>
<b>Table of Figures.....</b>	<b>v</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose of the SSAD.....	1
1.2 Status of the SSAD.....	1
<b>2. System Analysis.....</b>	<b>2</b>
2.1 System Analysis Overview.....	2
2.2 System Analysis Rationale.....	21
<b>3. Technology-Independent Model.....</b>	<b>22</b>
3.1 Design Overview.....	22
3.2 Design Rationale.....	26
<b>4. Technology-Specific System Design.....</b>	<b>27</b>
4.1 Design Overview.....	27
4.2 Design Rationale.....	34
<b>5. Architectural Styles, Patterns and Frameworks.....</b>	<b>35</b>

# Table of Tables

<i>Table 1: Actors Summary.....</i>	<i>3</i>
<i>Table 2.1: ER Diagram Summary.....</i>	<i>4</i>
<i>Table 3: Hardware Component Description.....</i>	<i>22</i>
<i>Table 4: Software Component Description.....</i>	<i>23</i>
<i>Table 5: Design Class Description.....</i>	<i>25</i>
<i>Table 6: Hardware Component Description.....</i>	<i>27</i>
<i>Table 7: Software Component Description.....</i>	<i>28</i>
<i>Table 8: Design Class Description.....</i>	<i>31</i>
<i>Table 9: Architectural Styles, Patterns, and Frameworks.....</i>	<i>35</i>

# Table of Figures

<i>Figure 1: System Context Diagram.....</i>	<i>3</i>
<i>Figure 2.1: ER Diagram.....</i>	<i>4</i>
<i>Figure 3.1: Process Diagram.....</i>	<i>7</i>
<i>Figure 4: Hardware Component Class Diagram.....</i>	<i>22</i>
<i>Figure 5: Software Component Class Diagram.....</i>	<i>22</i>
<i>Figure 6: Deployment Diagram.....</i>	<i>24</i>
<i>Figure 7: Design Class Diagram.....</i>	<i>25</i>
<i>Figure 8: Hardware Component Class Diagram.....</i>	<i>27</i>
<i>Figure 9: Software Component Class Diagram.....</i>	<i>28</i>
<i>Figure 10: Deployment Diagram.....</i>	<i>29</i>
<i>Figure 11: Design Class Diagram.....</i>	<i>30</i>
<i>Figure 12: Process Realization Diagram: Upload Document.....</i>	<i>32</i>

# **1. Introduction**

## **1.1 Purpose of the SSAD**

The purpose of this SSAD is to record the architecture of e-lock box system including detailed diagram of how system will work and be organized according to according to operational concept and requirements. The programmers and developers of our team can use this SSAD as reference to the architecture of the system that we are developing.

## **1.2 Status of the SSAD**

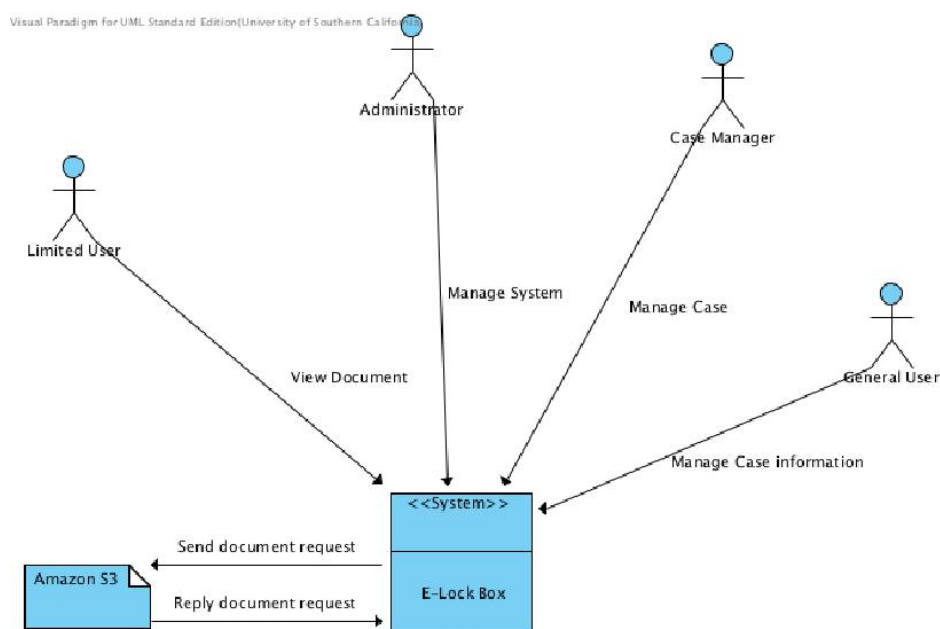
This SSAD is version 2.2, which we added more specific detail and correct some relation mistake in the ER diagram and finish the summary. Also, we added more detail in the design class diagram, including involving more controllers and boundaries and entities into the diagram. All of these improvements are based on version 2.1.

## **2. System Analysis**

### **2.1 System Analysis Overview**

The primary purpose of the Living Advantage's new website is to store foster youths' personal information such as birth certificate, student id, and driver's license safely so they can have access to their documents as they want. Also, foster youths are treated as limited users. There are case managers who help fosters. They keep track of all information of what they have done on foster youths like uploading personal information and tracking history. Administrators will manage all users, files and cases in the system.

## 2.1.1 System Context



**Figure 1: System Context Diagram**

**Table 1: Actors Summary**

Actor	Description	Responsibilities
Limited user (Foster Youth)	Users who use this system to store their vital document	View their vital document
Case manager	Users who mainly responsible for limited user.	Help foster youths Upload document Keep track of responsible youth
Administrator	Users who control every other users and files in the system.	Manage all users Manage all documents Manage all cases
General User	Users who has limited authority to the system	View case information Assist case manager and administrator to record the activities



