Kastle 2.0

**(KIPP)**

**CS4962 Senior Design**

**Software Requirements Document**

Prepared by

Gonzalez, Jose

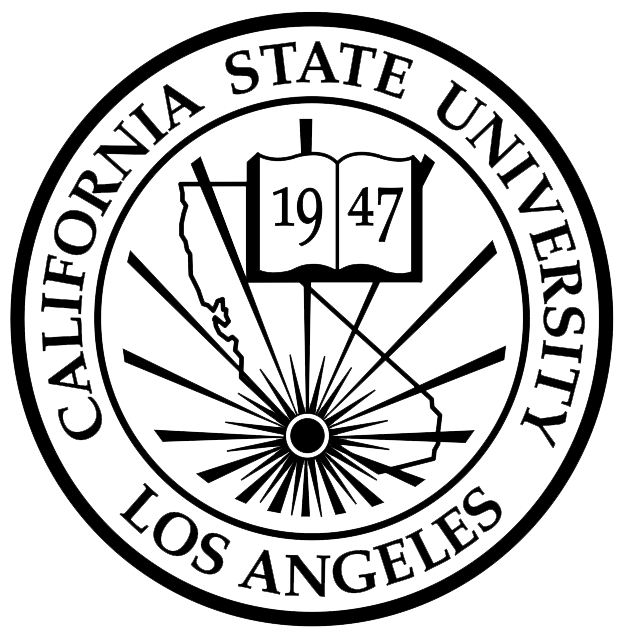
Mahbub, Safa

Perez, Juan

12/1/2017

CALIFORNIA STATE UNIVERSITY

LOS ANGELES



Los Angeles, California

**Kastle 2.0**

**(KIPP)**

**Prepared By:**

Gonzalez, Jose

Mahbub, Safa

Perez, Juan

**Approved by:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Russell Abbott Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Jungsoo (Sue) Lim Date

**Document Change Log**

|  |  |  |
| --- | --- | --- |
| **Update** | **Date Released** | **Changes** |
| 0.1.0 | 12/1/2017 | Delivery of the Software Requirements document |
| 0.1.1 | 02/19/2018 | Delivery of a wireframe for the web application from Knowledge Is Power Program Los Angeles (KIPPLA). |
|  |  |  |

# List of TBD Items

|  |  |  |  |
| --- | --- | --- | --- |
| Page | **Item** | **Description** | **Status** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**1.0 INTRODUCTION**

1.1 Purpose 3-3

1.2 Scope 3-3

1.2.1 Document Organization 3-3

1.2.2 Relationship to Other Documents 3-3

1.3 **Kastle 2.0** System Architecture 3-3

1.3.1 Context Diagram (Level 0 DFD) 3-4

1.3.2 Description of **Kastle 2.0** Major Units 4-4

1.4 Requirement Development Process 4-4

1.5 References 5-5

1.5.1 Controlling Documents 5-5

1.5.2 Applicable Documents 5-5

1.5.3 Standards 5-5

**2.0 DETAILED FUNCTIONAL DESCRIPTION OF THE *KASTLE 2.0***

2.1 Detailed Functional Description 6-6

2.1.1 Higher Level DFDs 6-6

2.1.2 Detailed Functional Description of Major Sub-Units 6-8

**3.0 KASTLE 2.0 NAME REQUIREMENTS**

3.1 Kastle 2.0 FUNCTIONAL REQUIREMENTS 9-10

3.1.1

3.2 Kastle 2.0 NON-FUNCTIONAL REQUIREMENTS 10-10

3.2.1

3.2 Kastle 2.0 HARDWARE REQUIREMENTS 10-10

**4.0 Kastle 2.0** **DETAILED DESIGN** 11-12

**5.0 KASTLE 2.0 ELEMENTS OF IMPLEMENTATION** 13-13

**A. ACRONYMS** A-1

**B. DATA DICTIONARY** B-1

**List of Figures**

Figure 1-1 Context Diagram (Level 0 Data Flow Diagram)

