# System and Software Architecture Description (SSAD)

**<e-Lockbox>**

**<Team 10>**

**<Team members and roles>**

|  |  |
| --- | --- |
| **Cheng Zhang** | **Prototyper / Chief Developer** |
| **Guancheng Li** | **Project Manager/ Prototyper/ Developer** |
| **Yutong Guo** | **Operational Concept Engineer/ Developer/ Tester** |
| **Qing Wei** | **Life Cycle Planner/ Feasibility Analyst/ Developer** |
| **Si Zhao** | **Software Architect/ Tester** |

**<04/27/17>**

# Version History

| Date | Author | Version | Changes made | Rationale |
| --- | --- | --- | --- | --- |
| 10/12/16 | Si Zhao | 1.0 | * Finished section 1 and section 2 of the report based on previous team’s work | * Initial draft for current project |
| 12/02/16 | Si Zhao | 2.0 | * Finished the first three sections of this report. | * Major description of our project. |
| 12/03/16 | Si Zhao | 2.1 | * Finished all sections of this report. | * Complete version of SSAD description. |
| 12/04/16 | Si Zhao | 2.2 | * Redesigned class diagram and use case diagram, and modified corresponding information in tables. | * Corrected a mistake about usage of the realization arrow in previous class diagram, and separated previous single use case diagram to several ones in order to prevent containing too much information. |
| 02/12/17 | Yutong Guo | 3.0 | * Revised use cases, software components and classes. | * RDC Package with v3.0 |
| 04/27/17 | YG, GL, CZ, QW | 4.0 | * Revised system context, artifacts and information * Updated use cases * Revised system structure, design classes and process realization | * As-Built Package with v4.0 |

# Table of Contents

System and Software Architecture Description (SSAD) i

Version History ii

Table of Contents iii

Table of Tables iv

Table of Figures vii

1. Introduction 1

1.1 Purpose of the SSAD 1

1.2 Status of the SSAD 1

2. System Analysis 2

2.1 System Analysis Overview 2

2.2 System Analysis Rationale 32

3. Technology-Independent Model 33

4. Technology-Specific System Design 34

4.1 Design Overview 34

4.2 Design Rationale 53

5. Architectural Styles, Patterns and Frameworks 54

# Table of Tables

[Table 1: Actors Summary 3](#_Toc481103832)

[Table 2: Artifacts and Information Summary 4](#_Toc481103833)

[Table 3: Process Description: User Login and Generating Verification Code 9](#_Toc481103834)

[Table 4: Typical Course of Action 9](#_Toc481103835)

[Table 5: Process Description: Create a New Case 10](#_Toc481103836)

[Table 6: Typical Course of Action 10](#_Toc481103837)

[Table 7: Process Description: View Case 10](#_Toc481103838)

[Table 8: Typical Course of Action 11](#_Toc481103839)

[Table 9: Process Description: Edit Case 11](#_Toc481103840)

[Table 10: Typical Course of Action 12](#_Toc481103841)

[Table 11: Process Description: Delete Case 12](#_Toc481103842)

[Table 12: Typical Course of Action 13](#_Toc481103843)

[Table 13: Process Description: Create Account for Youth 13](#_Toc481103844)

[Table 14: Typical Course of Action 14](#_Toc481103845)

[Table 15: Process Description: Inactivate Case 14](#_Toc481103846)

[Table 16: Typical Course of Action 14](#_Toc481103847)

[Table 17: Process Description: Activate Case 15](#_Toc481103848)

[Table 18: Typical Course of Action 15](#_Toc481103849)

[Table 19: Process Description: Create High-Level User 16](#_Toc481103850)

[Table 20: Typical Course of Action 16](#_Toc481103851)

[Table 21: Process Description: View User 16](#_Toc481103852)

[Table 22: Typical Course of Action 17](#_Toc481103853)

[Table 23: Process Description: Edit Own Profile 17](#_Toc481103854)

[Table 24: Typical Course of Action 18](#_Toc481103855)

[Table 25: Process Description: Inactivate User 18](#_Toc481103856)

[Table 26: Typical Course of Action 18](#_Toc481103857)

[Table 27: Process Description: Activate User 19](#_Toc481103858)

[Table 28: Typical Course of Action 19](#_Toc481103859)

[Table 29: Process Description: Create a New Activity 20](#_Toc481103860)

[Table 30: Typical Course of Action 20](#_Toc481103861)

[Table 31: Process Description: View Activity 20](#_Toc481103862)

[Table 32: Typical Course of Action 21](#_Toc481103863)

[Table 33: Process Description: Edit Activity 21](#_Toc481103864)

[Table 34: Typical Course of Action 22](#_Toc481103865)

[Table 35: Process Description: Delete Activity 22](#_Toc481103866)

[Table 36: Typical Course of Action 22](#_Toc481103867)

[Table 37: Process Description: Upload Vital Document 23](#_Toc481103868)

[Table 38: Typical Course of Action 23](#_Toc481103869)

[Table 39: Process Description: Download Document 24](#_Toc481103870)

[Table 40: Typical Course of Action 24](#_Toc481103871)

[Table 41: Process Description: Set Doc Invisible 25](#_Toc481103872)

[Table 42: Typical Course of Action 25](#_Toc481103873)

[Table 43: Process Description: Set Doc Visible 26](#_Toc481103874)

[Table 44: Typical Course of Action 26](#_Toc481103875)

[Table 45: Process Description: Send Email 27](#_Toc481103876)

[Table 46: Typical Course of Action 27](#_Toc481103877)

[Table 47: Process Description: Set Program Type 27](#_Toc481103878)

[Table 48: Typical Course of Action 28](#_Toc481103879)

[Table 49: Process Description: Set Document Type 28](#_Toc481103880)

[Table 50: Typical Course of Action 28](#_Toc481103881)

[Table 51: Process Description: Reset Any User’s Password 29](#_Toc481103882)

[Table 52: Typical Course of Action 29](#_Toc481103883)

[Table 53: Process Description: Set Survey Link 30](#_Toc481103884)

[Table 54: Typical Course of Action 30](#_Toc481103885)

[Table 55: Process Description: 10 Minutes Auto Logout 30](#_Toc481103886)

[Table 56: Typical Course of Action 31](#_Toc481103887)

[Table 57: Process Description: Reset Password for Self 31](#_Toc481103888)

[Table 58: Typical Course of Action 31](#_Toc481103889)

[Table 59: Hardware Component Description 35](#_Toc481103890)

[Table 60: Software Component Description 35](#_Toc481103891)

[Table 61: Design Class Description 1 38](#_Toc481103892)

[Table 62: Design Class Description 2 39](#_Toc481103893)

[Table 63: Design Class Description 3 39](#_Toc481103894)

[Table 64: Design Class Description 4 40](#_Toc481103895)

[Table 65: Design Class Description 5 41](#_Toc481103896)

[Table 66: Design Class Description 6 42](#_Toc481103897)

[Table 67: Design Class Description 7 42](#_Toc481103898)

[Table 68: Design Class Description 8 43](#_Toc481103899)

[Table 69: Design Class Description 9 45](#_Toc481103900)

[Table 70: Design Class Description 10 47](#_Toc481103901)

[Table 71: Design Class Description 11 48](#_Toc481103902)

[Table 72: Design Class Description 12 49](#_Toc481103903)

[Table 73: Design Class Description 13 50](#_Toc481103904)

[Table 74: Architectural Styles, Patterns, and Frameworks 54](#_Toc481103905)

# Table of Figures

[Figure 1: System Context Diagram 2](#_Toc481103910)

[Figure 2: Artifacts and Information Diagram 4](#_Toc481103911)

[Figure 3: Process Diagram – Overall 6](#_Toc481103912)

[Figure 4: Process Diagram – Admin 7](#_Toc481103913)

[Figure 5: Process Diagram – Case Manager 8](#_Toc481103914)

[Figure 6: Process Diagram – Staff 8](#_Toc481103915)

[Figure 7: Process Diagram – Youth 9](#_Toc481103916)

[Figure 8: Hardware Component Class Diagram 34](#_Toc481103917)

[Figure 9: Software Component Class Diagram 34](#_Toc481103918)

[Figure 10: Deployment Diagram 35](#_Toc481103919)

[Figure 11: Design Class Diagram 1 38](#_Toc481103920)

[Figure 12: Design Class Diagram 2 39](#_Toc481103921)

[Figure 13: Design Class Diagram 3 39](#_Toc481103922)

[Figure 14: Design Class Diagram 4 40](#_Toc481103923)

[Figure 15: Design Class Diagram 5 41](#_Toc481103924)

[Figure 16: Design Class Diagram 6 42](#_Toc481103925)

[Figure 17: Design Class Diagram 7 42](#_Toc481103926)

[Figure 18: Design Class Diagram 8 43](#_Toc481103927)

[Figure 19: Design Class Diagram 9 45](#_Toc481103928)

[Figure 20: Design Class Diagram 10 47](#_Toc481103929)

[Figure 21: Design Class Diagram 11 48](#_Toc481103930)

[Figure 22: Design Class Diagram 12 49](#_Toc481103931)

[Figure 23: Design Class Diagram 13 50](#_Toc481103932)

[Figure 24: Sequence Diagram – Login 51](#_Toc481103933)

[Figure 25: Sequence Diagram - Create Case 52](#_Toc481103934)

[Figure 26: Sequence Diagram – Create High Level User 52](#_Toc481103935)

### Introduction

#### Purpose of the SSAD

The SSAD document describes the architecture of e-Lockbox system so as to provide reference for developers and users. The content includes diagrams of system organization and operations according to requirements and operational concepts.

#### Status of the SSAD

This SSAD is version 4.0 for As-Built Package. We have redesigned system context, artifacts and information, use cases, system structure, design classes and process realization. A lot of changes have made compared to previous version.

### System Analysis

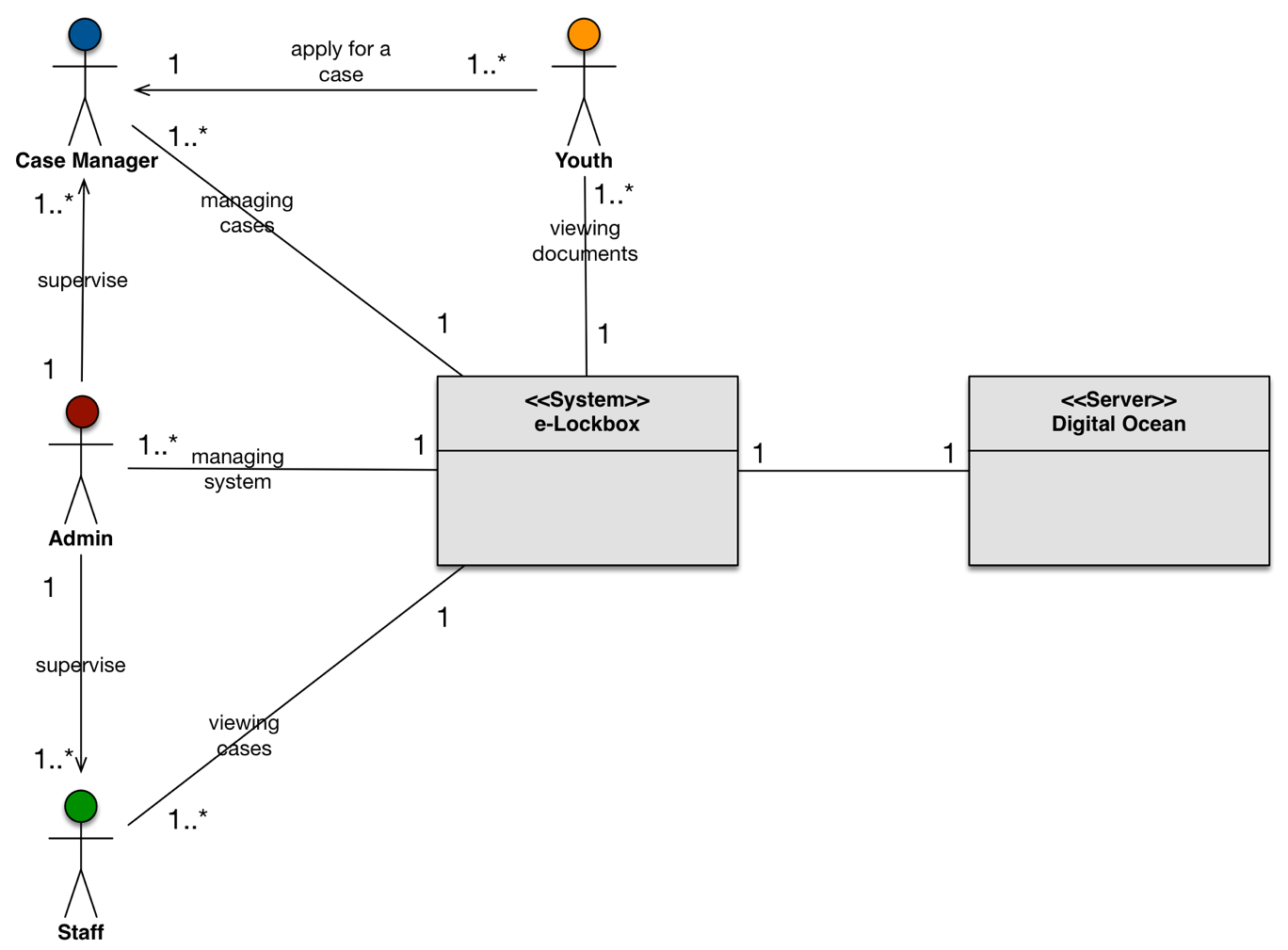
#### System Analysis Overview

The primary purpose of the e-Lockbox (the Living Advantage Inc’s website) is to provide a highly secure system that stores and provides access to underprivileged youths’ vital identity documents such as birth certificate, social security card, and driver’s license.

There are case managers who can create accounts for youths and help them upload these documents. The system can keep track of all information about what activities they have done, such as storing personal information and phone calls.

Administrator who has the highest authority can manage all users, cases, documents, activities and settings in the system. Case Managers can manage youth users, cases, documents and activities under their charge. Staff can only view user information, case information of youth and activities. Youth can view their own case and documents and take survey.