## 2.1.2 Artifacts & Information

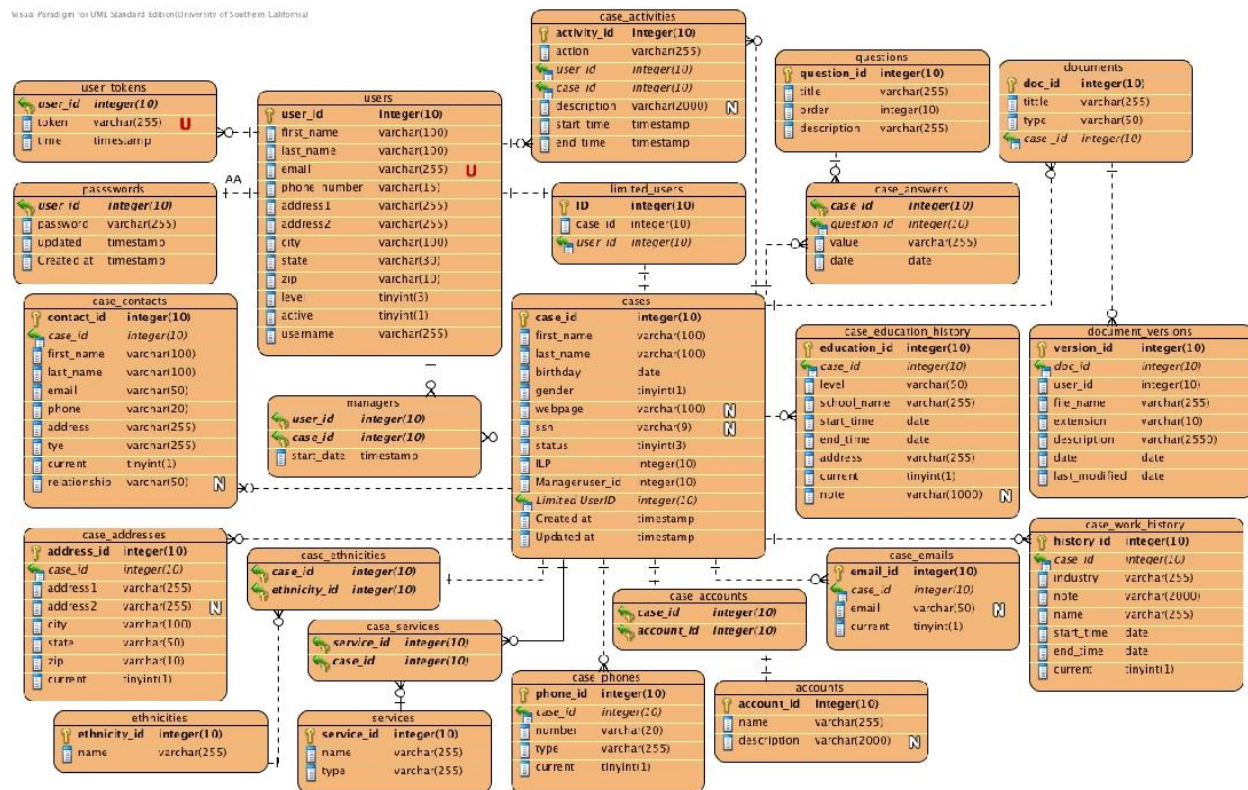


Figure 2.1: ER Diagram

Table 2.1: ER Diagram Summary

Table Name	Description
Password	It only contains users email address and password. The password will be encrypted
Users	Users can be four different type, Administrators, Case manager, general users, and limited users. It contains basic information about Users such as first name, last name, etc.
Administrator	It shows who the administrators are
Case manager	It shows who the case managers are
General users	It shows who the general users are
Limited users	It shows who the limited users are
Uploads	It shows which managers or administrators upload limited user's vital document
Vital Documents	It contains vital documents and type and which date the file is uploaded
Manages	It contains which case manager manage a limited users
Cases	It contains information about the cases
Contacts	It shows which users has which contact lists

Additional contacts	It shows additional information about the limited users
Activities	All activities such as upload, email contact updated in the table
Logs	It contains which case manager, administrator made contacts limited users
Education History	Limited user's education history will be stored in here
Educates	It contains the relationship between education history and limited user
Work history	Limited user's education history will be stored in here
Works	It contains the relationship between work history and limited user

## 2.1.3 Behavior

Use cases grid:

ID	Name	Primary Actors	Supporting Actors
UC01	View Document	Limited User Case Manager Administrator	
UC02	Edit Case	Case Manager Administrator	
UC03	Inactivate User	Administrator	
UC04	Create New Case	Case Manager Administration	
UC05	Report Generation	Case Manager Administrator	
UC06	Edit User	Administrator	
UC07	Activate	Administrator	
UC08	Upload Document	Case Manager Administrator	
UC09	Reset Password	Case Manager Administrator	
UC10	Check Case Information	General User Administrator Case Manager	
UC11	Create High Level User	Administrator	
UC12	Log Activity	General User Administrator Case Manager	
UC13	Create A Limited User	Case Manager Administrator	

Visual Paradigm for UML Standard Edition (University of Southern California)

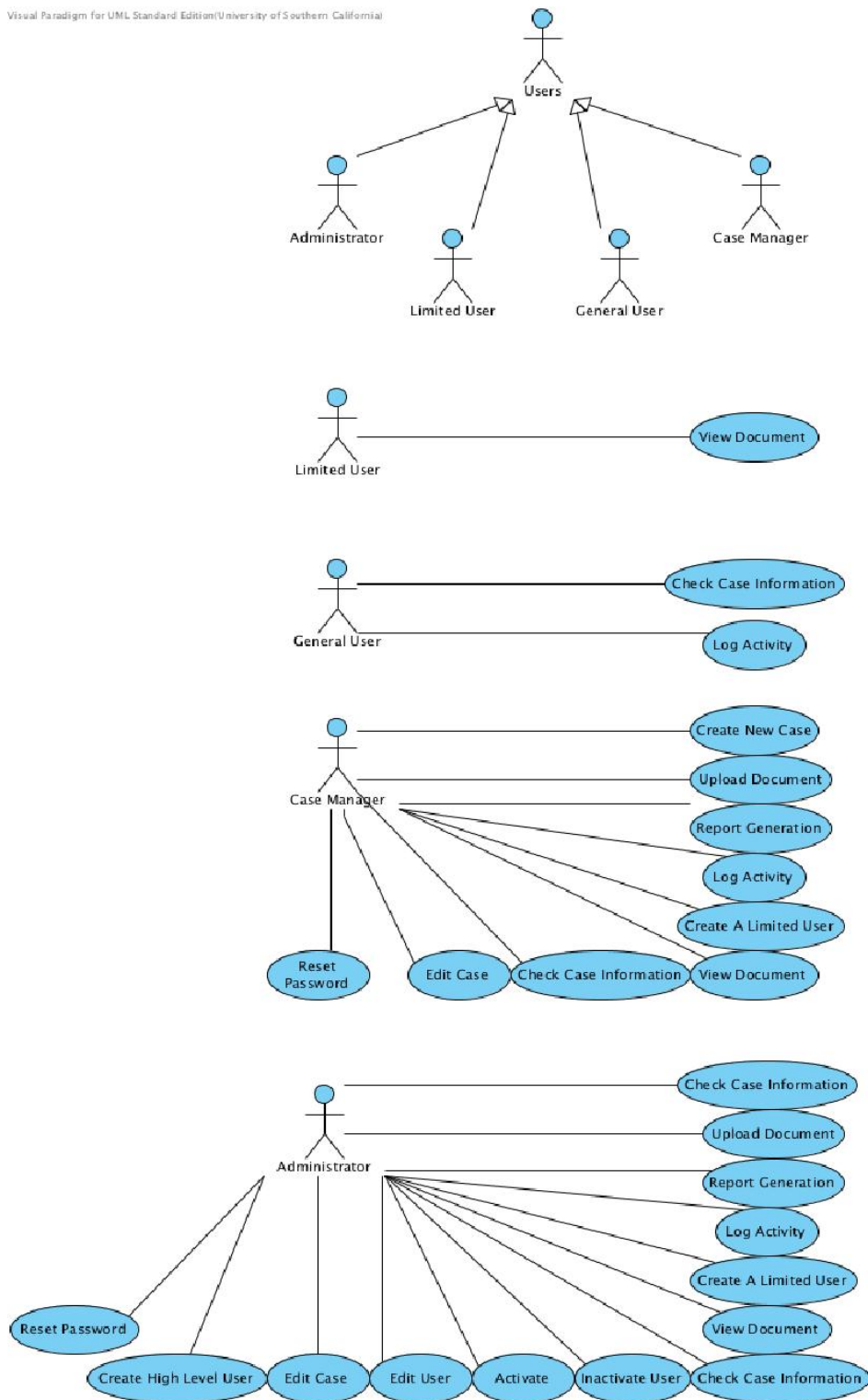


Figure 3.1: Process Diagram

## 2.1.3.1 Capability x

### 2.1.3.1.1 Check Case Information

#### 2.1.3.1.1.1 Use Case Descriptions

Main			
Super Use Case			
Author	dejiemen		
Date	Nov 27, 2013 1:35:07 PM		
Brief Description	Administrator, case manager and general user can see the case information of a foster youth		
Preconditions	1. Case has been built successfully in the system 2. Administrator, case manager and general user has logged into the system 3. Database has been set up		
Post-conditions	1. System will return the page of case information 2. Return unauthorized action information		
Flow of Events		Actor Input	System Response
	1	[Administrator and case manager and general user] Click a specific case	
	2		Send request to case database
	3		Return the case information page

### 2.1.3.1.2 Reset Password

#### 2.1.3.1.2.1 Use Case Descriptions

Main	
Super Use Case	
Author	dejiemen
Date	Nov 27, 2013 2:41:19 PM
Brief Description	If administrator and case manager want to change their password of the system, they can reset the password. Other kind of user want to change the password, they need to contact case manager or

	administrator.		
<b>Preconditions</b>	1. The user should exist in the system. 2. Administrator and case manager should login to the system 3. Database has been set up		
<b>Post-conditions</b>	1. New password will be refreshed into the database and return reset succeed message 2. Return invalid information 3. Return unauthorized action information		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	[Administration and case manager] Click "Reset password"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed, return password reset page
	5	Type in the new password and click "Submit"	
	6		Validate the information from password reset page
	7		If validate failed, return invalid information message
	8		If validate succeed, refresh the new password into the database and return reset succeed message
	9		Return to case management page

### 2.1.3.1.3 Create High Level User

#### 2.1.3.1.3.1 Use Case Descriptions

Main	
<b>Super Use Case</b>	
<b>Author</b>	dejiemen
<b>Date</b>	Nov 27, 2013 2:17:51 PM