Figure 2-1 Level 1 Data Flow Diagram

Figure 4.1.1 Level 2 Data Flow Diagram-Main Module

Figure 4.1.2 Level 2 Data Flow Diagram-Display Module

Figure 4.1.3 Level 2 Data Flow Diagram-Authentication Module

Figure 4.1.4 Level 2 Data Flow Diagram-User Module

Figure 4.1.5 Level 2 Data Flow Diagram-Content manager Module

**List of Tables**

None

**1.0 INTRODUCTION**

**1.1 Purpose**

The purpose of this document is **four-fold**:

a) Completely define a full set of requirements for the **Kastle 2.0 – Section 3.0**.

b) Completely define the design for the **Kastle 2.0 – Section 4.0**.

c) Define and partially implement feasible modules for the **Kastle 2.0 – Section 5.0**.

d) Completely define the Test Plan for the **Kastle 2.0 – Section 6.0**.

The complete definition of all **Kastle 2.0** requirements provides the source requirement inputs for the development of the subsequent supporting software subsystems documents.

**1.2 Scope**

The documentation developed as part of this Senior Design CS4961 class, starts with the SRD including elements of Software Design and parts of a Test Plan.

The scope of this document includes the following:

* All functional and non-functional requirements on the **Kastle 2.0** are captured. This includes Verification & Validation (V&V) requirements, as well as inter-software subsystems requirements.g
* A complete set of **Kastle 2.0** Requirements. These requirements are organized by key **Kastle 2.0** functional units shown on the Level 1 DFD. The Level 1 DFD is shown on page [TBD].
* The functional requirements defined in the **Kastle 2.0** Requirements section have been expanded to include more specific hardware requirements.

**1.2.1 Document Organization**

The organization of this document provides a natural flow or allocation of requirements to each succeeding section.

Details regarding the overall document are given in sub-section 1.5 below.

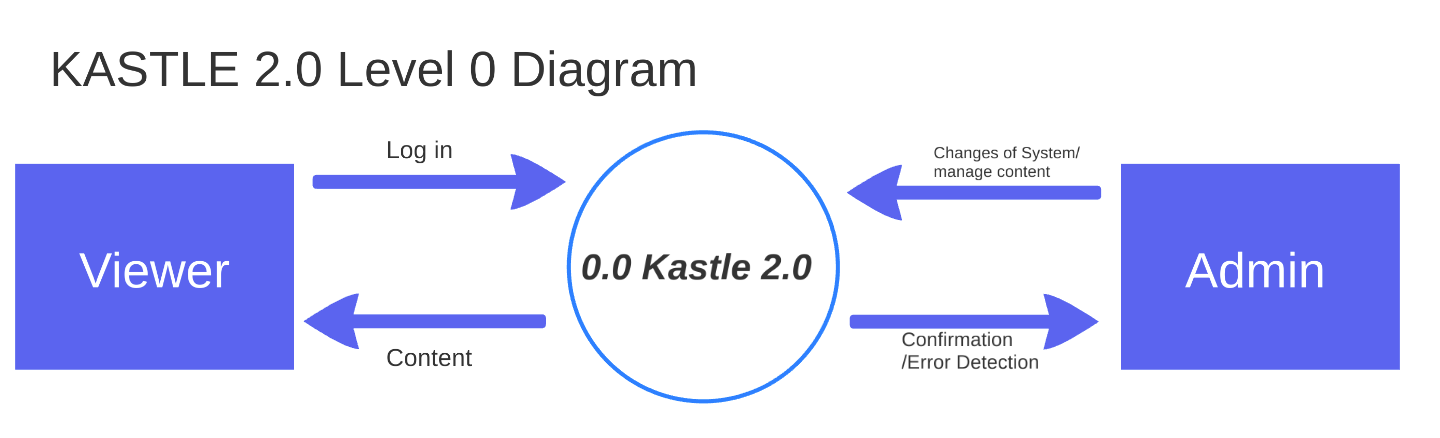
**1.2.2 Relationship to Other Documents**

The **Kastle 2.0** SRD is a complete self contained document. Some relationships to other documents in the literature are indicated below in sub-section 1.5.