##### System Context

****

**Figure 1: System Context Diagram**

Table : Actors Summary

| **Actor** | **Description** | **Responsibilities** |
| --- | --- | --- |
| Administrator | Users who has the top authority to manage the whole system | Manage all users, cases, documents, activities and settings. |
| Case Manager | Users who is mainly responsible for youth users | Help youths with documents, activities.  Manage associated cases.  Keep track of responsible youth users. |
| Staff | Users who has limited authority to the system | View user information, case information of youth and activities. |
| Youth | Users who can only access to their own case and documents | View their own case and documents.  Take survey. |

##### Artifacts & Information

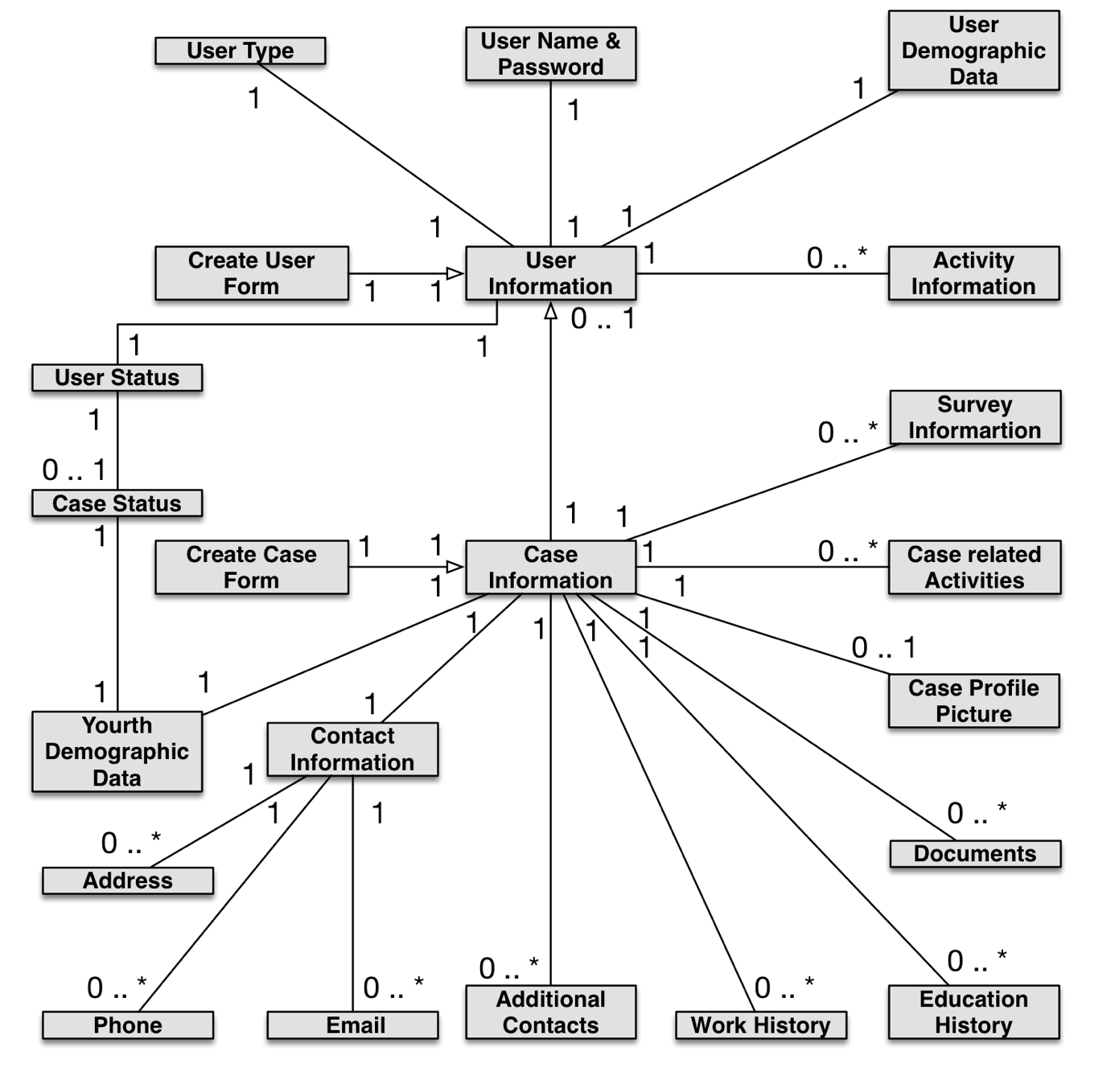


Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

|  |  |
| --- | --- |
| **Artifact** | **Purpose** |
| Create User Form | It provides a form to fill up the information on the users except the youth user. |
| Create Case Form | It provides a form to fill up the information on the case of a youth. |
| User Information | It contains all the information about a user. |
| Case Information | It contains all the information about a case. |
| User name & Password | It stores the login information for each user. |
| User Type | It Indicates which type is the user belongs to among admin, case manager, staff or youth user. |
| User Demographic Data | It contains all the demographic information about a user. |
| User Status | It indicates the status of the user, which could be in active or inactive state for a user. |
| Activity Information | It contains all the activity information related to a user. |
| Youth Demographic Data | It contains all the demographic information about a case, includes name, telephone, email, SSN and etc. |
| Case Status | It indicates the status of the case, which could be in active or inactive state for a user. |
| Contact Information | It contains the contact information about the case. |
| Additional Contacts | It contains the additional contacts information about the case, includes email, address and phone. |
| Address | It contains the additional address for the additional contacts of the case. |
| Phone | It contains the additional phone numbers for the additional contacts of the case. |
| Email | It contains the additional emails for the additional contacts of the case. |
| Work History | It indicates the youth’s work history is if applicable. |
| Education History | It indicates the youth’s education history if applicable. |
| Documents | It stores the documents uploaded under the case. |
| Case Profile Picture | It stores the case profile picture for the youth user. |
| Case Related Activities | It contains all the information about the activities related to the case. |
| Survey Information | It contains the survey description and the link to the survey. |

##### Behavior

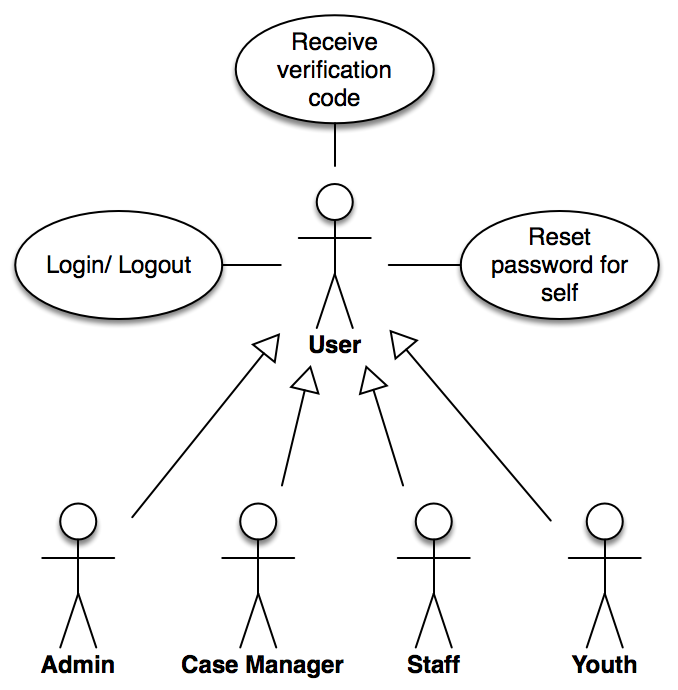


Figure : Process Diagram – Overall

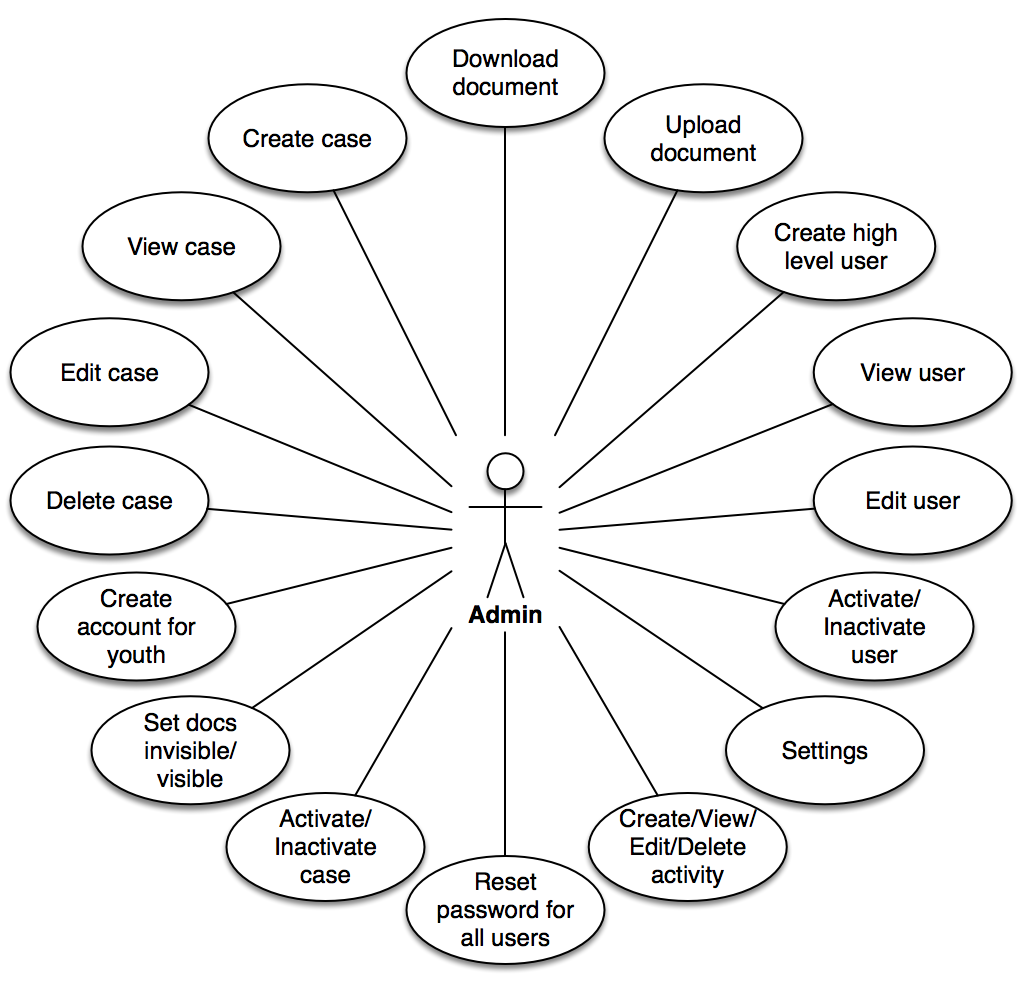


Figure : Process Diagram – Admin

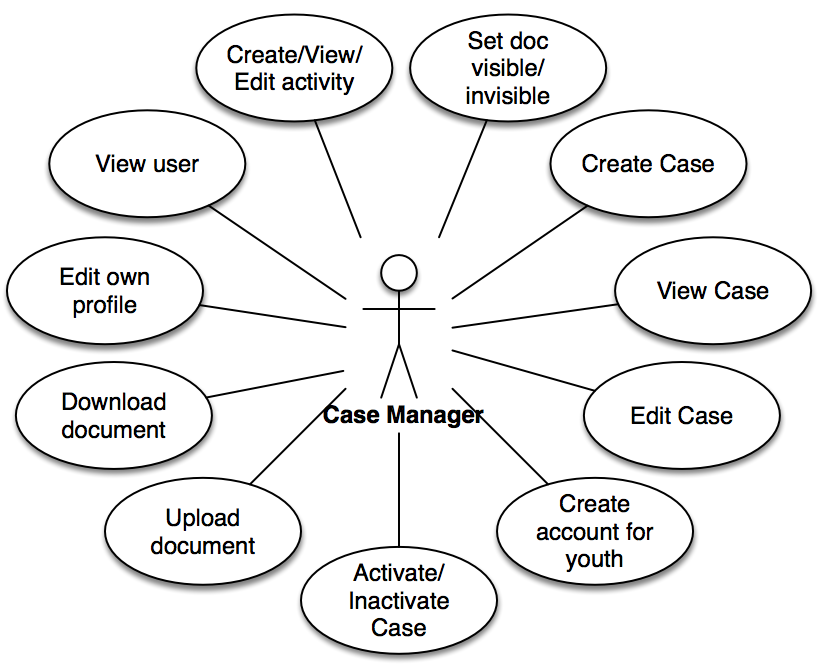


Figure : Process Diagram – Case Manager

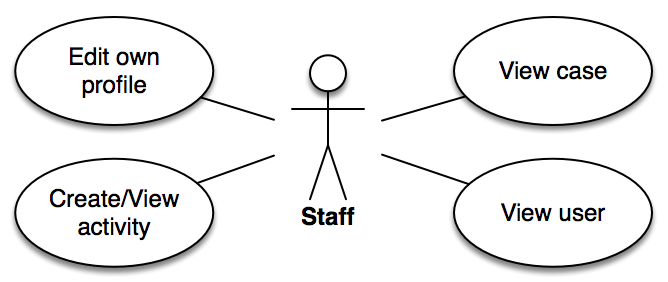


Figure : Process Diagram – Staff

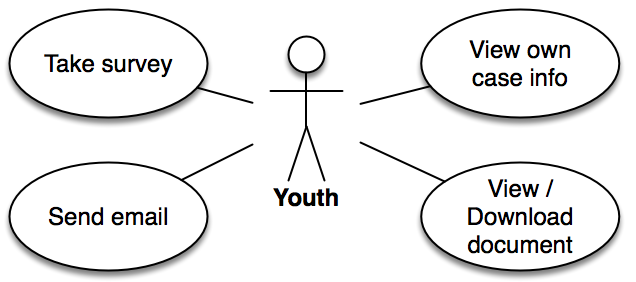


Figure : Process Diagram – Youth

###### 2.1.3.1 Capabilities

2.1.3.1.1 Process 1:

Table : Process Description: User Login and Generating Verification Code

|  |  |
| --- | --- |
| **Identifier** | UC-1: User Login and Generating Verification Code |
| **Purpose** | Provide login system that after user inputting name and password, check user’s active status and generate verification code sent by email, and then allow user to login |
| **Requirements** | WC\_3935; WC\_3978; WC\_3996; WC4005 |
| **Development Risks** | Team members don’t have related developing experience |
| **Pre-conditions** | 1. The user email is existing in database, and password match corresponding entry 2. The status of user should be active 3. Inputted verification code matches the one sent through email |
| **Post-conditions** | None |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [All users]  Type in username and password, then click "Log in" |  |
| **2** |  | Validate username and password |
| **3** |  | If validation failed, return to login page with error message |
| **4** |  | If validate succeed, check the status of user to see whether it is active |
| **5** |  | If the user is not active, return to login page with error message |
| **6** |  | If the user is active, generate verification code, send email containing code, encrypt and store code in DB |
| **7** | Input verification code |  |
| **8** |  | Check whether the inputted verification code is right |
| **9** |  | If the code is wrong, return to login page with error message |
| **10** |  | If the code is right, redirect to user’s corresponding page according to user’s role |

2.1.3.1.2 Process 2:

Table : Process Description: Create a New Case

|  |  |
| --- | --- |
| **Identifier** | UC-2: Create a New Case |
| **Purpose** | Create a new case for a new youth user in the system. The new case will include the basic information about this youth. |
| **Requirements** | WC\_3890; WC\_3937 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator and case manager can create a new case 2. Administrator or case manager has logged into the system |
| **Post-conditions** | A new case will be created into the database and showed on the brief and detailed case page |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator or case manager]  Click "Create Case" |  |
| **2** |  | Verify the user if he/she is an administrator, case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to a case form page |
| **5** | Type in the information and click "Create Case" |  |
| **6** |  | Validate the information from case form page |
| **7** |  | If validate failed, show error message |
| **8** |  | If validate succeed, insert case information into case database and show succeed message |

2.1.3.1.3 Process 3:

Table : Process Description: View Case

|  |  |
| --- | --- |
| **Identifier** | UC-3: View Case |
| **Purpose** | Administrator, case manager and staff can view case information in brief and detail |
| **Requirements** | WC\_3896; WC\_3897; WC\_3941; WC\_3942; WC\_3980; WC\_3981 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator, case manager and staff can view case information 2. Administrator, case manager or staff has logged into the system |
| **Post-conditions** | System will return the page of brief case information and each authorized user can view the information even go into the detail |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager or staff]  Click “View Cases” |  |
| **2** |  | Verify the user if he/she is an administrator, case manager or staff |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to brief case information page |
| **5** | Click the view button of specific case |  |
| **6** |  | Verify the user if he/she is an administrator, case manager or staff |
| **7** |  | If verify fail, return to fail page |
| **8** |  | If verify success, return to detailed case information page |