<b>Brief Description</b>	When other people want to get into the system as a case manager or general user or administrator, administrator can create such high level user in the system.		
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Administrator has logged into the system</li> <li>2. Database has been set up</li> </ol>		
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. A new user will be created in the system and inserted into user database.</li> <li>2. Return invalid information</li> <li>3. Return unauthorized action information</li> </ol>		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	[Administrator] Click "Use Management"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed return user management page
	5	Click "Create user"	
	6		Return a user information page
	7	Type in the information and click "Submit"	
	8		Validate the information from user information page
	9		If validate failed, return invalid information message
	10		If validate succeed, insert the user into user database

#### 2.1.3.1.4 Edit Case

##### 2.1.3.1.4.1 Use Case Descriptions

Main	
<b>Super Use Case</b>	
<b>Author</b>	dejiemen
<b>Date</b>	Nov 27, 2013 2:32:24 PM

<b>Brief Description</b>	If some information of foster youth has changed, administrator and case manager can edit the case information of a foster youth		
<b>Preconditions</b>	1. Case should exist in the system 2. Administrator and case manager should login to the system 3. Database has been set up		
<b>Post-conditions</b>	1. New case information will be refreshed into the database 2. Return invalid information message 3. Return unauthorized action information		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	[Administrator and case manager] Choose a specific case	
	2		Send request to case database and receive the case information
	3		Return a case information page
	4	Click "Edit"	
	5		Validate the role of user
	6		If validate failed, return unauthorized action information
	7		If validate succeed, return a case information page for editing
	8	Type in the new case information and click "Submit"	
	9		Validate the information from case information page
	10		If validate failed, return invalid information message
	11		If validate succeed, refresh the new information into database
	12		Return a new case information page



### 2.1.3.1.5 Edit User

#### 2.1.3.1.5.1.1 Use Case Descriptions

Main			
Super Use Case			
Author	dejiemen		
Date	Nov 27, 2013 2:25:19 PM		
Brief Description	If some information of high level user has changed, administrator can change the information of high level user		
Preconditions	1. User should exist in the system 2. Administrator should login the system 3. Database has been set up		
Post-conditions	1. New user information will be refreshed into the database 2. Return invalid information 3. Return unauthorized action information		
Flow of Events		Actor Input	System Response
	1	[Administrator] Click "User Management"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed, return user management page
	5	Click a specific user	
	6		Send request to user database and return the user information page
	7	Click "Edit"	
	8		Return a user information page for editing
	9	Type in new information and click "Submit"	
	10		Validate the information from user information page
	11		If validate failed, return invalid information message
	12		If validate succeed, refresh

			new information into database
	13		Return a new user information page

### 2.1.3.1.6 Activate

#### 2.1.3.1.6.1 Use Case Descriptions

Main			
Super Use Case			
Author	dejiemen		
Date	Nov 27, 2013 2:11:51 PM		
Brief Description	If some off system users want to come back system, administrator can activate corresponding user		
Preconditions	<ol style="list-style-type: none"> <li>1. User should exist in the system</li> <li>2. User should be inactivated</li> <li>3. Administrator has logged into the system</li> <li>4. Database has been set up</li> </ol>		
Post-conditions	<ol style="list-style-type: none"> <li>1. A user will be successful activated in the system</li> <li>2. Return unauthorized action information</li> <li>3. Return wrong action message</li> </ol>		
Flow of Events		Actor Input	System Response
	1	[Administrator] Click "User management"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed, return user management page
	5	Select a specific user and click "Activate"	
	6		Check the status of user
	7		If the status is activated, return wrong action message
	8		If the status is inactivated, activate the user and change

		the status of user in the database

### 2.1.3.1.7 Inactivate User

#### 2.1.3.1.7.1 Use Case Descriptions

Main			
<b>Super Use Case</b>			
<b>Author</b>	dejiemen		
<b>Date</b>	Nov 27, 2013 2:03:48 PM		
<b>Brief Description</b>	If a user is no longer in the system, administrator can inactivate the user		
<b>Preconditions</b>	1. User should exist in the system 2. Administrator has logged into the system 3. User should be activated 4. Database has been set up		
<b>Post-conditions</b>	1. A limited user and the related case will be inactivate in the system, only administrator can see inactivate case and then activate it. 2. Return unauthorized action information 3. Return wrong action message		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	[Administrator] Click "User management"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed, return user management page
	5	Select a specific user and click "Inactivate"	
	6		Check the status of user
	7		If the status is inactivated, then return wrong action message
	8		If the status is activated, then inactivate the user and change the status of user in

		database

### 2.1.3.1.8 View Document

#### 2.1.3.1.8.1 Use Case Descriptions

Main			
Super Use Case			
Author	dejiemen		
Date	Nov 26, 2013 5:23:54 PM		
Brief Description	Administrator and case managers and limited users can see the vital document in the system.		
Preconditions	1. Document should has been uploaded into the system 2. Administrators or case mangers or limited user have logged into the system 3. Database has been set up		
Post-conditions	1. Users can see the information of the vital document 2. Return unauthorized action information		
Flow of Events		Actor Input	System Response
	1	Click a specific document record	
	2		Validate the role of users
	3		If validate failed, return unauthorized information
	4		If validate succeed, send doc request to document database
	5		Database return document and generate a preview of document
	6	Click "Download" or "Print"	
	7		Validate the role of users
	8		If validate failed, return unauthorized information
	9		If validate succeed, send request corresponding controller

### 2.1.3.1.9 Create A Limited User

#### 2.1.3.1.9.1 Use Case Descriptions

Main			
Super Use Case			
Author	dejiemen		
Date	Nov 27, 2013 1:15:55 PM		
Brief Description	Administrator or case manager can create a limited user so that a foster youth will have an username and password to login to the system		
Preconditions	1. Case should exist in the system 2. Administrator and case manager has logged into the system 3. Database has been set up		
Post-conditions	1. A new user will be generated and the username and password will be recorded in the database and get generate succeed message. 2. Return unauthorized action information		
Flow of Events		Actor Input	System Response
	1	[Administrator or case manager] Click "Create user"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed, generate a random password related to the username, which is the email address in the case
	5		Insert the new username and password in the user database and send request to case database
	6		Connect case with a limited user and return generate succeed message

### 2.1.3.1.10 Log Activity

#### 2.1.3.1.10.1 Use Case Descriptions

Main			
Super Use Case			
Author	dejiemen		
Date	Nov 27, 2013 1:08:14 PM		
Brief Description	Administrator, case manager and general user can log activities related to a foster youth		
Preconditions	1. Case should exist in the system 2. Administrator or case manager has logged into the system 3. Database has been set up		
Post-conditions	1. A new record of activity will appear in the case information 2. Return "Description should not be blank" message		
Flow of Events`		Actor Input	System Response
	1	[Administrator, case manager and general user] Click "Add new"	
	2		Return activity description page
	3	Type in the information and click "Save"	
	4		Validate the description of activity
	5		If description is blank, return "Description should not be blank" message
	6		If validate succeed, insert activity into case
	7		Update new activity record in case information page

### 2.1.3.1.11 Report Generation

#### 2.1.3.1.11.1 Use Case Descriptions

Main
------

<b>Super Use Case</b>			
<b>Author</b>	dejiemen		
<b>Date</b>	Nov 27, 2013 12:57:40 PM		
<b>Brief Description</b>	Administrator or case manager can generate different kinds of report to get information from the system		
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Administrator and case manager has logged into the system</li> <li>2. All database has been set up</li> </ol>		
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. A new report will be generated from the system automatically and can be print and download</li> <li>2. Return unauthorized action information</li> <li>3. Return "Report cannot be blank" message</li> <li>4. Return invalid information message</li> </ol>		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	[Administrator and case manager] Click "Report"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed, return report generation page
	5	Choose the kind of report want to generate and type in necessary information	
	6		Validate the choice of report
	7		If choice is blank, return "Report cannot be blank" message
	8		If validate succeed, then validate the information from report page
	9		If validate failed, return invalid information message
	10		If validate succeed, send request to case database, document database and user database
	11		Receive information and generate a preview of report
	12	Click "Print"	