**1.3 Kastle 2.0 Architecture**

1.3.1 Detailed Context Diagram (DFD Level 0)

The **Project Name** architecture is summarized in the Context Diagram (DFD Level 0) given below. A more complete Functional Description is given in Section 2 of this document. The Context Diagram provides the overall structure of the software modules and all its inputs and outputs. The notation used corresponds to that defined for any Data Flow Diagram (DFD).



**1.3.2** Description and major functions of the **Kastle 2.0**

The project referred as Kastle 2.0 was brought to us by Knowledge is Power Program of Los Angeles also known as KIPP LA. Kastle 2.0 is a site where the KIPP LA organization will be able to maintain all their data (in the form of a Tableau dashboard) in one website rather than maintaining multiple websites for the different level of users. Added functionality to the site will include comment dashboards, bookmarking dashboards users would like to refer to in the future, and a recent dashboard so users can see the newest data available.

Successful use of the site will reduce the workload an admin will have to do in order to add new dashboards, manage user permissions, improve navigation system, implement a filtered search for dashboards, and utilize a single sign in to access all necessary data (i.e from the website and from the Tableau Private server). It shall also reduce the strain on a user to sign into multiple accounts in order to see the dashboards.

**1.4 Documentation Development Process**

The **Kastle 2.0** detailed functional description is documented in section 2.0. Basically, Section 2 is a succinct software description document. The overall detailed functional description is based on higher level DFDs (above level 1). All major functional units are described in detail in this part of the document.

In general, all requirements affecting **Kastle 2.0** are captured in Section 3.0 of this document. These requirements are a refinement and completion of requirements first collected as part of this Software Engineering project. The document is cited in Section 1.2.2. This section is the one worked in most detail to become a reasonably complete Software Requirements Document (SRD). It includes both functional and non-functional software requirements together with several detailed “rational” paragraphs whenever necessary to complete the understanding of each requirement.

Section 4 is the detailed **Kastle 2.0** Software Design Description Document (SDD). This part of the document includes all higher level DFDs as described in section 2 plus all interface units. The document is highly technical and it is based on section 2 descriptions. An important component is the addition of a SIS (software interface specification) document in sub-section 4.2.

Section 5 includes elements of a partial implementation of **Kastle 2.0**. This section includes the various constraints that effectively limit the implementation as well as the sub-units that will be coded. The implementation goals are defined and the code and pseudo code are included as an attachment to this section.

**1.5 References**

All references used in the creation of this document are listed below.

**1.5.1 Controlling Documents**

1) There is no document controlling this document.

**1.5.2 Applicable Documents**

1) No additional applicable document has been used in the production of this document.

**1.5.3 Standards**

No Standard has been used in the creation of this document. However, some Standards described in textbooks have been examined as a reference. In particular, the IEEE standard has been briefly discussed in class.