2.1.3.1.4 Process 4:

Table : Process Description: Edit Case

|  |  |
| --- | --- |
| **Identifier** | UC-4: Edit Case |
| **Purpose** | If some information of a youth has changed, administrator and case manager can edit the case information of the youth |
| **Requirements** | WC\_3891; WC\_3938 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Case should exist in the system 2. Case should be in status of active 3. Administrator or case manager has logged into the system |
| **Post-conditions** | Updated case information will be refreshed into the database and showed on the brief and detailed case information page |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator or case manager]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator or case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Click "Edit" |  |
| **6** |  | Show a pop-up window with current case information |
| **7** | Type in the changed case information and click "Save" |  |
| **8** |  | Validate the information from detailed case information page |
| **9** |  | If validate failed, show error message |
| **10** |  | If validate succeed, refresh the new information into database |
| **11** |  | Return to an updated detailed case information page |

2.1.3.1.5 Process 5:

Table : Process Description: Delete Case

|  |  |
| --- | --- |
| **Identifier** | UC-5: Delete Case |
| **Purpose** | If a case is no longer needed, administrator can delete the case of youth |
| **Requirements** | WC\_3892 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator can delete case 2. Case should exist in the system 3. Administrator has logged into the system |
| **Post-conditions** | The case and all the case related information will be deleted from database and brief case information page |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Click "Delete" |  |
| **6** |  | Show a confirmation pop-up window for deleting |
| **7** | Type in the name of youth in this case |  |
| **8** |  | Validate the input from the confirmation pop-up window |
| **9** |  | If validate failed, the red button will be not able to click |
| **10** |  | If validate succeed, the red button will be able to click |
| **11** | Click the red button |  |
| **12** |  | Delete this case from database |
| **13** |  | Return to a brief case page without the deleted case |

2.1.3.1.6 Process 6:

Table : Process Description: Create Account for Youth

|  |  |
| --- | --- |
| **Identifier** | UC-6: Create Account for Youth |
| **Purpose** | Administrator and case manager can create an account for youth so that a youth could have an username and password to log into the system |
| **Requirements** | WC\_4008; WC\_3965 |
| **Development Risks** | None |
| **Pre-conditions** | 1. The case for youth should be created 2. Create a youth account can only be done in the case information page 3. Administrator or case manager has logged into the system |
| **Post-conditions** | A new user will be generated and the username and password will be recorded into the database. |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator or case manager]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator or case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Click "Create Account" |  |
| **6** |  | Show a pop-up window for inputting password for the username, which is email |
| **7** | Type in the password twice and click “Create and Activate” |  |
| **8** |  | Validate the two passwords |
| **9** |  | If validate failed, show error message |
| **10** |  | If validate succeed, insert the new username and password into the user database and hide the “Create Account” button |

2.1.3.1.7 Process 7:

Table : Process Description: Inactivate Case

|  |  |
| --- | --- |
| **Identifier** | UC-7: Inactivate Case |
| **Purpose** | If a case needs to suspend temperately in the system, administrator and case manager can inactivate the case |
| **Requirements** | WC\_3894; WC\_3940 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Case should exist in the system 2. Case should in the status of active 3. Administrator or case manager has logged into the system |
| **Post-conditions** | A case will be inactive in the system and this case can’t be edited. |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator or case manager]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator or case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Click "Inactivate" |  |
| **6** |  | Change the status of case in the database into inactive and show on the brief and detailed case information page |

2.1.3.1.8 Process 8:

Table : Process Description: Activate Case

|  |  |
| --- | --- |
| **Identifier** | UC-8: Activate Case |
| **Purpose** | If some temperate suspend cases need to come back, administrator can activate corresponding case |
| **Requirements** | WC\_3893; WC\_3939 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Case should exist in the system 2. Case should be inactivated 3. Only administrator and case manager can activate an inactivated case 4. Administrator or case manager has logged into the system |
| **Post-conditions** | A case will be successful activated in the system |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator or case manager]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator or case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Click "Activate" |  |
| **6** |  | Change the status of case in the database into active and show on the brief and detailed case information page |

2.1.3.1.9 Process 9:

Table : Process Description: Create High-Level User

|  |  |
| --- | --- |
| **Identifier** | UC-9: Create High-Level User |
| **Purpose** | Only administrator can create another administrator as well as case manager and staff |
| **Requirements** | WC\_3920 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator can create high-level user 2. Administrator has logged into the system 3. Create high-level user can only happen in the user management system |
| **Post-conditions** | A new user will be created in the system and database will get the record of this user’s username, password and some basic information such as name, phone, address and level |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Click " Create User" |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to a user form page |
| **5** | Type in the information and click "Create and Activate Account" |  |
| **6** |  | Validate the information from user form page |
| **7** |  | If validate failed, show error message |
| **8** |  | If validate succeed, insert the user information into user database and show succeed message |

2.1.3.1.10 Process 10:

Table : Process Description: View User

|  |  |
| --- | --- |
| **Identifier** | UC-10: View User |
| **Purpose** | Administrator, case manager and staff can view user information in brief and detail |
| **Requirements** | WC\_3922; WC\_3923; WC\_3967; WC\_3968; WC\_3987; WC\_3988 |
| **Development Risks** | None |
| **Pre-conditions** | 1. User should exist in the system 2. Administrator, case manager or staff has logged into the system |
| **Post-conditions** | System will return the page of brief user information and each authorized user can view the information even go into the detail |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager, staff]  Click " View Users" |  |
| **2** |  | Verify the user if he/she is an administrator, case manager or staff |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to brief user information page |
| **5** | Click the view button of specific user |  |
| **6** |  | Verify the user if he/she is an administrator, case manager or staff |
| **7** |  | If verify fail, return to fail page |
| **8** |  | If verify success, return to detailed user information page |

2.1.3.1.11 Process 11:

Table : Process Description: Edit Own Profile

|  |  |
| --- | --- |
| **Identifier** | UC-11: Edit Own Profile |
| **Purpose** | If some information of user has changed, he/she can change the information of his/her own profile. Only administrator, case manager and staff can edit their own profile. |
| **Requirements** | WC\_3921; WC\_3966; WC\_4007 |
| **Development Risks** | None |
| **Pre-conditions** | 1. User account should exist in the system 2. Administrator, case manager or staff has logged into the system |
| **Post-conditions** | Updated user information will be refreshed into the database and showed on the brief and detailed user information page |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager or staff]  Click view button of their own account |  |
| **2** |  | Verify the user if he/she is an administrator, case manager or staff |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed user information page |
| **5** | Click "Edit" |  |
| **6** |  | Return to a detailed user profile page for editing |
| **7** | Type in changed information and click "Save" |  |
| **8** |  | Validate the information from detailed user profile page |
| **9** |  | If validate failed, show error message |
| **10** |  | If validate succeed, refresh new information into database |
| **11** |  | Return to an updated detailed user information page |

2.1.3.1.12 Process 12:

Table : Process Description: Inactivate User

|  |  |
| --- | --- |
| **Identifier** | UC-12: Inactivate User |
| **Purpose** | If a user is no longer in the system, administrator can inactivate the user. |
| **Requirements** | WC\_3924 |
| **Development Risks** | None |
| **Pre-conditions** | 1. User should exist in the system 2. User should be in the status of active 3. Administrator has logged into the system |
| **Post-conditions** | A user will be inactivated in the system, and only administrator can inactivate a user. And the inactivated user can’t log into the system. |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Click the view button of specific user |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed user information page |
| **5** | Click "Inactivate" |  |
| **6** |  | Change the status of user in the database into inactive and show on the brief and detailed user information page |

2.1.3.1.13 Process 13:

Table : Process Description: Activate User

|  |  |
| --- | --- |
| **Identifier** | UC-13: Activate User |
| **Purpose** | If some off-system users want to come back to the system, administrator can activate corresponding user. |
| **Requirements** | WC\_3923 |
| **Development Risks** | None |
| **Pre-conditions** | 1. User should exist in the system 2. User should in the status of inactive 3. Administrator has logged into the system |
| **Post-conditions** | A user will be successful activated in the system, and only administrator can activate a user. |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Click the view button of specific user |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed user information page |
| **5** | Click "Activate" |  |
| **6** |  | Change the status of user in the database into active and show on the brief and detailed user information page |

2.1.3.1.14 Process 14:

Table : Process Description: Create a New Activity

|  |  |
| --- | --- |
| **Identifier** | UC-14: Create a New Activity |
| **Purpose** | Create a new activity in the system. The new activity will include the recent operations or events related or unrelated to specific case. |
| **Requirements** | WC\_3926; WC\_3969; WC\_3989 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator, case manager and staff can create a new activity 2. Administrator, case manager or staff has logged into the system |
| **Post-conditions** | A new activity will be created into the database and showed on the dashboard |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager or staff]  Enter Dashboard and click "Create" |  |
| **2** |  | Verify the user if he/she is an administrator, case manager or staff |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to a activity form page |
| **5** | Type in the information and click "Save" |  |
| **6** |  | Validate the information from activity information page |
| **7** |  | If validate failed, show error message |
| **8** |  | If validate succeed, insert activity information into activity database and show succeed message |

2.1.3.1.15 Process 15:

Table : Process Description: View Activity

|  |  |
| --- | --- |
| **Identifier** | UC-15: View Activity |
| **Purpose** | Administrator, case manager and staff can view activity information in brief and detail |
| **Requirements** | WC\_3927; WC\_3928; WC\_3970; WC\_3971; WC\_3990; WC\_3991 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator, case manager and staff can view activity information 2. Administrator, case manager or staff has logged into the system |
| **Post-conditions** | System will return the page of brief activity information and each authorized user can view the information even go into the detail |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager or staff]  Enter Dashboard |  |
| **2** |  | Verify the user if he/she is an administrator, case manager or staff |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to brief activity information page |
| **5** | Click view button of specific activity |  |
| **6** |  | Verify the user if he/she is an administrator, case manager or staff |
| **7** |  | If verify fail, return to fail page |
| **8** |  | If verify success, return to detailed activity information page |

2.1.3.1.16 Process 16:

Table : Process Description: Edit Activity

|  |  |
| --- | --- |
| **Identifier** | UC-16: Edit Activity |
| **Purpose** | If some information of an activity has changed, administrator, case manager and staff can edit the activity information |
| **Requirements** | WC\_3929; WC\_3972; WC\_3992; |
| **Development Risks** | None |
| **Pre-conditions** | 1. Activity should exist in the system 2. Administrator, case manager has logged into the system |
| **Post-conditions** | Updated activity information will be refreshed into the database and showed on the dashboard and detailed activity information page |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager]  Click the view button of specific activity |  |
| **2** |  | Verify the user if he/she is an administrator or case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed activity information page |
| **5** | Click "Edit", type in the changed activity information and click "Save" |  |
| **6** |  | Validate the information from detailed activity information page |
| **7** |  | If validate failed, show error message |
| **8** |  | If validate succeed, refresh the new information into database |
| **9** |  | Return to an updated dashboard |

2.1.3.1.17 Process 17:

Table : Process Description: Delete Activity

|  |  |
| --- | --- |
| **Identifier** | UC-17: Delete Activity |
| **Purpose** | If an activity is no longer needed, administrator can delete the activity. |
| **Requirements** | WC\_3930 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator can delete an activity 2. Activity should exist in the system 3. Administrator has logged into the system |
| **Post-conditions** | The activity will be deleted from database and dashboard. |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Enter Dashboard |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to dashboard |
| **5** | Click delete button |  |
| **6** |  | Show a confirmation pop-up window for deleting |
| **7** | Click “Delete” |  |
| **8** |  | Delete this activity from database |
| **9** |  | Delete this case from database |
| **10** |  | Return to dashboard without the deleted activity |

2.1.3.1.18 Process 18:

Table : Process Description: Upload Vital Document

|  |  |
| --- | --- |
| **Identifier** | UC-18: Upload Vital Document |
| **Purpose** | Administrator or case manager can upload documents for youth in the system |
| **Requirements** | WC\_3914; WC\_3959 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator and case manager can upload documents 2. Administrator or case manager has logged into the system 3. Upload document function is only in case information page 4. The case should be in the status of active |
| **Post-conditions** | A new vital document will be uploaded into the system and a record of new uploaded document will appear in the Vital Document part of detailed case information page |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator or case manager]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator or case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Find Vital Document part and click “Add” |  |
| **6** |  | Show a pop-up window for uploading file |
| **7** | Type in basic information of the document, choose file and click "Upload File" |  |
| **8** |  | Validate the input from the pop-up window |
| **9** |  | If validate failed, show error message |
| **10** |  | If validate succeed, insert document basic information into document database and store document on the server |
| **11** |  | Return to the detailed case information page with new uploaded document record |

2.1.3.1.19 Process 19:

Table : Process Description: Download Document

|  |  |
| --- | --- |
| **Identifier** | UC-19: Download Document |
| **Purpose** | Administrator, case manager and youth can download the vital document in the system |
| **Requirements** | WC\_3917; WC\_3962; WC\_4002 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator can download all the documents 2. Case manager can download documents under his/her charge 3. Youth can download documents of his/her own 4. The documents should exist in the system 5. Administrator or case manager or youth has logged into the system |
| **Post-conditions** | Users can view or print the downloaded vital document |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager or youth]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator, case manager or youth |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Find Vital Document part and click view button or link on title |  |
| **6** |  | Validate the role of users |
| **7** |  | If validate failed, return to fail page |
| **8** |  | If validate succeed, show the document download page |
| **9** | Click “Save” |  |
| **10** |  | The document has been downloaded successfully and return to detailed case information page |

2.1.3.1.20 Process 20:

Table : Process Description: Set Doc Invisible

|  |  |
| --- | --- |
| **Identifier** | UC-20: Set Doc Invisible |
| **Purpose** | If some documents don’t be allowed to be viewed by youth, administrator and case manager can set them as invisible |
| **Requirements** | WC\_3919; WC\_3964 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Doc should exist in the system 2. Doc should be visible 3. Only administrator and case manager can set docs as invisible 4. Administrator or case manager has logged into the system |
| **Post-conditions** | This doc can’t be seen by corresponding youth in the system. |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator or case manager]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator or case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Find Vital Document part and click edit button |  |
| **6** |  | Show a pop-up window for editting |
| **7** | Change “visible” into “invisible”, click “Save” |  |
| **8** |  | Change the status of the doc in the database into invisible and show on the detailed case information page |

2.1.3.1.21 Process 21:

Table : Process Description: Set Doc Visible

|  |  |
| --- | --- |
| **Identifier** | UC-21: Set Doc Visible |
| **Purpose** | If some documents can be allowed to see by youth, administrator or case manager can set them as visible |
| **Requirements** | WC\_3918; WC\_3963 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Doc should exist in the system 2. Doc should be invisible 3. Only administrator and case manager can set docs as visible 4. Administrator or case manager has logged into the system |
| **Post-conditions** | This doc can be seen by corresponding youth in the system |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator or case manager]  Click the view button of specific case |  |
| **2** |  | Verify the user if he/she is an administrator or case manager |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to detailed case information page |
| **5** | Find Vital Document part and click edit button |  |
| **6** |  | Show a pop-up window for editting |
| **7** | Change “invisible” into “visible”, click “Save” |  |
| **8** |  | Change the status of the doc in the database into visible and show on the detailed case information page |

2.1.3.1.22 Process 22:

Table : Process Description: Send Email

|  |  |
| --- | --- |
| **Identifier** | UC-22: Send Email |
| **Purpose** | Youth can send email to his/her manager in the system |
| **Requirements** | WC\_4006 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Youth can access to his/her manager’s email 2. Youth has logged into the system |
| **Post-conditions** | Email will be sent to corresponding manager from specific youth |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Youth]  Click email icon |  |
| **2** |  | Verify the user if he/she is a youth |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, show a pop-up mailbox |
| **5** | Type in information and send the Email |  |
| **6** |  | Return to youth page |