	13		Send request to corresponding controller

### 2.1.3.1.12 Upload Document

#### 2.1.3.1.12.1 Use Case Descriptions

Main			
<b>Super Use Case</b>			
<b>Author</b>	dejiemen		
<b>Date</b>	Nov 27, 2013 12:46:39 PM		
<b>Brief Description</b>	Administrators or case managers can upload documents for a foster youth in the system		
<b>Preconditions</b>	1. Case should exist in the system 2. Administrator or case manager has logged into the system 3. The paper edition of document has been scanned into the local machine as electrical edition 4. Database has been set up		
<b>Post-conditions</b>	1. A new vital document will be uploaded into the system and a record of new uploaded document will appear in the case information 2. Return invalid document description message 3. Return unauthorized action information		
<b>Flow of Events</b>		<b>Actor Input</b>	<b>System Response</b>
	1	[Administrator and case manager] Click "Add new"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed, return document upload page
	5	Choose document and type in the description of the document and click "Submit"	
	6		Validate the information of description
	7		If validate failed, return invalid description message



	8		If validate succeed, insert document into document database
	9		Update the case information page with new uploaded document record

### 2.1.3.1.13 Create New Case

#### 2.1.3.1.13.1 Use Case Descriptions

Main			
Super Use Case			
Author	dejiemen		
Date	Nov 27, 2013 12:37:52 PM		
Brief Description	Create a new case for a new foster youth of the system. The new case will include the basic information about this foster youth.		
Preconditions	<ol style="list-style-type: none"> <li>Administrators or case managers has logged into the system</li> <li>Database has been set up</li> </ol>		
Post-conditions	<ol style="list-style-type: none"> <li>A new case will be created into the database and return succeed message</li> <li>Return invalid message for invalid information in the case information page</li> <li>Return unauthorized action information</li> </ol>		
Flow of Events		Actor Input	System Response
	1	[Administrator or case manager] Click "Create new case"	
	2		Validate the role of user
	3		If validate failed, return unauthorized action information
	4		If validate succeed, return a case information page
	5	Type in the information and click "Submit"	
	6		Validate the information from case information page
	7		If validate failed, return

			invalid information message
	8		If validate succeed, insert case into case database return succeed message

### 2.1.4 Modes of Operation

The E-Lockbox will operate in only one mode, so nothing further need to be said of modes of operation.

## 2.2 System Analysis Rationale

1. Based on how users will interact with system, we want to give the basic description of four different levels of users are available in the E-Lockbox system:

(1) Limited Users: This user will be the foster youth responsible by Living Advantage. User can view their vital files in the system and contact case manager offline when it's necessary.

(2) Case managers: This user can upload vital files for foster youth and view foster youth's files. Also, the case managers can help foster youth when they need help, and track history, generate report.

(3) Administrators: This user is the highest level of authority in the system. User will be responsible to manage other users, files and cases in the system including activate and deactivate files, cases and users etc.

(4) General user: This user will be able to view case information, which not include the information of vital files of foster youths.

2. According to the system context diagram and user case diagram, we want to make following explanation:

(1) Each manipulation of each role according to system context diagram may correspond to many user cases listed from 2.1.3.1, which the user cases start from 2.1.3.1 will not be the whole of this system. We will update more user cases based on new version of operational concept, requirements and prototype.

### 3. Technology-Independent Model

#### 3.1 Design Overview

##### 3.1.1 System Structure

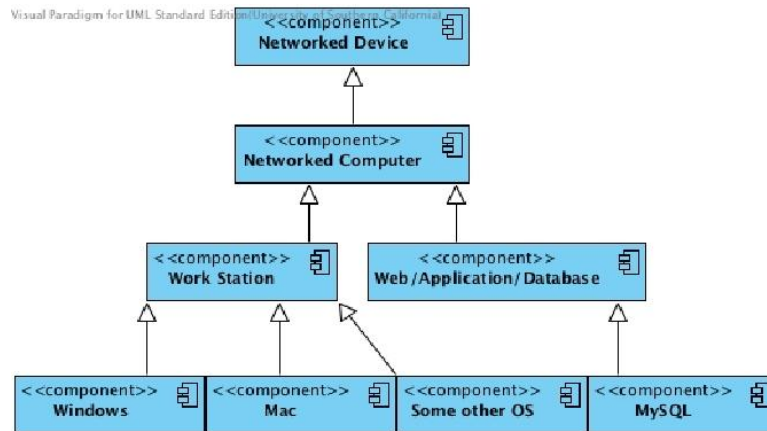


Figure 4: Hardware Component Class Diagram

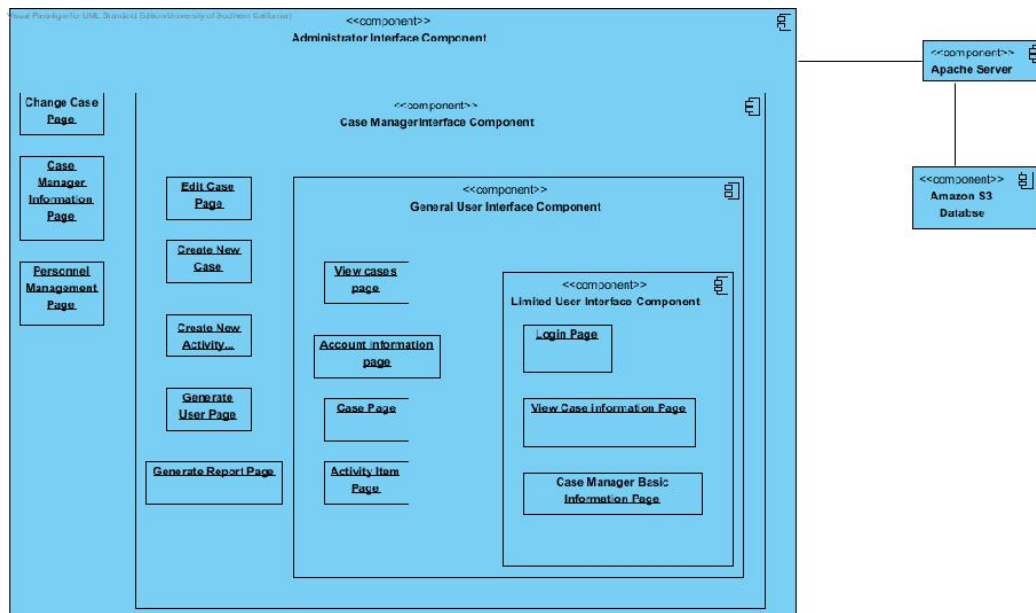


Figure 5: Software Component Class Diagram

Table 3: Hardware Component Description

Hardware Component	Description
--------------------	-------------

Networked Computer	It is a server that receive requests from Client and response the request
Work Station	Workstation is a station that a client runs. It request to the server. It includes multiple kind of operating system with different kind of browsers like Google chrome.
Web/application/database	

**Table 4: Software Component Description**

<b>Software Component</b>	<b>Description</b>
Login Page	User can Login through this page
View Case Information Page	It contains the information about the case of the chosen limited user.
Case Manager Basic Information Page	It contains the information about the case manager who controls a case
View Cases Page	It contains list of the cases that can be clicked to view detail
Account Information Page	It contains the logged in user's information and the user can update and change their password
Case Page	It contains all information for a foster youth including contact information, education and work history and vital document records
Activity Item Page	It contains all activity information
Edit Case Page	Case Manager and Administrator can edit the case page
Create New Case Page	Through this page, selected user can create new case.
Create New Activity Page	It can create new activity.
Generate User Page	It can create new user
Generate Report Page	Selected users can create report about the limited users' demographic, etc.
Case Manager Information Page	It contains a selected case manager's information.
Personnel Management Page	It contains all users' information. Through this page, administrator can activate, or deactivate user accounts