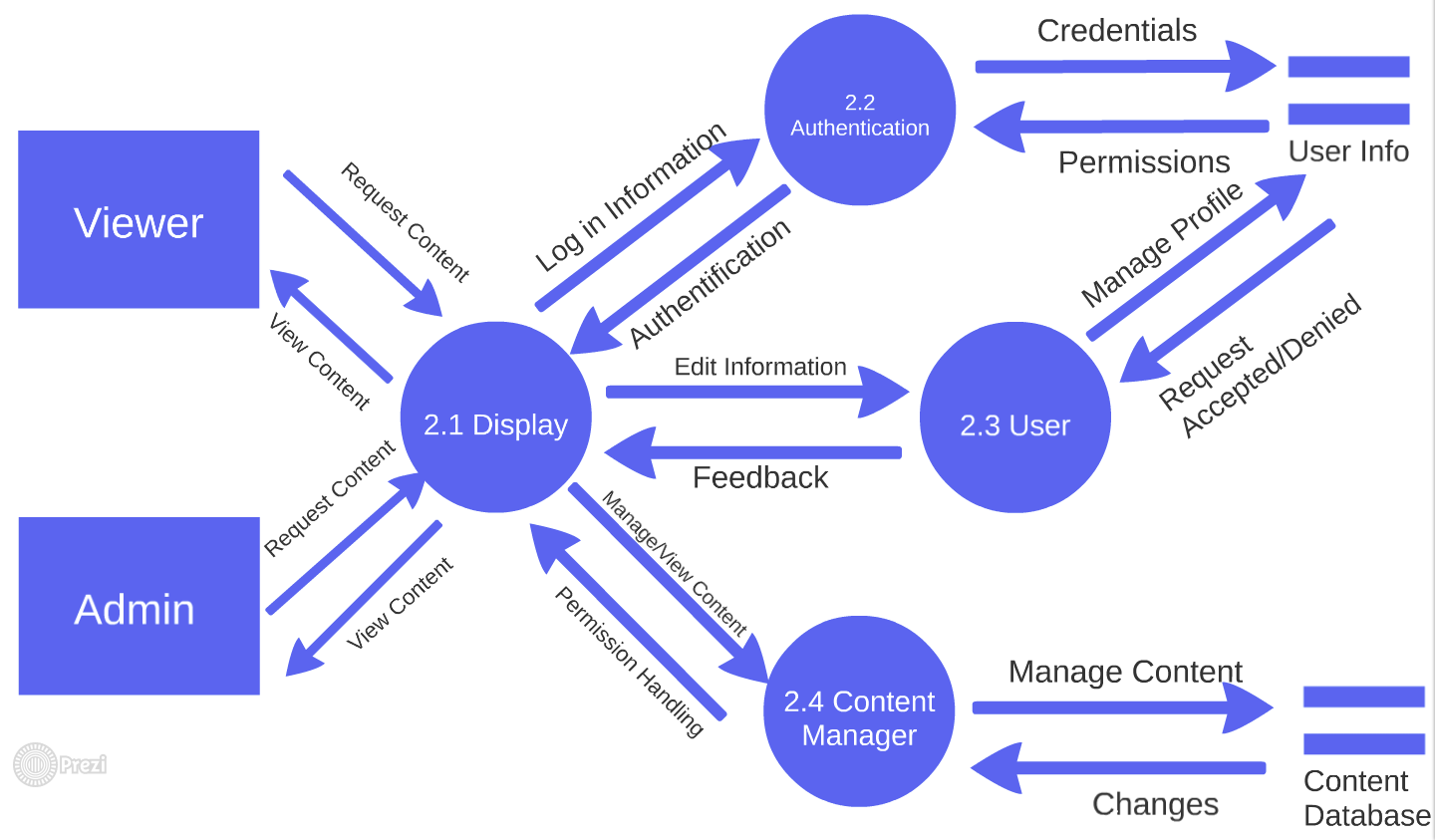
**2.0 DETAILED FUNCTIONAL DESCRIPTION**

2.1 Detailed **Kastle 2.0** Functional Description.

The major tool used to design **Kastle 2.0** is the Data Flow Diagram, DFD. The rational behind the selection of DFDs as the preferred design tool, was their simplicity and versatility. In the future more sophisticated tools may be used particularly if a correlation from Design to Requirement to Implementation and Testing is found to be a necessary addition.

2.1.1 Higher Level Data Flow Diagrams.

The **Kastle 2.0** major functional design components are shown in the DFDs below.



2.1.2 Detailed Description of **Kastle 2.0** Major Sub-Units

The **Kastle 2.0** major functional subunits shown in the DFDs in the previous sub-section, are described in detail below.