2.1.3.1.23 Process 23:

Table : Process Description: Set Program Type

|  |  |
| --- | --- |
| **Identifier** | UC-23: Set Program Type |
| **Purpose** | Administrator can set program type in the system |
| **Requirements** | WC\_3931 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator can set program type in the system 2. Administrator has logged into the system |
| **Post-conditions** | Program type will be made as the selection of specific function and added to the database |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Click “Program Settings” |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to a program setting page |
| **5** | Click "Add" |  |
| **6** |  | Show a pop-up window for adding program type |
| **7** | Type in basic information of the program, click "Add" |  |
| **8** |  | Insert program information into program database |
| **9** |  | Return to the program setting page |

2.1.3.1.24 Process 24:

Table : Process Description: Set Document Type

|  |  |
| --- | --- |
| **Identifier** | UC-24: Set Document Type |
| **Purpose** | Administrator can set document type in the system |
| **Requirements** | WC\_3932 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator can set document type in the system 2. Administrator has logged into the system |
| **Post-conditions** | Document type will be made as the selection of specific function and added to the database |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Click “Document Settings” |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to a document setting page |
| **5** | Click "Add" |  |
| **6** |  | Show a pop-up window for adding document type |
| **7** | Type in basic information of the document, click "Add" |  |
| **8** |  | Insert document information into document database |
| **9** |  | Return to the document setting page |

2.1.3.1.23 Process 25:

Table : Process Description: Reset Any User’s Password

|  |  |
| --- | --- |
| **Identifier** | UC-25: Reset Any User’s Password |
| **Purpose** | Administrator can reset any user’s password in the system |
| **Requirements** | WC\_3936 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator can make reset any user’s password in the system 2. Administrator has logged into the system |
| **Post-conditions** | Corrasponding user’s password has been changed in the database |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Click “Reset Password” |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to a password resetting page |
| **5** | Type in email and password and click "Reset" |  |
| **6** |  | Validate the information from resetting password page |
| **7** |  | If validate failed, return error message |
| **8** |  | If validate succeed, insert new password into user database |
| **9** |  | Return to the password resetting page |

2.1.3.1.23 Process 26:

Table : Process Description: Set Survey Link

|  |  |
| --- | --- |
| **Identifier** | UC-26: Set Survey Link |
| **Purpose** | Administrator can set survey link in the system |
| **Requirements** | WC\_4010 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Only administrator can make settings in the system 2. Administrator has logged into the system |
| **Post-conditions** | Settings will be made as the selection of specific function and added to the database |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator]  Click “Survey Settings” |  |
| **2** |  | Verify the user if he/she is an administrator |
| **3** |  | If verify fail, return to fail page |
| **4** |  | If verify success, return to a survey setting page |
| **5** | Click "Add" |  |
| **6** |  | Show a pop-up window for adding survey link |
| **7** | Type in basic information of the survey, choose program and click "Add" |  |
| **8** |  | Insert survey information into survey database |
| **9** |  | Return to the survey setting page |

2.1.3.1.24 Process 27:

Table : Process Description: 10 Minutes Auto Logout

|  |  |
| --- | --- |
| **Identifier** | UC-27: 10 Minutes Auto Logout |
| **Purpose** | Administrator can set programs and document types by self |
| **Requirements** | WC\_3933; WC\_3976; WC\_3994; WC\_4003 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Administrator, case manager, staff or youth has logged into the system 2. User logged into the system does not make any operations in the system at least 10 minutes |
| **Post-conditions** | User has logged out and need to log into the system again |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager, staff or youth]  Have no operations at least 10 minutes |  |
| **2** |  | Session is destroyed and user has logged out of the system |

2.1.3.1.26 Process 28:

Table : Process Description: Reset Password for Self

|  |  |
| --- | --- |
| **Identifier** | UC-28: Reset Password for Self |
| **Purpose** | Any type of users can reset password by self |
| **Requirements** | WC\_4009 |
| **Development Risks** | None |
| **Pre-conditions** | 1. Administrator, case manager, staff or youth has not logged into the system |
| **Post-conditions** | User’s password has been changed in the database |

Table : Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | [Administrator, case manager, staff or youth]  Click “Login” on the home page |  |
| **2** |  | Return to login page |
| **3** | Click “Forgot your password?” |  |
| **4** |  | Return to request page |
| **5** | Type in email to receive link and click "request" |  |
| **6** |  | Receive email with reset link and button |
| **7** | Click reset link or button |  |
| **8** |  | Return to reset password page |
| **9** | Type in new password twice and click “Reset Password” |  |
| **10** |  | Validate the input from the reset password page |
| **11** |  | If validate failed, show error message |
| **12** |  | If validate succeed, update new password into user database |
| **13** |  | Return to home page |

##### Modes of Operation

The e-Lockbox system only operates in one mode, so there is no description about operation modes.

#### System Analysis Rationale

1. Based on how users will interact with the system, given the basic descriptions of four different type users of the e-Lockbox system as below:

(1) Administrator: This user has the highest level authority in the system. This user can manage all users accounts, documents, and cases in the system especially including delete, activate or inactivate documents, cases, and users etc.

(2) Case manager: This user can upload vital files for foster youth, view foster youth’s documents, edit case information of youth whom the manager is in charge of. Also, the case managers can activate or inactive cases.

(3) Staff: This user will only be able to view case and user information.

(4) Youth: This user will be the youth whom the Living Advantage is in charge of. User can view their own files stored in the system, and contact the case manager as needed.

2. According to the system context diagram and user case diagram, made following explanation:

Each user role and their functionalities is corresponding to use cases listed in 2.1.3.1, and the user cases listed in 2.1.3.1 is the whole thing of this system.

### Technology-Independent Model

This section was left out on purpose. Since our team has already known specific technology to use, design the system and software architecture, so we moved the content in this section to the next section and updated the design overview and design rationale.

### Technology-Specific System Design

#### Design Overview

##### System Structure

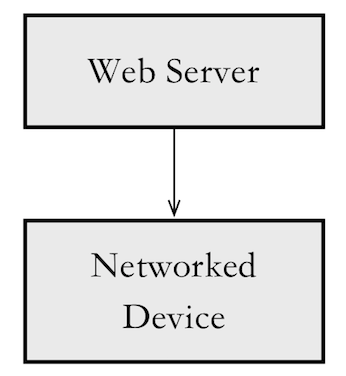


Figure : Hardware Component Class Diagram

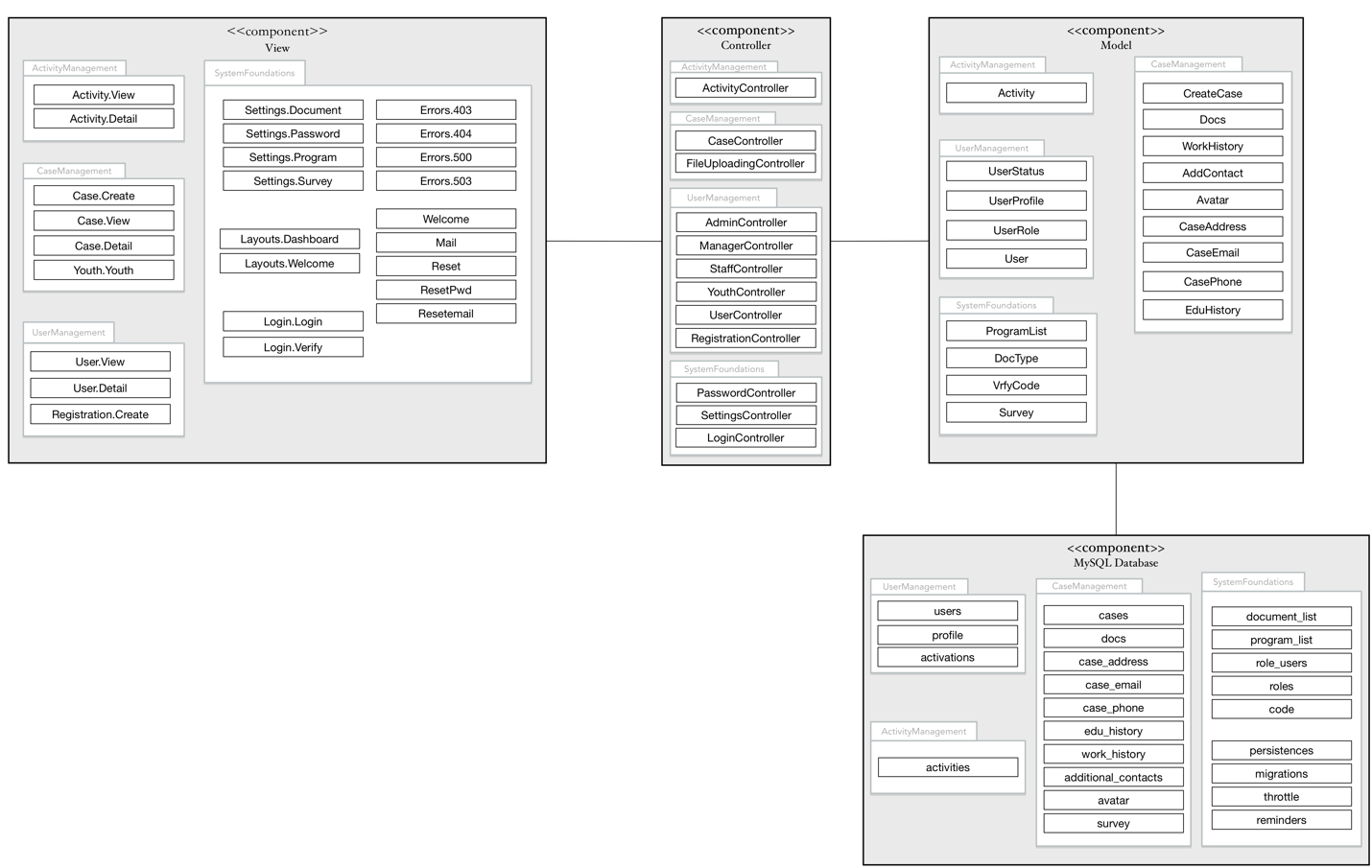


Figure : Software Component Class Diagram

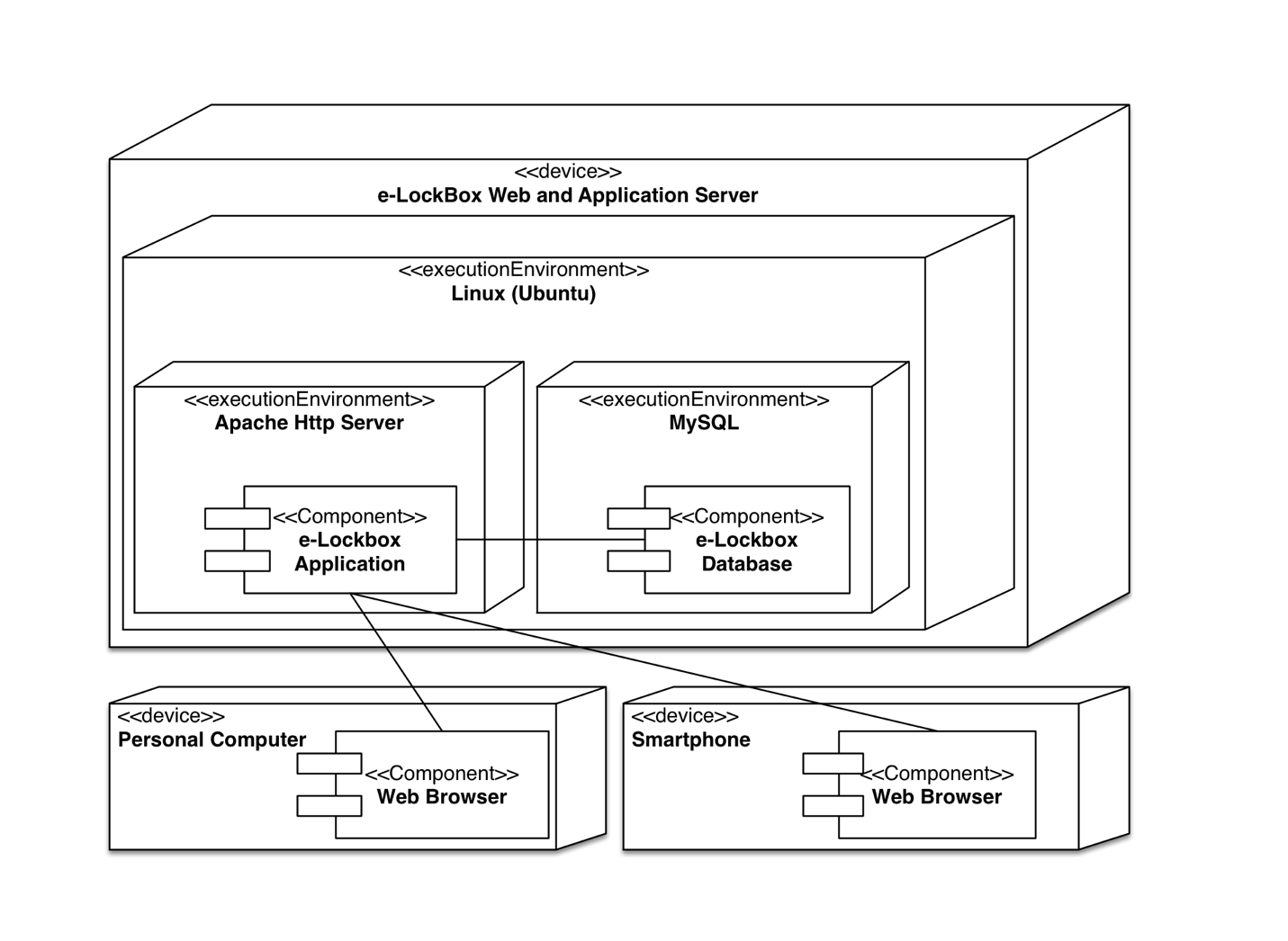


Figure : Deployment Diagram

Table : Hardware Component Description

|  |  |
| --- | --- |
| **Hardware Component** | **Description** |
| Networked Device | Networked device is a client-used device that can request data from server, and send data to server, like computer, mobile phone. |
| Web Server | Web server is the host of this system. It can handle the request from client devices and response data. |