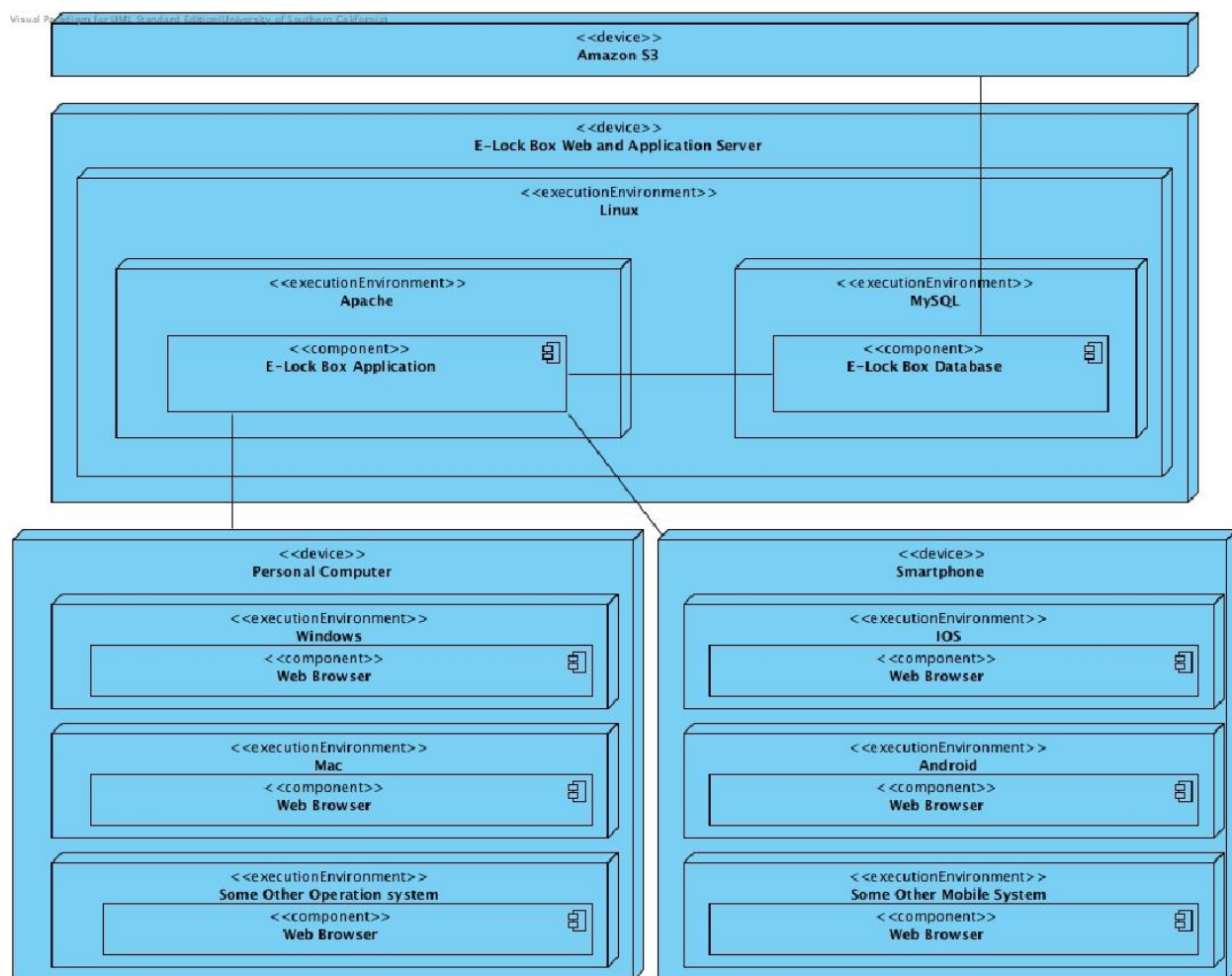


Figure 6: Deployment Diagram

## 3.1.2 Design Classes

### 3.1.2.1 <Classes n>

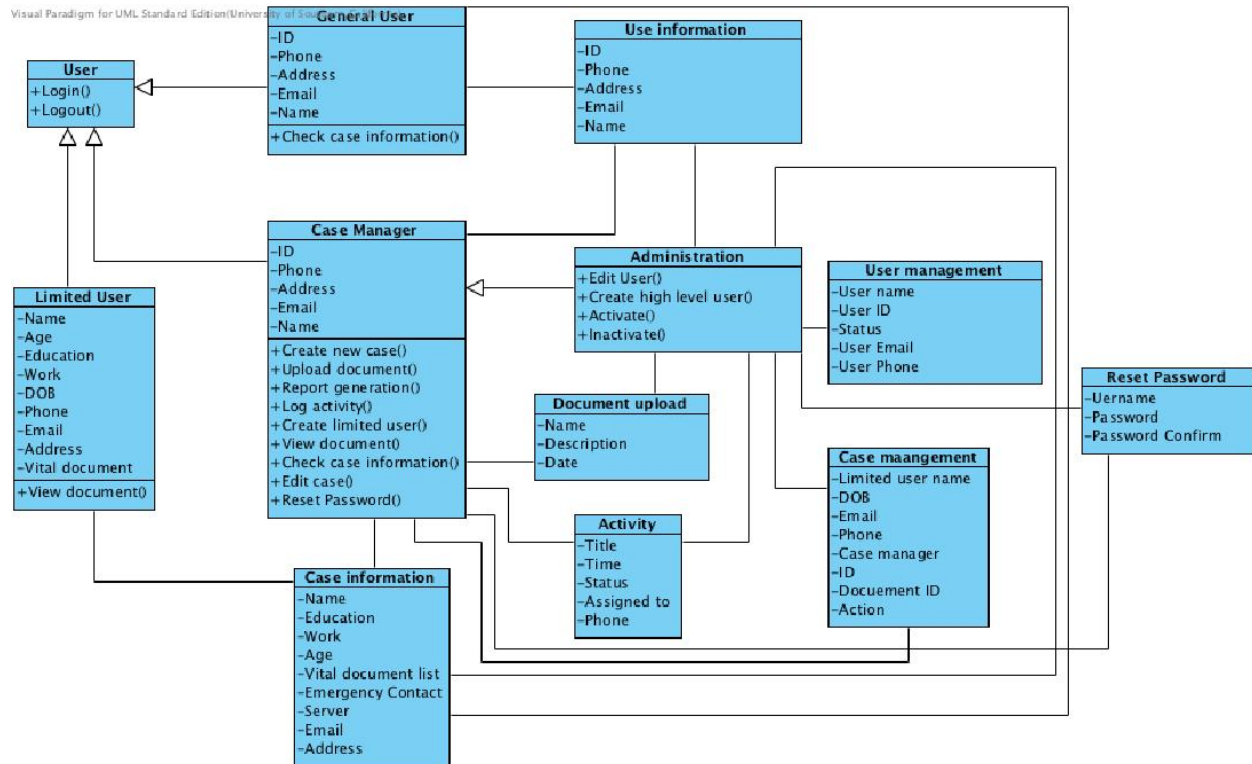


Figure 7: Design Class Diagram

Table 5: Design Class Description

Class	Type	Description
User		Any user can login and logout of the system
Limited User		Limited user will have his basic information and he only can view document.
General User		General user has his basic information and he has limited access to the system
Case manager		Case manager will has his basic information and some authorized access to the system
Case information		This class will help limited user class to record all necessary information of him.
User information		This class will help general user, case manager and administrator to record necessary information.
Administrator		Administrator inherits from case manager

		with adding more access to the system.
Document upload		This class will help to record necessary information when uploading a document.
Activity		This class will log information of an activity.
User management		This class will be used in user management.
Case management		This class will be used in case management.
Reset password		This class will be used to reset password.

## 3.2 Design Rationale

We designed the system according to different level of access to the system of different kind of user. Each user will have its information, while limited user is the reason we building this system so they will have more specific information in the system. Other classes or pages are used to implement the different level of access and different kind of function of the system according to the requirement from clients.

We deploy the system according to the fact that clients may not be able to supply device to store real data in the system, so we will store the real data in the Amazon and the database we have are mainly contain the index of each kind of data.

We descript software and hardware in high-level abstraction, so we don't go into the detail of each part. For design class diagram, we just include some basic class in the system, which will be mainly used, and the whole system may be more complicated.