**Display Module (DM) - Module 2.1**

The Display Module (DM) will connect with every module to present the user an appropriate webpage. The DM will get user input to establish user credentials and then exchange information with the Authentication Module. Once a user has been verified the DM will reveal to the User the operations that they can use with the Content Manager Module based on their permission level.

2.1.1 View Module

The View Module (VM) controls the lay out of the page. It interacts with the other sub-modules in the DM to configure the appropriate webpage. Receiving information from the Menu Options Module, it will display the menu categories that a User is capable of seeing. Receiving information from the Theme Control Module, it will display the appropriate them that a User is capable of seeing, specifically if the user is an Admin or a general User. Receiving information from the Widgets and Plugin Module, the VM will display all the different functionalities the website should host.

2.1.2 Menu Option Module

The Menu Options Module (MOM) Is in control of connecting the VM with a User’s permission. The MOM will get permission information from the User Module. It will then send this information to the VM to enable it to display the appropriate categorizes and sub-categories based on the user’s permission.

2.1.3 Theme Control Module

The Theme Control Module (TCM) Is in control of displaying certain features to the Admin and omitting them from the Users. Once the User is signed in, the website will have already marked the user as an Admin or User. If the user is an Admin the TCM will display operations that the Admin can use in order to configure or change the website.

2.1.4 Widgets and Plugins Module

The Widgets and Plugin Module (WPM) is in charge of sending different Widgets and Plugins to the VM. These Widget and Plugins help configure the website and give added functionality to the website.

**Authentication Module (AM) - Module 2.2**

The Authentication Module (AM) is in charge of making sure all users that log in are credible users. It will offer different permissions to different users once they are logged in. these permissions are managed by the User Module but is enforced by the AM.

2.2.1 Login Request Module

The Login Request Module (LRM) is in charge of discovering new users. If a user attempts to login to the Website the LRM will redirect them to the Login page. Once a User has inputted their login information, the LRM will verify to make sure the user inputs are valid inputs. It will then send this information back to the AM to send the information the Sub Authentication Module.

Once the LMR has gotten the notification from the AM that a user has been verified, it will then redirect the user to their homepage. If the User has not been verified the user will be prompted to input information again. If the user cannot remember their username or password the LRM will allow them to request a change of information.

2.2.2 Authentication Module

The Authentication Module (AAM) is in charge of getting user credentials and matching it with the User’s database. If the credentials do not match any thing on record it will deny the user access. If the credentials do hit a match in the records, it will allow the user to process into the website with their permissions.

2.2.3 Profile Module

The Profile Module (PM) will allow the user to change their basic information. It will allow users to change their picture, the school they are associated with, their favorite pages, an their book marked pages.

**User Module (UM) - Module 2.3**

The User Module (UM) will only be visible to Admins, which will be verified through the AM. This module allows the Admin to create, delete, and change permissions of users.

2.3.1 Create User Module

The Create User Module (CUM) will allow an admin to create a new user. The Admin will have to input the User’s name, password, email, tableau credentials, and permissions. The CUM will connect to the User’s database to check for duplicate information and inform the Admin of any matches and allow then to change inputs.

2.3.2 Delete User Module

The Delete User Module (DUM) will allow an admin to delete an existing user. The Admin will be able to look at a list of existing users, and get more detailed information on ones that are chosen. Once a user has been determined the DUM will connect to the User’s database to delete the record of the user and inform the Admin of the changes.

2.3.4 Change Permissions Module

The Change Permission Module (CPM) will allow an admin to manage an existing user’s permissions. The Admin will be able to look at a list of existing users, and get more detailed information on ones that are chosen. Once a user has been determined the CPM will connect to the User’s database to check for the record and change their settings based on the inputs. It will then inform the Admin of the changes.

**Content Manager (CMM) - Module 2.4**

The Content Manager Module (CMM) will what a user can o in regards to the different content on the website. The main goal in this module is to use user’s permission level to determine if the user can view, create, edit, delete, or publish certain content.