Table 60: Software Component Description

|  |  |
| --- | --- |
| **Software Component** | **Description** |
| Activity.View | This is a webpage where admins, case managers and staff can view a table of activities, and create an activity. Admins can delete activities on this page. Not available for youth. |
| Activity.Detail | This is a webpage where admins, case managers and staff can view details of a chosen activity. And admins, case managers can also edit an activity on this page. Not available for youth. |
| Case.Create | This is a webpage where admins and case managers can create a case. Not available for staff and youth. |
| Case.View | This is a webpage where admins, case managers and staff can view a table of cases. Not available for youth. |
| Case.Detail | This is a webpage where admins, case managers, staff can view details of a chosen case. Admin and case managers can edit all case-related information. Admin can delete cases on this page. |
| Youth.Youth | This is a webpage where youths can view his/her case information. Not available for admins, case managers or staff. |
| User.View | This is a webpage where admins, case managers and staff can view a table of users. Not available for youth. |
| User.Detail | This is a webpage where admins, case managers and staff can view details of a chosen user. Admins can also edit users’ information. Case managers and staff can only edit one’s information. Not available for youth. |
| Registration.Create | This is a webpage where admins can create a user. Not available for case managers, staff and youth. |
| Settings.Document | This is a webpage where admins can view, edit, create and delete a document type. Not available for case managers, staff and youth. |
| Settings.Program | This is a webpage where admins can view, edit, create and delete a program type. Not available for case managers, staff and youth. |
| Settings.Password | This is a webpage where admins can reset others’ password. Not available for case managers, staff and youth. |
| Settings.Survey | This is a webpage where admins can create, edit a survey to a certain group of youths. Not available for case managers, staff and youth. |
| Layouts.Dashboard | This is a template for all pages which are under Activity, Case, User and Settings. |
| Layouts.Welcome | This is a template for all pages which are under Login. |
| Login.Login | This is a webpage where users can login. Available for all users. |
| Login.Verify | This is a webpage where users can verify themselves by submitting verification code. Available for all users. |
| Errors.403 | This is an error page for 403 Error. Any 403 error will cause this page. |
| Errors.404 | This is an error page for 404 Error. Any 404 error will cause this page. |
| Errors.500 | This is an error page for 500 Error. Any 500 error will cause this page. |
| Errors.503 | This is an error page for 503 Error. Any 503 error will cause this page. |
| Welcome | This is the home webpage. |
| Mail | This is a template for verification email. |
| Resetemail | This is a template for reset password email. |
| Reset | This is a webpage where users can request to change their password. |
| ResetPwd | This is a webpage where users can submit their new password. |
| ActivityController | This is a controller which handles all requests about activity management, like create, edit, delete, view, detail, etc. |
| CaseController | This is a controller which handles all requests about case management except uploading a file. |
| FileUploadingController | This is a controller which handles uploading file request. |
| AdminController | This is a controller which controls the admins’ pages. |
| ManagerController | This is a controller which controls the managers’ pages. |
| StaffController | This is a controller which controls the staff’s pages. |
| YouthController | This is a controller which controls the youths’ pages. |
| UserController | This is a controller which handles all requests about user management, like edit, inactivate, activate. |
| RegistrationController | This is a controller which handles creating new accounts requests. |
| PasswordController | This is a controller which handles all requests about password, like reset password. |
| SettingsController | This is a controller which handles all requests about settings, like document type, program list, etc. |
| LoginController | This is a controller which handles all requests about login, like login, verification, etc. |
| Activity | This is a model which connects to “activities” table in database. It saves all information about activities. |
| UserStatus | This is a model which connects to “activations” table in database. It saves user status. |
| UserProfile | This is a model which connects to “profile” table in database. It saves user profile, like phone number, etc. |
| UserRole | This is a model which connects to “role\_users” table in database. It saves every user’s role. |
| User | This is a model which connects to “users” table in database. It saves users’ email, password, name, etc. |
| ProgramList | This is a model which connects to “program\_list” table in database. It saves all the program types. |
| DocType | This is a model which connects to “document\_list” table in database. It saves all the document types. |
| VrfyCode | This is a model which connects to “code” table in database. It saves verification code for every user. |
| Survey | This is a model which connects to “survey” table in database. It saves survey information, like description, link, etc. |
| CreateCase | This is a model which connects to “cases” table in database. It saves case information, like name, DOB, etc. |
| Docs | This is a model which connects to “docs” table in database. It saves case documents’ information, like path, name, etc. |
| WorkHistory | This is a model which connects to “work\_history” table in database. It saves case related work history information, like start date, end date, etc. |
| AddContact | This is a model which connects to “additional\_contacts” table in database. It saves case related contacts, like phone number, name, etc. |
| Avatar | This is a model which connects to “avatar” table in database. It saves case avatars’ information, like path, name, etc. |
| CaseAddress | This is a model which connects to “case\_address” table in database. It saves case related address information, like address, status, etc. |
| CaseEmail | This is a model which connects to “case\_email” table in database. It saves case related address information, like email, status, etc. |
| CasePhone | This is a model which connects to “case\_phone” table in database. It saves case related address information, like phone number, status, etc. |
| EduHistory | This is a model which connects to “edu\_history” table in database. It saves case related education history information, like start date, end date, etc. |

##### Design Classes

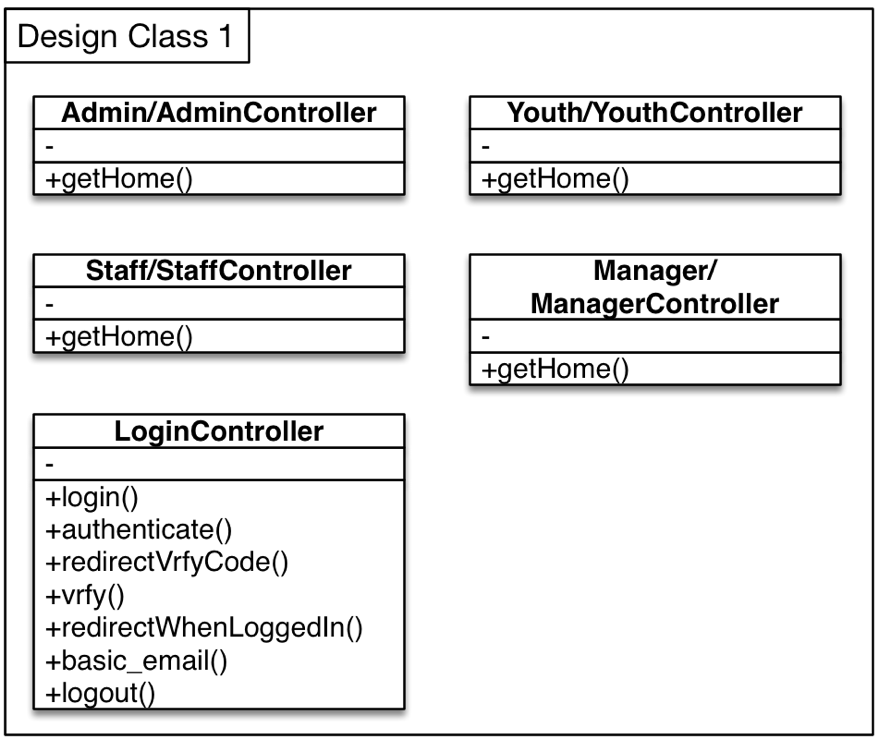


Figure : Design Class Diagram 1

Table 61: Design Class Description 1

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Admin/AdminController | Controller | Admin Controller defines admin’s homepage. |
| Manager/ManagerController | Controller | Manager Controller defines manager’s homepage. |
| Staff/StaffController | Controller | Staff Controller defines staff’s homepage. |
| Youth/YouthController | Controller | Youth Controller defines youth’s homepage. |
| LoginController | Controller | Login controller can control user login with verification code, and log out function. |

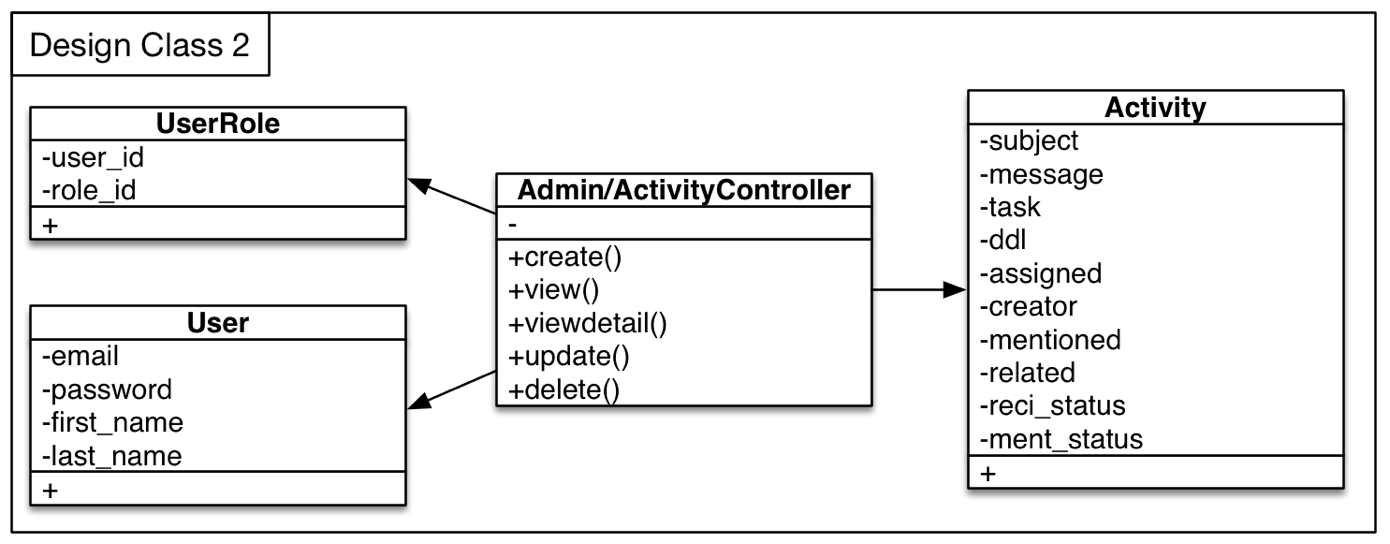


Figure : Design Class Diagram 2

Table 62: Design Class Description 2

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Admin/ActivityController | Controller | This controller defines some methods related to activity, including create, delete, update and view. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| Activity | Model | This model connects controllers and activities table. |

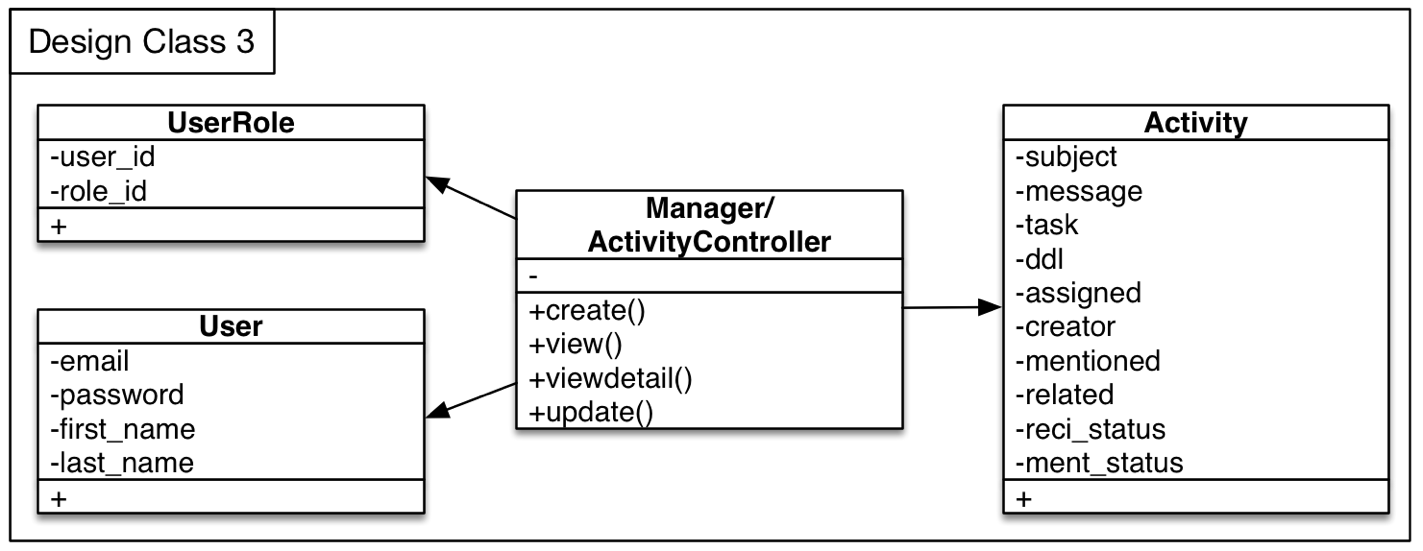


Figure : Design Class Diagram 3

Table 63: Design Class Description 3

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Manager/ActivityController | Controller | This controller defines some methods related to activity, including create, update and view. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| Activity | Model | This model connects controllers and activities table. |

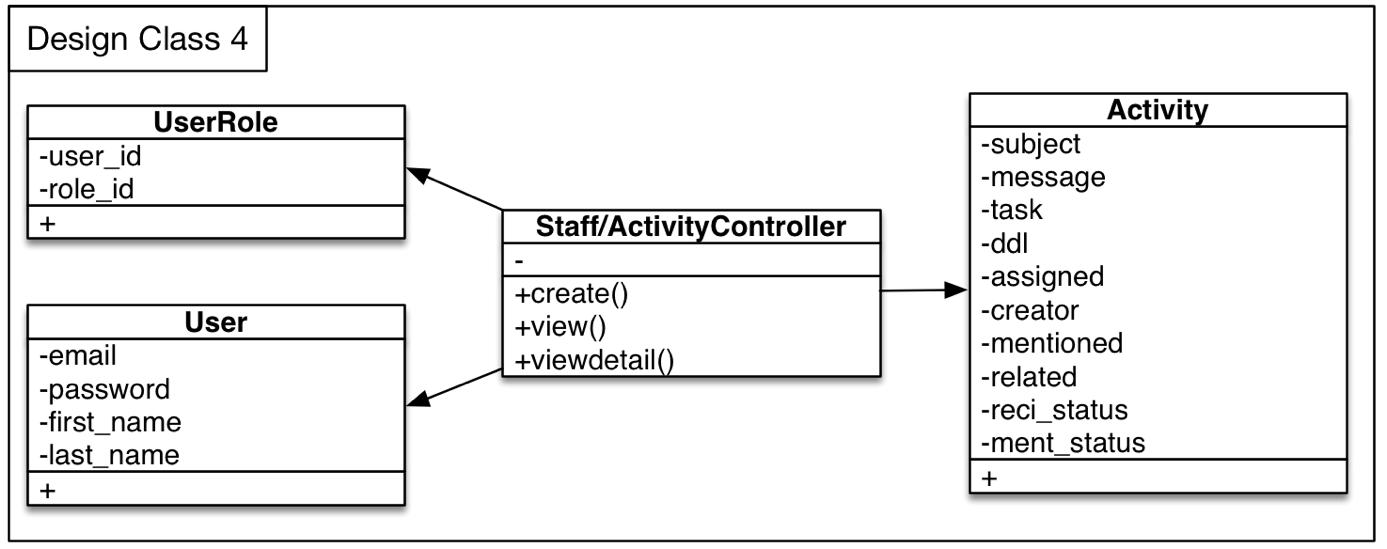


Figure : Design Class Diagram 4

Table 64: Design Class Description 4

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Staff/ActivityController | Controller | This controller defines some methods related to activity, including create and view. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| Activity | Model | This model connects controllers and activities table. |