## 4. Technology-Specific System Design

### 4.1 Design Overview

#### 4.1.1 System Structure

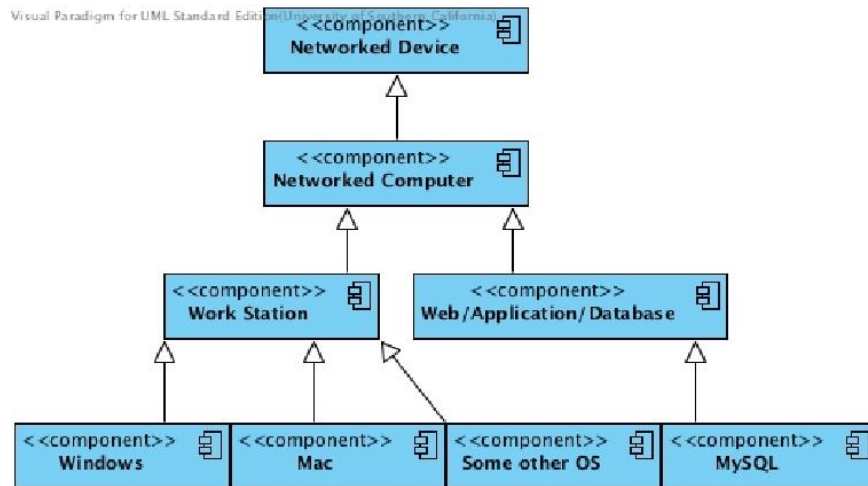


Figure 8: Hardware Component Class Diagram

Table 6: Hardware Component Description

Hardware Component	Description
Networked Computer	It is a server that receive requests from Client and response the request
Work Station	Workstation is a station that a client runs. It request to the server. It includes multiple kind of operating system with different kind of browsers like Google chrome.
Web/application/database	



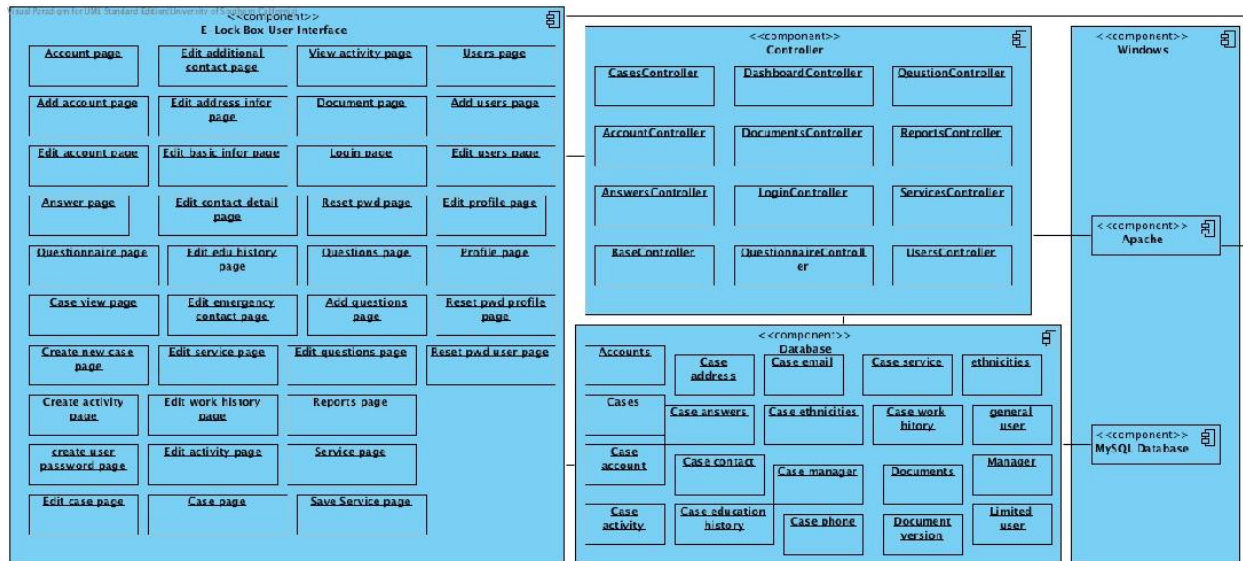


Figure 9: Software Component Class Diagram

Table 7: Software Component Description

Software Component	Description
Case management page	It contains a list of cases that a case manager is responsible for and also it's the main page when a case manage or administrator or general user login the system.
User management page	It contains a list of users in the system, including their users' id and status and so on. Only administrator can access it.
Case information page	It contains all necessary information the system need to use for a foster youth, including name, email, phone, address, and education history and so on.
Vital document preview page	It contains screenshot of vital document.
Report preview page	It contains screenshot of generated report.
Home page	It used to login the system by any users.
Search result page	It contains the result of a search.
Reset password page	It contains the necessary information for resetting password, including username, new password and password confirm etc.
User information page	It contains necessary information for a high-level user, which means administrator, case manager and general user, including name, phone, and email and so on.
Activity page	It contains the necessary information you need to type in to build a legal activity record.
Document upload page	It contains necessary description for uploaded document and some functions for choosing the document you want to upload.
Report generation page	It contains the type of report you want to generate.
Activity history page	It contains the history of the activities for a foster youth.
Work history page	It contains the history of education of a foster youth.

Case management controller	It used to control all case related workflow in the system. Any message that related to the case needs to deliver to it first.
Display	It used to display different kind of message or information.
User management controller	It used to control all user related workflow, which user only means high-level users.
Report controller	It used to generate report and interact with different kind of database.
Database: Documents	It contains all vital document index
Database: Cases	It contains all cases index
Database: Users	It contains all user index

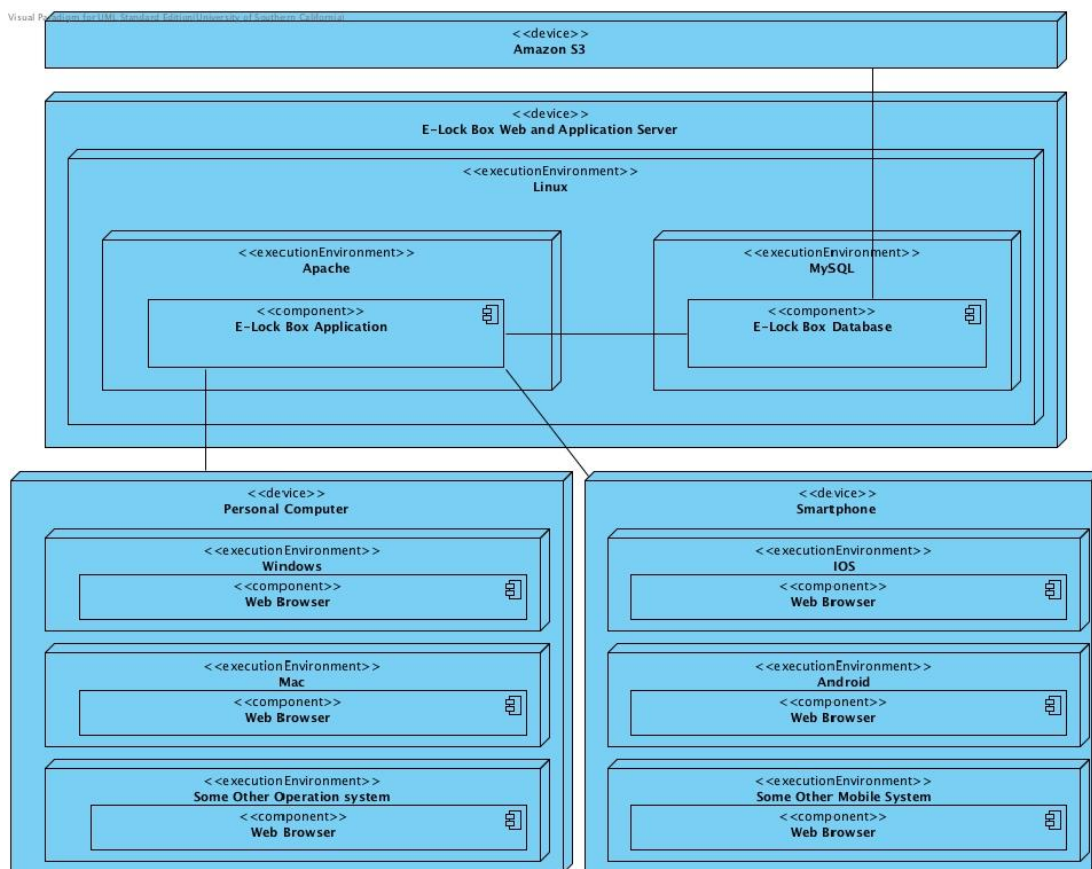


Figure 10: Deployment Diagram

## 4.1.2 Design Classes

### 4.1.2.1 <Classes n>

Visual Paradigm for UML Standard Edition (University of Southern California)