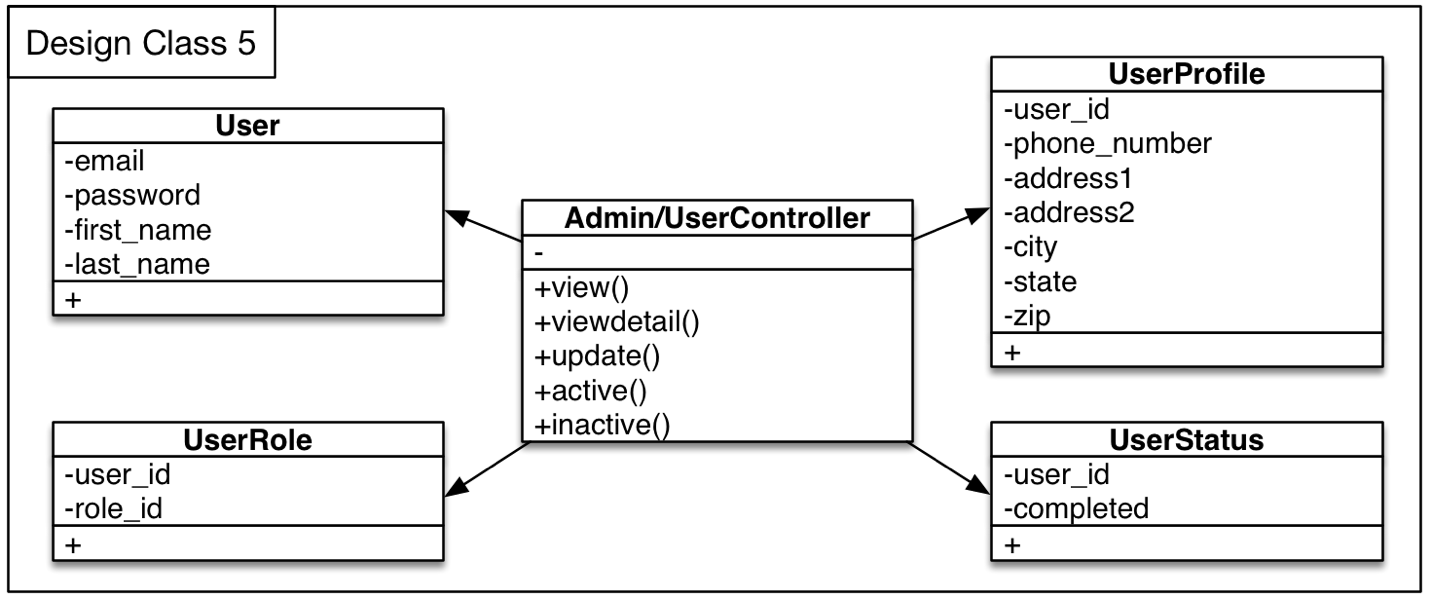


Figure : Design Class Diagram 5

Table 65: Design Class Description 5

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Admin/UserController | Controller | This controller defines some methods related to user, including create, delete, update, view, activate and inactivate. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| UserProfile | Model | This model connects controllers and activities table. |
| UserStatus | Model | This model connects controllers and activations table. |

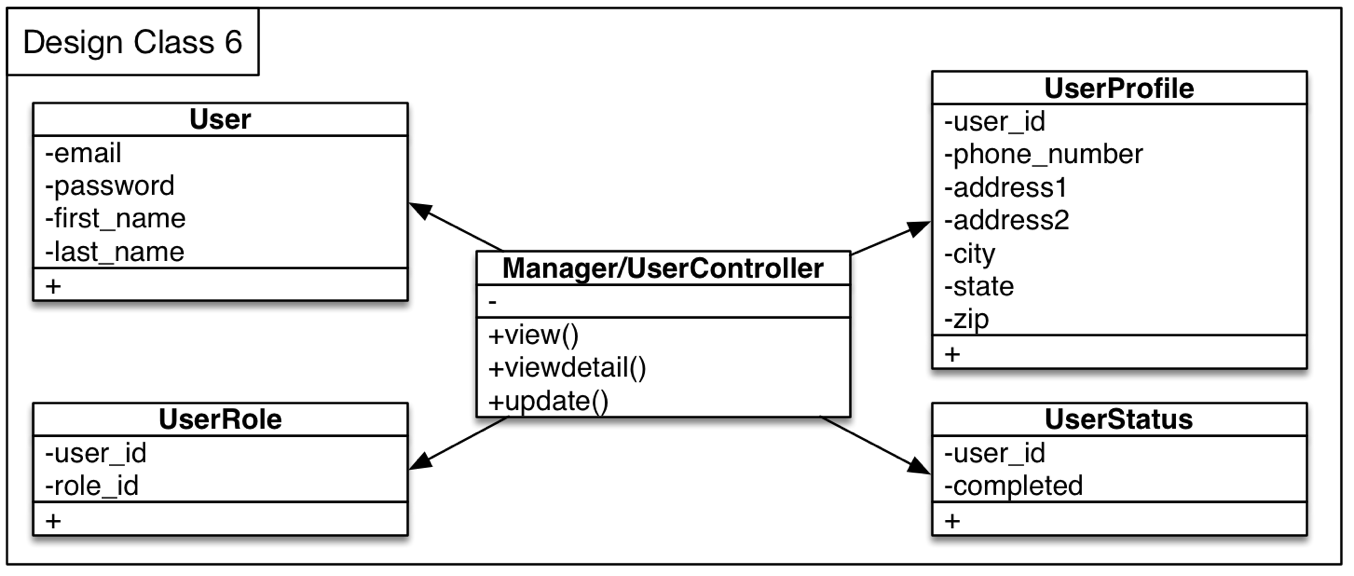


Figure : Design Class Diagram 6

Table 66: Design Class Description 6

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Manager/UserController | Controller | This controller defines some methods related to user, including update and view. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| UserProfile | Model | This model connects controllers and activities table. |
| UserStatus | Model | This model connects controllers and activations table. |

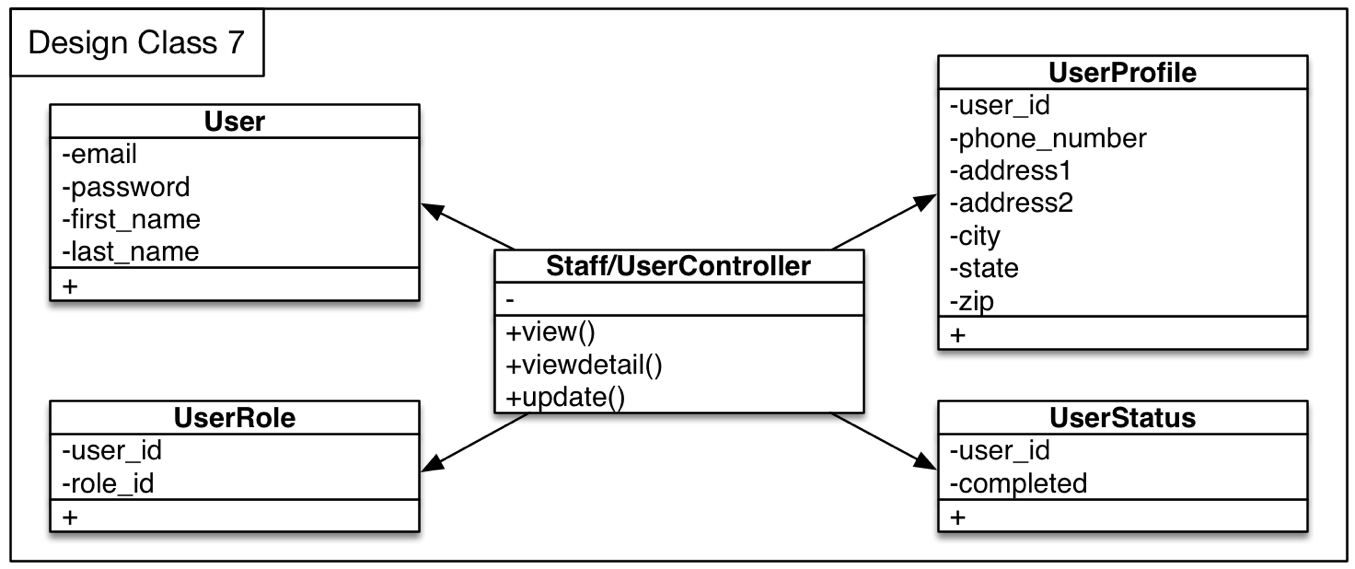


Figure : Design Class Diagram 7

Table 67: Design Class Description 7

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Staff/UserController | Controller | This controller defines some methods related to user, including update and view. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| UserProfile | Model | This model connects controllers and activities table. |
| UserStatus | Model | This model connects controllers and activations table. |

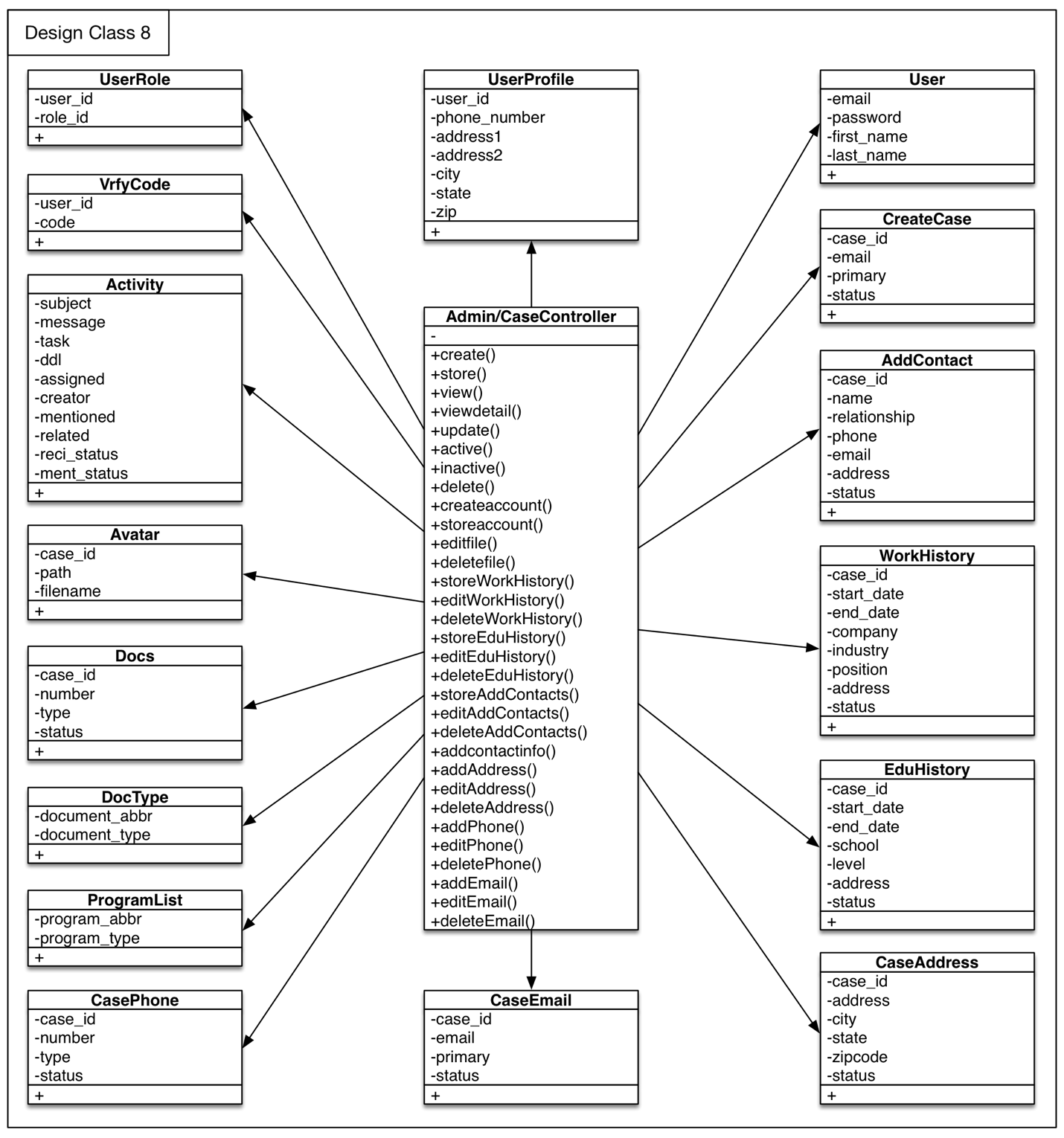


Figure : Design Class Diagram 8

Table 68: Design Class Description 8

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Admin/CaseController | Controller | This controller defines some methods related to case, including create, delete, update, view, activate, inactivate and some functions to case related information. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| UserProfile | Model | This model connects controllers and activities table. |
| VrfyCode | Model | This model connects controllers and code table. |
| Activity | Model | This model connects controllers and activities table. |
| Avatar | Model | This model connects controllers and avatar table. |
| Docs | Model | This model connects controllers and docs table. |
| DocType | Model | This model connects controllers and document\_list table. |
| ProgramList | Model | This model connects controllers and program\_list table. |
| CasePhone | Model | This model connects controllers and case\_phone table. |
| CaseEmail | Model | This model connects controllers and case\_email table. |
| CaseAddress | Model | This model connects controllers and case\_address table. |
| EduHistory | Model | This model connects controllers and edu\_history table. |
| WorkHistory | Model | This model connects controllers and work\_history table. |
| AddContact | Model | This model connects controllers and additional\_contacts table. |
| CreateCase | Model | This model connects controllers and cases table. |

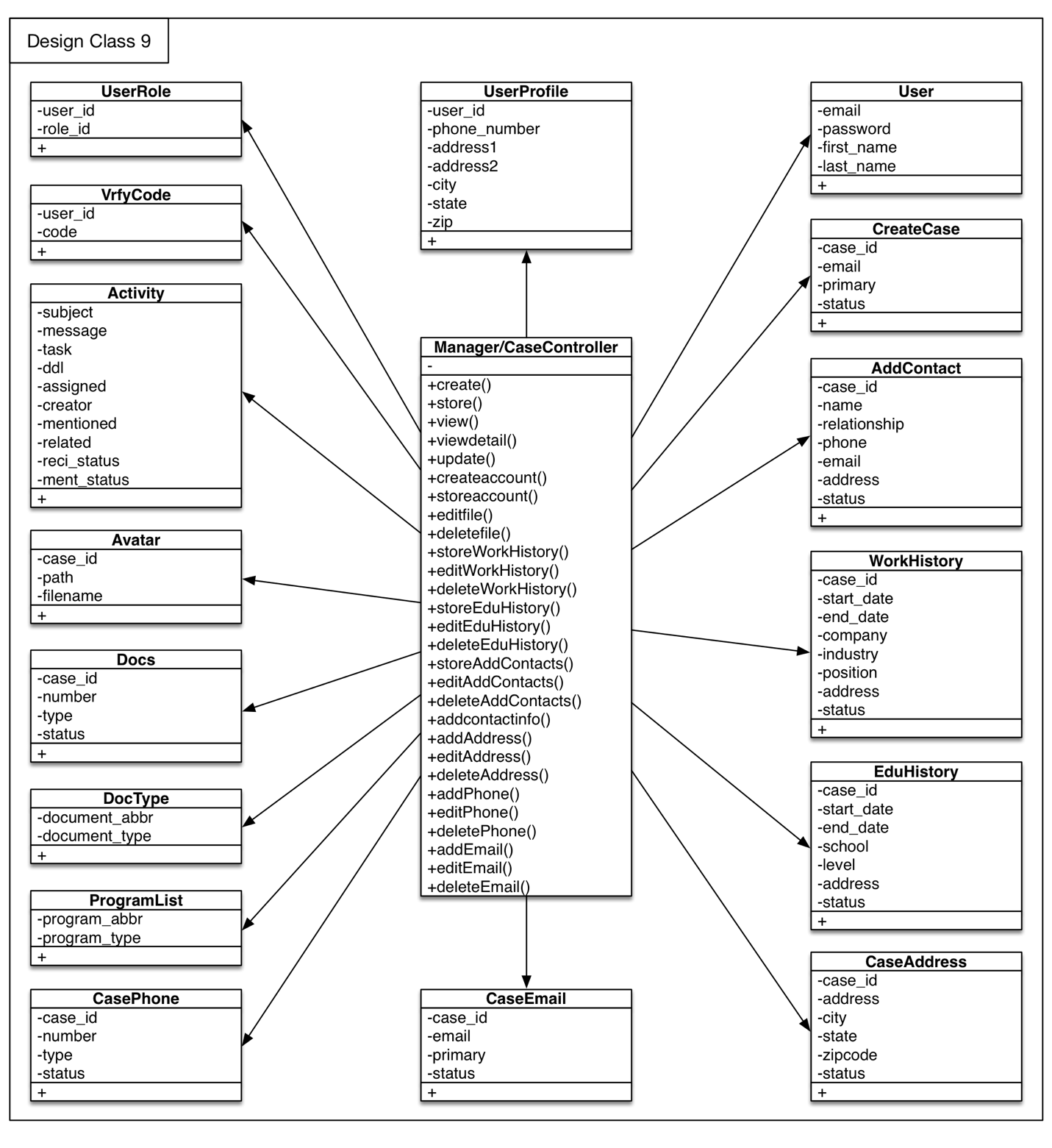


Figure : Design Class Diagram 9

Table 69: Design Class Description 9

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Manager/CaseController | Controller | This controller defines some methods related to case, including create, update, view and some functions to case related information. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| UserProfile | Model | This model connects controllers and activities table. |
| VrfyCode | Model | This model connects controllers and code table. |
| Activity | Model | This model connects controllers and activities table. |
| Avatar | Model | This model connects controllers and avatar table. |
| Docs | Model | This model connects controllers and docs table. |
| DocType | Model | This model connects controllers and document\_list table. |
| ProgramList | Model | This model connects controllers and program\_list table. |
| CasePhone | Model | This model connects controllers and case\_phone table. |
| CaseEmail | Model | This model connects controllers and case\_email table. |
| CaseAddress | Model | This model connects controllers and case\_address table. |
| EduHistory | Model | This model connects controllers and edu\_history table. |
| WorkHistory | Model | This model connects controllers and work\_history table. |
| AddContact | Model | This model connects controllers and additional\_contacts table. |
| CreateCase | Model | This model connects controllers and cases table. |

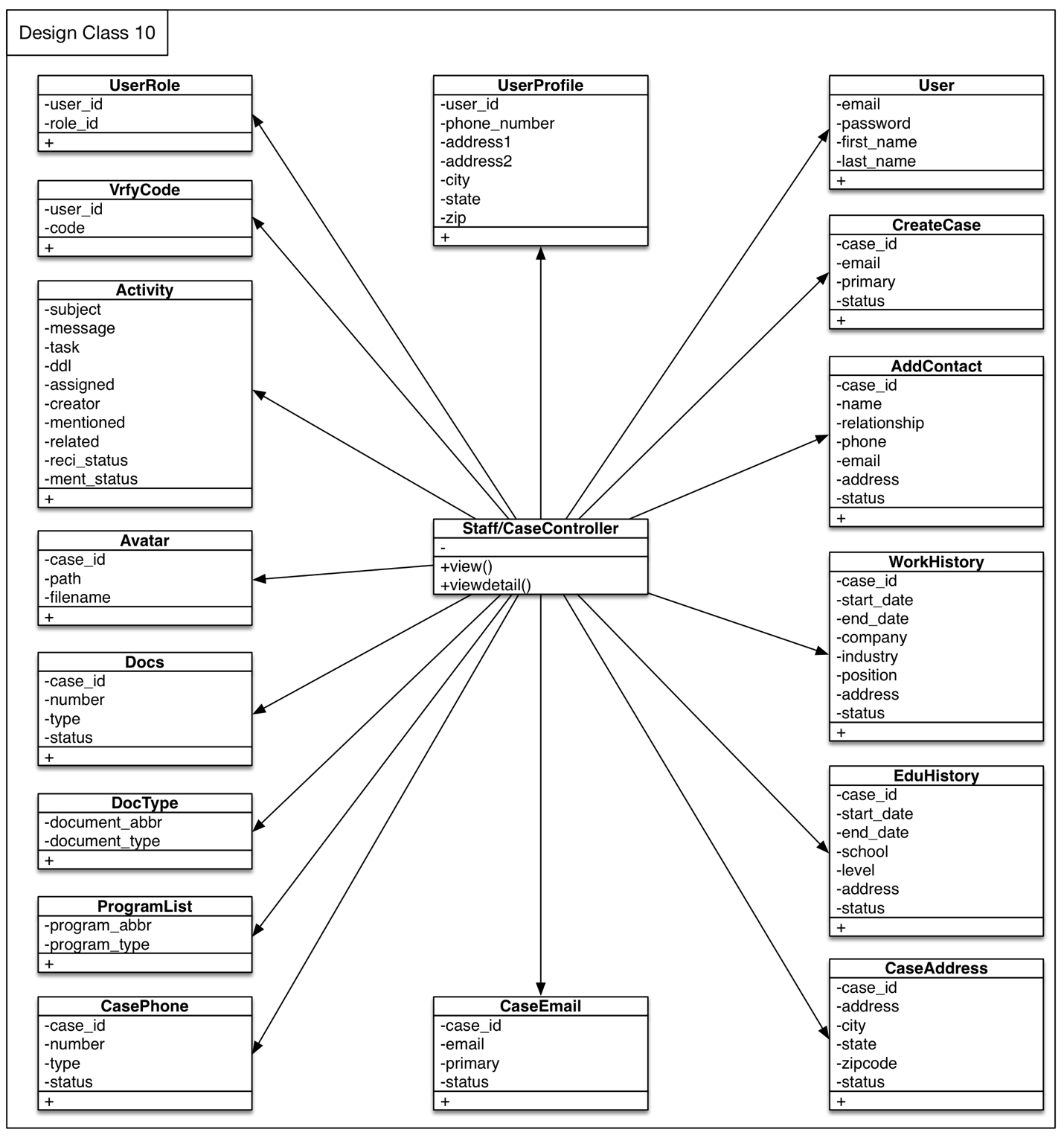


Figure : Design Class Diagram 10

Table 70: Design Class Description 10

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Staff/CaseController | Controller | This controller defines some methods related to case view. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| UserProfile | Model | This model connects controllers and activities table. |
| VrfyCode | Model | This model connects controllers and code table. |
| Activity | Model | This model connects controllers and activities table. |
| Avatar | Model | This model connects controllers and avatar table. |
| Docs | Model | This model connects controllers and docs table. |
| DocType | Model | This model connects controllers and document\_list table. |
| ProgramList | Model | This model connects controllers and program\_list table. |
| CasePhone | Model | This model connects controllers and case\_phone table. |
| CaseEmail | Model | This model connects controllers and case\_email table. |
| CaseAddress | Model | This model connects controllers and case\_address table. |
| EduHistory | Model | This model connects controllers and edu\_history table. |
| WorkHistory | Model | This model connects controllers and work\_history table. |
| AddContact | Model | This model connects controllers and additional\_contacts table. |
| CreateCase | Model | This model connects controllers and cases table. |

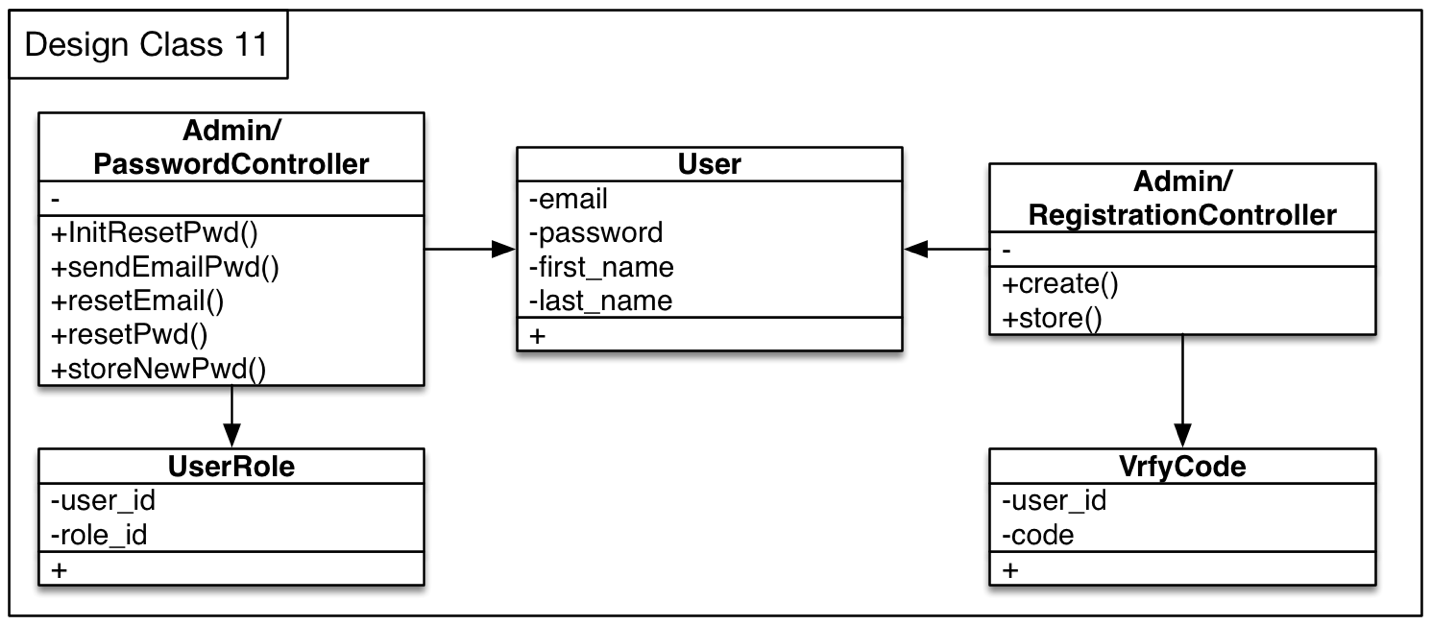


Figure : Design Class Diagram 11

Table 71: Design Class Description 11

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Admin/PasswordController | Controller | This controller defines some methods related to reset password. |
| Admin/RegistrationController | Controller | This controller defines some methods related to create user. |
| User | Model | This model connects controllers and users table. |
| UserRole | Model | This model connects controllers and role\_users table. |
| VrfyCode | Model | This model connects controllers and code table. |

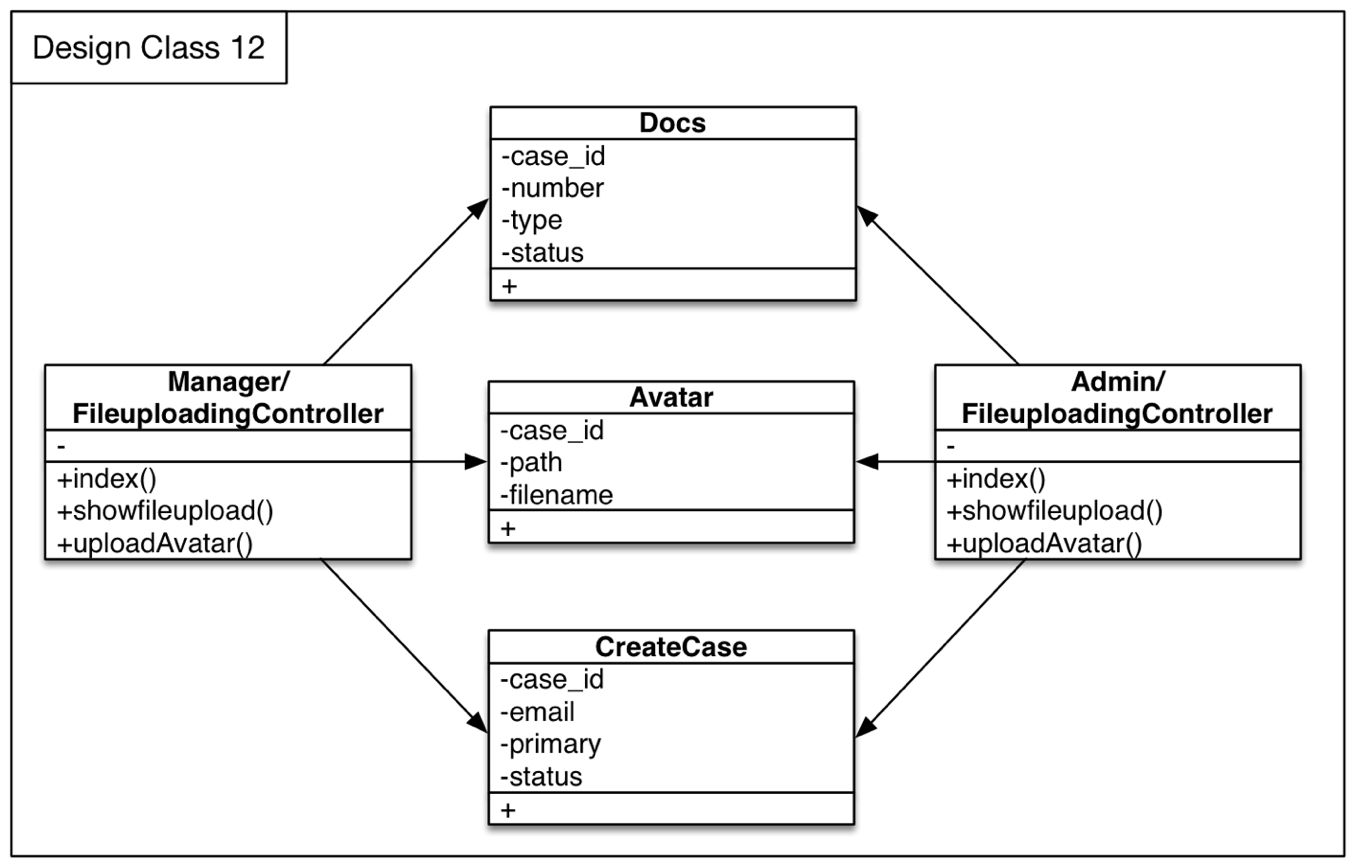


Figure : Design Class Diagram 12

Table 72: Design Class Description 12

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Manager/FileuploadingController | Controller | This controller defines some methods related to file uploading. |
| Admin/FileuploadingController | Controller | This controller defines some methods related to file uploading. |
| Docs | Model | This model connects controllers and docs table. |
| Avatar | Model | This model connects controllers and avatar table. |
| CreateCase | Model | This model connects controllers and cases table. |