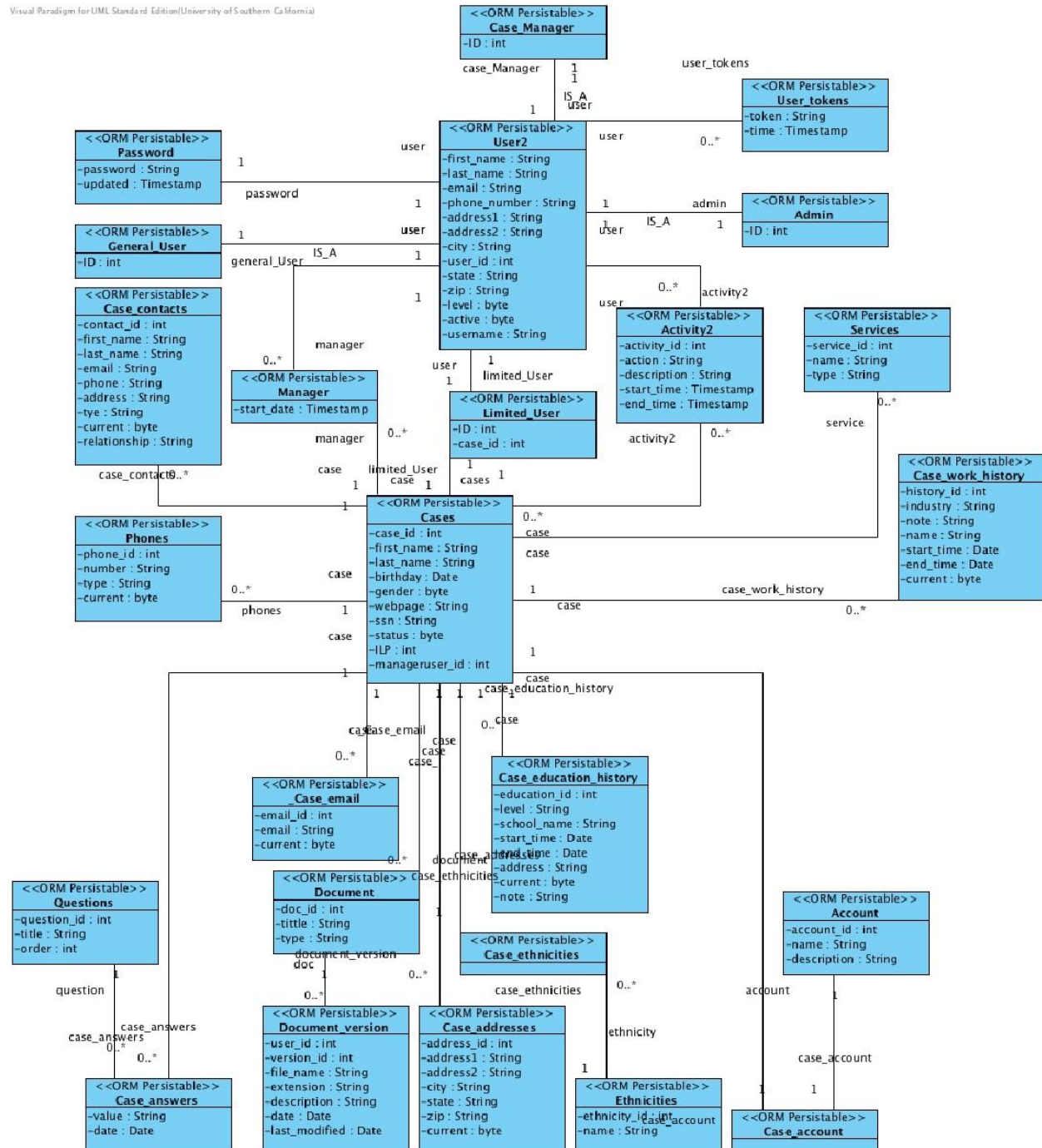


Figure 11: Design Class Diagram

**Table 8: Design Class Description**

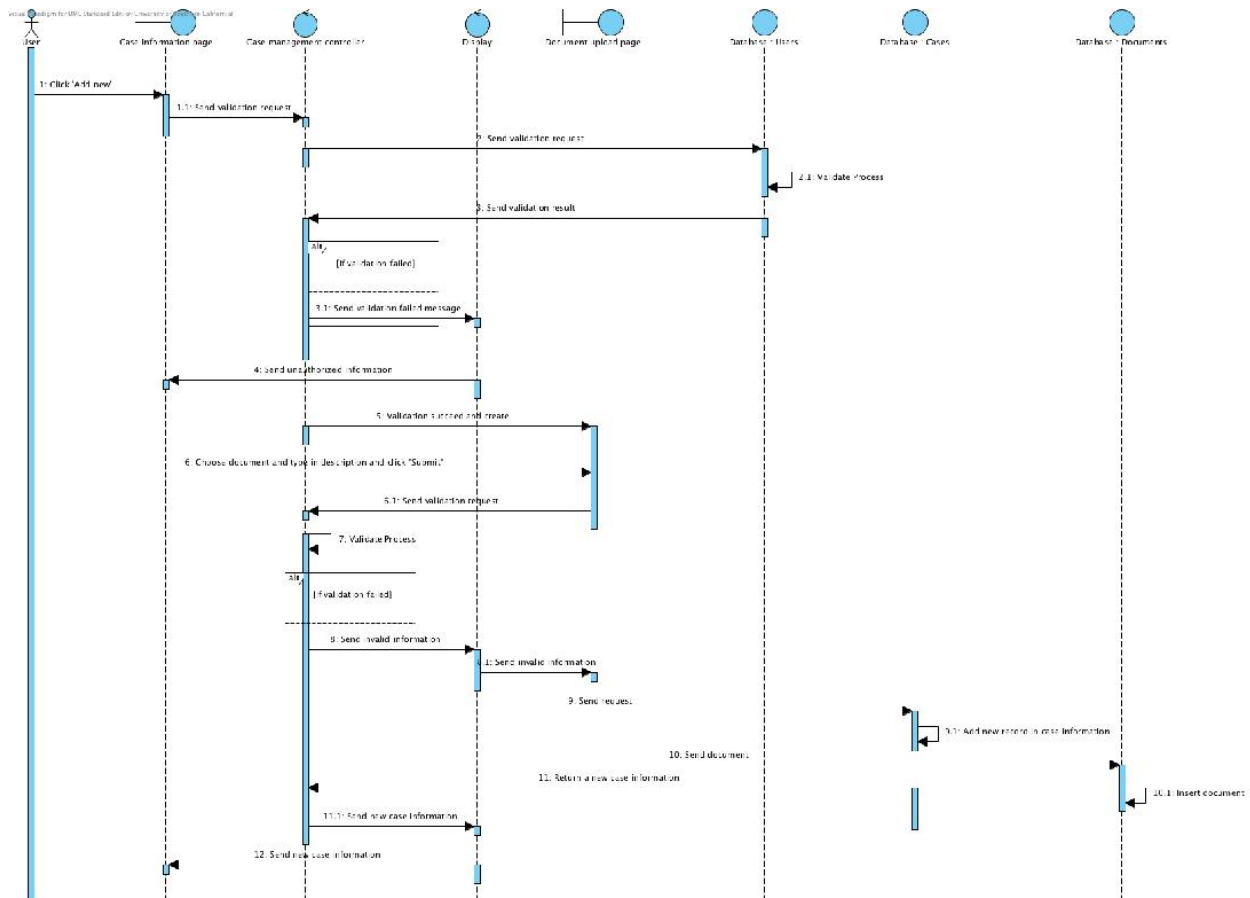
<b>Class</b>	<b>Type</b>	<b>Description</b>
User		Any user can login and logout of the system
Limited User		Limited user will have his basic information and he only can view document.
General User		General user has his basic information and he has limited access to the system
Case manager		Case manager will has his basic information and some authorized access to the system
Case information		This class will help limited user class to record all necessary information of him.
User information		This class will help general user, case manager and administrator to record necessary information.
Administrator		Administrator inherits from case manager with adding more access to the system.
Document upload		This class will help to record necessary information when uploading a document.
Activity		This class will log information of an activity.
User management		This class will be used in user management.
Case management		This class will be used in case management.
Reset password		This class will be used to reset password.
User management controller		This is used for dealing with all user related issue.
Case management controller		This is used for dealing with all case related issue
Report		This class will be used to generate report.
Report controller		This is used for generating report related issue
Database: Users		This is used to store user entity.
Database: Documents		This is used to store vital document entity.
Database: Cases		This is used to store case entity.