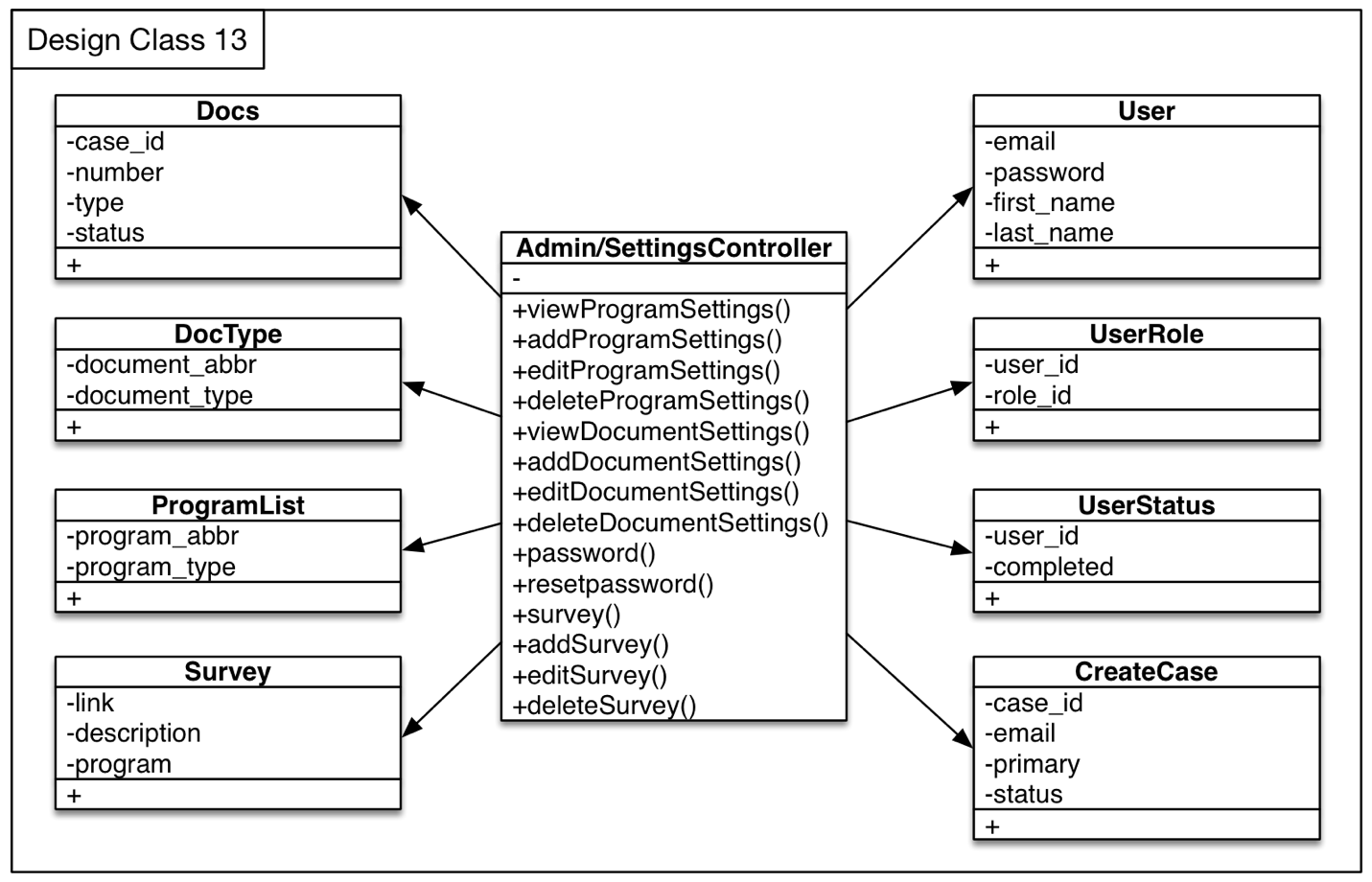


Figure : Design Class Diagram 13

Table 73: Design Class Description 13

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Admin/SettingsController | Controller | This controller defines some methods related to settings including manipulate program types, document types, password, etc. |
| DocsType | Controller | This model connects controllers and document\_list table. |
| Docs | Model | This model connects controllers and docs table. |
| ProgramList | Model | This model connects controllers and program\_list table. |
| Survey | Model | This model connects controllers and survey table. |
| UserRole | Model | This model connects controllers and role\_users table. |
| User | Model | This model connects controllers and users table. |
| UserStatus | Model | This model connects controllers and activations table. |
| CreateCase | Model | This model connects controllers and cases table. |

##### Process Realization

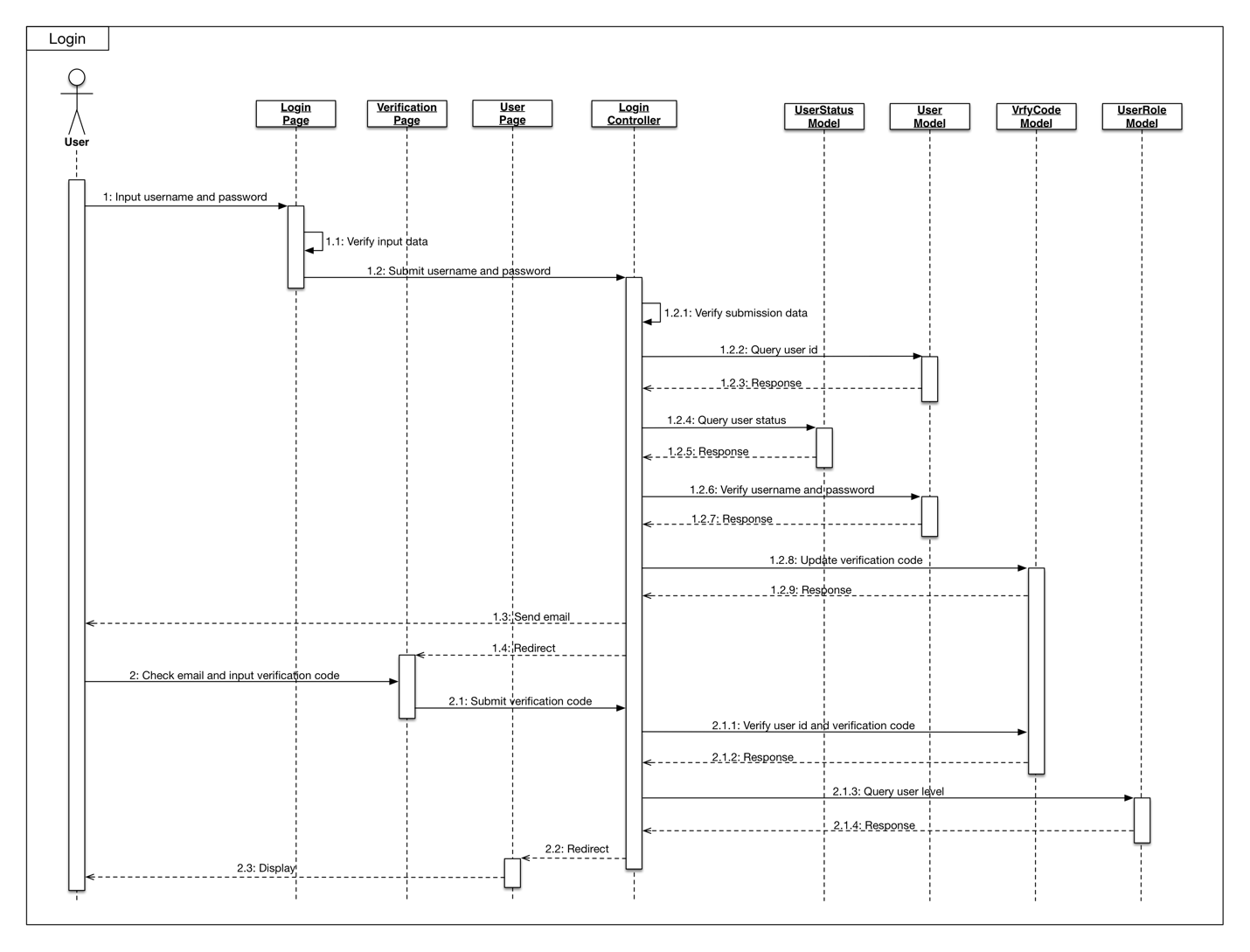
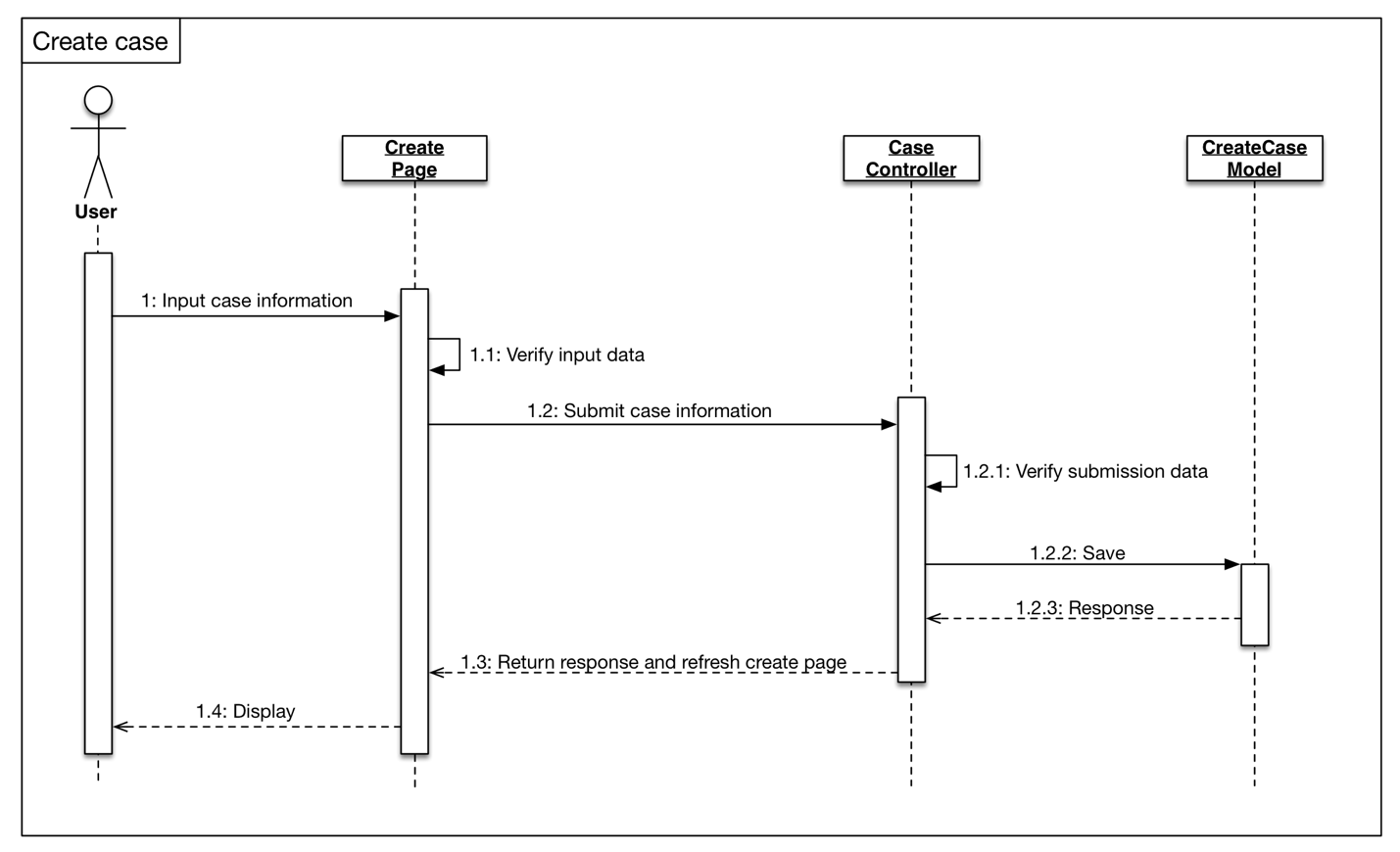
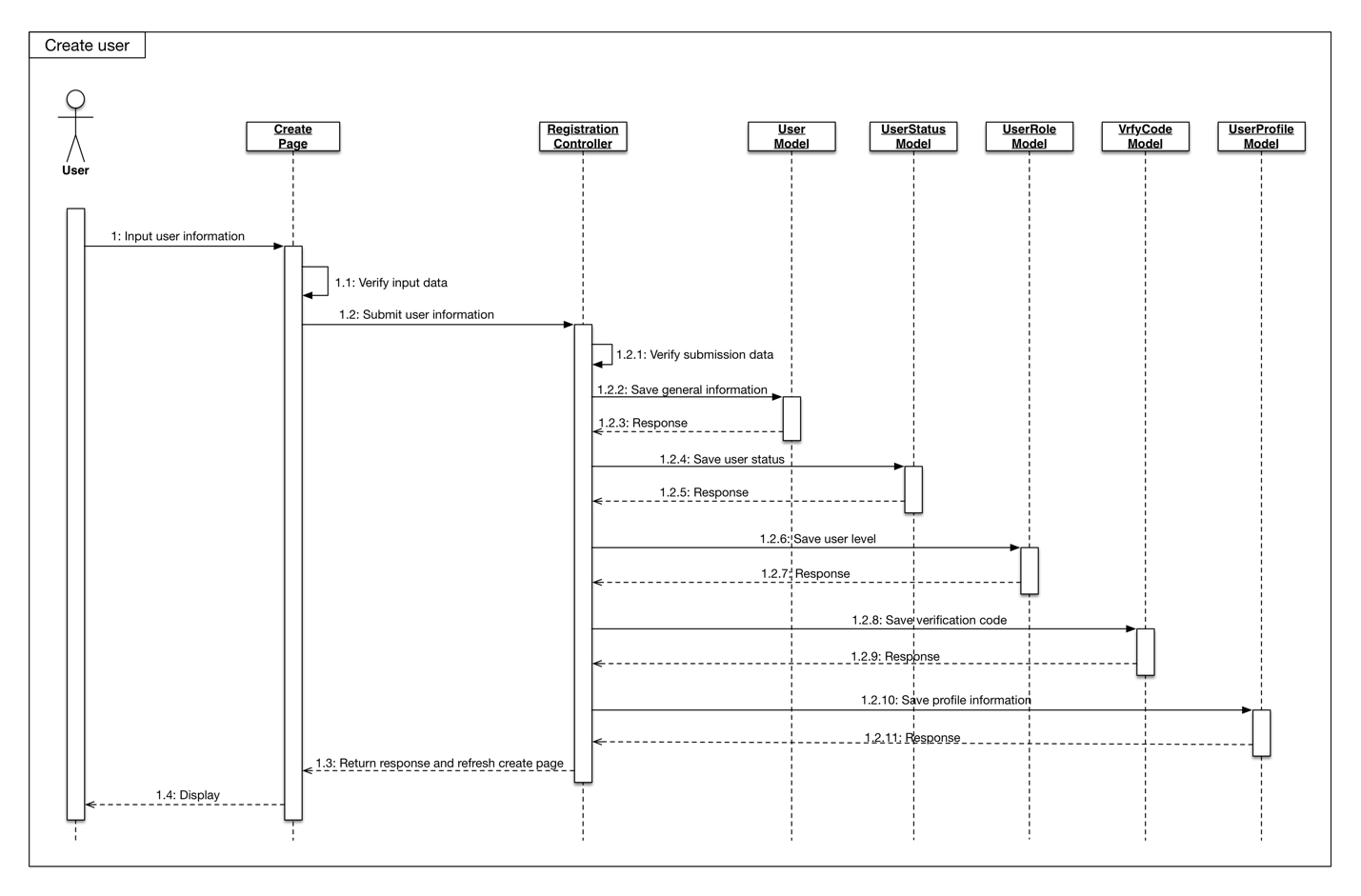


Figure : Sequence Diagram – Login



**Figure 25: Sequence Diagram - Create Case**



**Figure 26: Sequence Diagram – Create High Level User**

#### Design Rationale

We designed the system according to different levels of access to the system of different kind of user. Each user will have its information, while youth user is the reason we building this system, so they will have their important information and document stored safely 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 DigitalOcean and the database we have are mainly contain the index of each kind of data.

### Architectural Styles, Patterns and Frameworks

In the table below, you can see our architectural styles, patterns and frameworks that we used so far.

Table : Architectural Styles, Patterns, and Frameworks

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Benefits, Costs, and Limitations** |
| MVC | Model–View–Controller (MVC) is a software architecture pattern, which separates the representation of information from the user's interaction. This architecture includes models, views, and controllers, followings are the description of each component:  **Controller:** 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.  **Model:** 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.  **View:** View requests information from the model that it needs for generating an output representation to the user. | **Benefits:**  1. Separation of concerns:  The separation of three components, allows the re-use of the business logic across applications. Multiple User Interfaces can be developed without concerning the codebase.  2. Developer specialization and focus:  The UI for developers helps focus exclusively on the UI screens without bogged down with business logic.  **Costs:**  There is no specific cost required to use MVC architecture.  **Limitations:**  1.Because of the decoupling of MVC, it may increase complexity of the system.  2.MVC needs multiple programmers and knowledge of multiple programming.  3.Inefficient data access in view. |