### 4.1.3 Process Realization

The process diagram will show some sequence diagram for most risky or important use cases in the system.

#### 4.1.3.1 Upload Document

This is one of most important use case in the system. If we want to store vital document from foster youth secure, we need this function to upload the document into a safe database.



**Figure 12: Process Realization Diagram: Upload Document**

#### 4.1.3.2 Create new case

For this use case, it's one of the highest priorities in the system. When we want to help any foster youth, we need to create a case for him or her and then track case information. We can have a case without create a related user, which means the related foster youth may not be a limited user in the system but once they become a limited user in the system, they will be connected to a exist case.

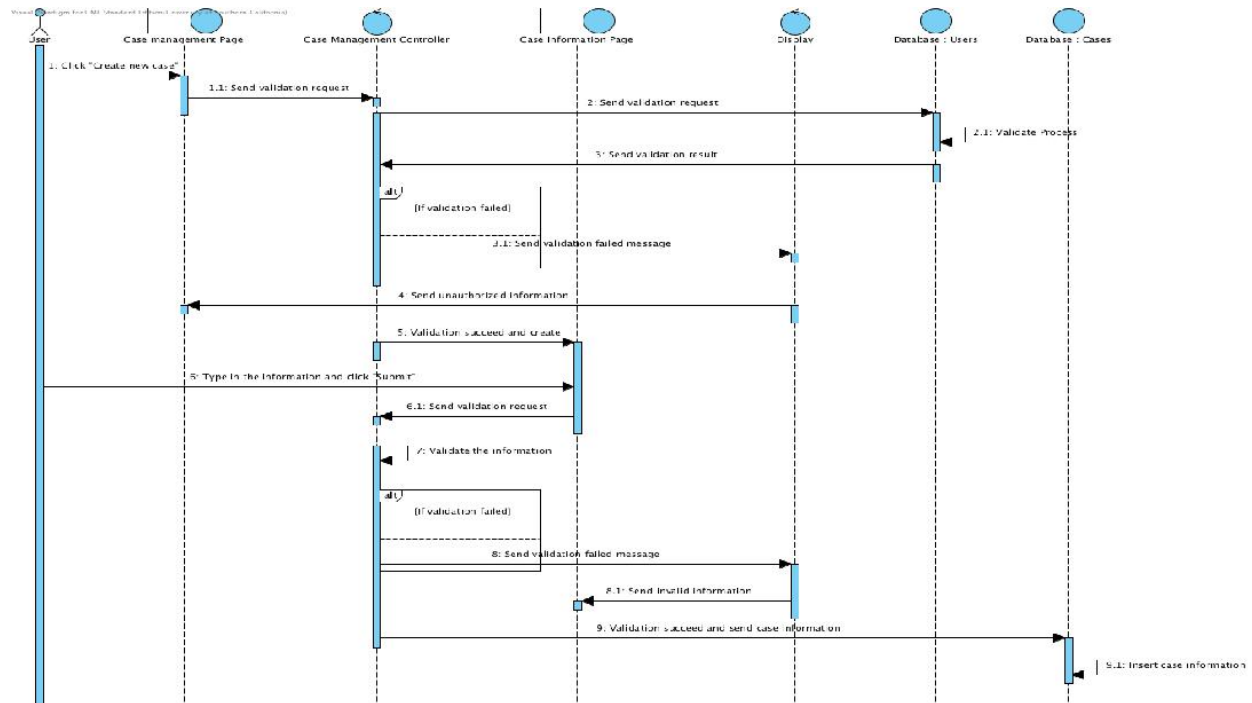


Figure 16: Process Realization Diagram: Create new case

#### 4.1.3.3 Create high-level user

Create a high level user just like create a case for foster youth, this is the only way to add more people to help foster youth. Also, it's one of the most priority functions.

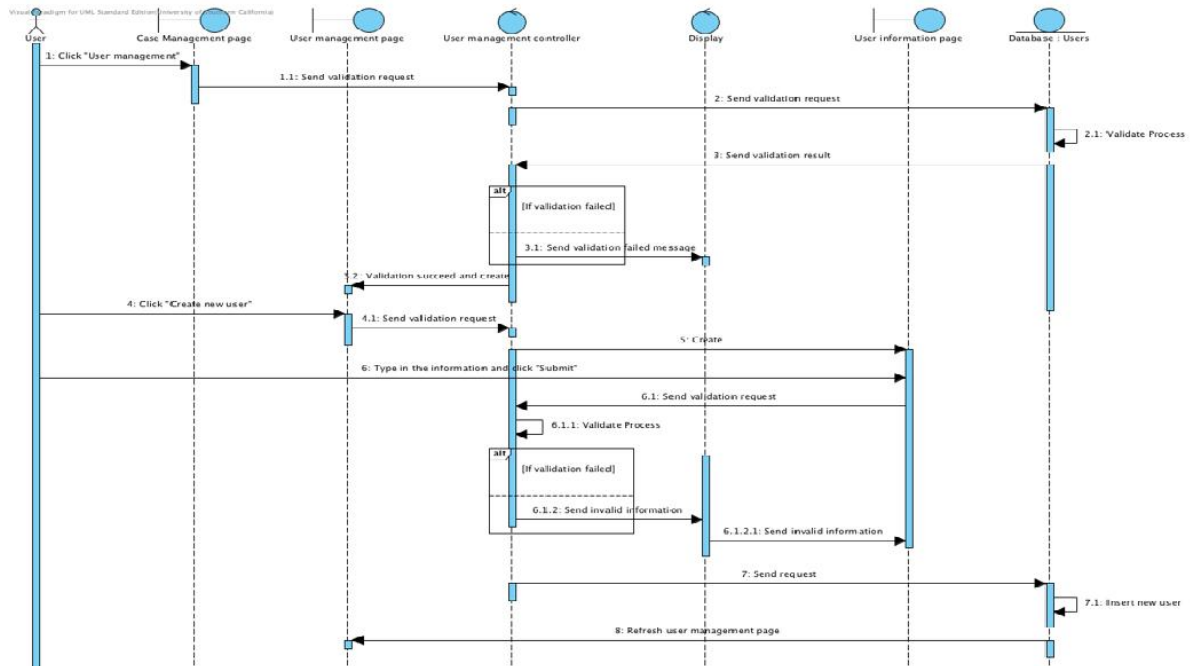


Figure 17: Process Realization Diagram: Create high-level user

## 4.2 Design Rationale

We designed the system according to different level of access to the system of different kind of user. Each user will have its information, while limited user is the reason we building this system so they will have more specific information in the system. Other classes or pages are used to implement the different level of access and different kind of function of the system according to the requirement from clients.

We deploy the system according to the fact that clients may not be able to supply device to store real data in the system, so we will store the real data in the Amazon and the database we have are mainly contain the index of each kind of data.

We descript software and hardware in high-level abstraction, so we don't go into the detail of each part. For design class diagram, we just include some basic class in the system, which will be mainly used, and the whole system may be more complicated.

## 5. Architectural Styles, Patterns and Frameworks

**Table 9: Architectural Styles, Patterns, and Frameworks**

Name	Description	Benefits, Costs, and Limitations
MVC Architecture	<p>Model–view–controller (MVC) is a software architecture pattern, which separates the representation of information from the user's interaction with it. It include model, view and controller in this architecture, following are the descriptions of each:</p> <p><b>Controller:</b> A controller can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model.</p> <p><b>Model:</b> A model notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them.</p> <p><b>View:</b> View requests information from the model that it needs for generating an output representation to the user.</p>	<p><b>Benefits:</b></p> <ol style="list-style-type: none"> <li>1. Separation of concerns: The separation the three components, allows the re-use of the business logic across applications. Multiple User Interfaces can be developed without concerning the codebase.</li> <li>2. Developer specialization and focus: The developers of UI can focus exclusively on the UI screens without bogged down with business logic.</li> </ol> <p><b>Costs:</b> There is no specific cost required to use MVC architecture.</p> <p><b>Limitations:</b></p> <ol style="list-style-type: none"> <li>1. Because of the decoupling of MVC, it may increase complexity of the system.</li> <li>2. MVC need multiple programmer and knowledge of multiple programming.</li> <li>3. Inefficiency of data access in view.</li> </ol>