2.4.1 Create Content Module

The Create Content Module (CCM) will allow User’s with permission to create a webpage to be published onto the website. If the User has permission, the page will be sent to the Content Authorizer Module. If the User does not have permission, the page will be sent to the Publish Content Module.

2.4.2 Edit Content Module

The Edit Content Module (ECM) will allow User’s with permission to edit a webpage to be published onto the website. If the User has permission, the page will be sent to the Content Authorizer Module. If the User does not have permission, the page will be sent to the Publish Content Module.

2.4.3 View Content Module

The View Content Module (VCM) will allow User’s with permission to view a webpage. If the User has permission, the page will be sent to the VCM to be displayed with the DM. If the User does not have permission, the User will be denied access to the page.

2.4.4 Publish Content Module

The Publish Content Module (PCM) will only be accessible to an Admin, which has already been verified through the AM. Here the admin can see a list of pages that are waiting to be created, edited, or delete. The Admin will have the power to either deny any of these actions or allow them through.

2.4.5 Delete Content Module

The Delete Content Module (DCM) will allow User’s with permission to delete a webpage already published into the website. If the User has permission, the page will be sent to the Content Authorizer Module. If the User does not have permission, the page will be sent to the Publish Content Module.

2.4.6 Content Authorizer Module

The Content Authorizer Module (CAM) acts as the gate to the Content Database. If a user has permission to create, delete, edit, or view a webpage, the module verifies their permission. It then requests the information from the database to be displayed in the DM. If the User does not have permission, the request is sent to the PCM. If the PCM has verified a change the CAM will then request the changes to occur.

**3.0 Kastle 2.0REQUIREMENTS**

3.1 **Kastle 2.0** Functional Requirements

This Section collects all **Kastle 2.0** Functional Requirements. The Section includes the complete set of functional requirements with explanation and rational where the statement of the requirement was deemed insufficient or needing additional background/justification. All requirements relate to the design modules described in Section 2. An effort has been made to standardize the correlation between the design modules and the requirements to make their access and organization more consistent. For example, module 2.1 requirements are labeled 3.1, sub-module 2.1.1 requirements are labeled 3.1.1 and so on. The list of requirements follows.

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.1 Display Module (DM) | |
| Requirement No. | Requirement Description |
| 3.1-1 | DM shall display a home screen with implemented Widgets. |
|  | DM shall display a home screen with implemented plugins. |
|  | DM shall display a login page. |
|  | DM shall show the KIPPLA logo on every webpage. |
|  | DM shall show the User name on every webpage. |
|  | DM shall show FAQ link on every webpage. |
|  | DM shall display menu options based on user permission. |
|  | DM shall display Tableau dashboards. |
|  | DM shall display comments on every webpage. |
|  | DM shall display favorite tag on every webpage. |

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.2 Authentication Module (AM) | |
| Requirement No. | Requirement Description |
| 3.2-1 | AM shall take user credential inputs. |
|  | AM shall verify user credentials. |
|  | AM shall check user permission when user tries to manage content. |
|  | AM shall allow user’s to change their profile picture. |
|  | AM shall allow user’s to change their name. |
|  | AM shall allow user’s to change their School. |

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.3 User Module (UM) | |
| Requirement No. | Requirement Description |
| 3.3-1 | UM shall verify the user is an admin |
|  | UM shall allow admin to delete users. |
|  | UM shall allow admin to add users. |
|  | UM shall allow admin to manage user permission. |
|  | UM shall allow admin to manage tableau credentials. |
|  | UM shall allow admin to manage user school. |

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.4 Content Manager Module (CMM) | |
| Requirement No. | Requirement Description |
| 3.4-1 | CMM shall check user permission. |
|  | CMM shall allow users to view content. |
|  | CMM shall allow users to view content. |
|  | CMM shall allow users to create content. |
|  | CMM shall allow users to delete content. |
|  | CMM shall allow admin to view content. |
|  | CMM shall allow admin to publish new content. |
|  | CMM shall allow admin to publish edited content. |
|  | CMM shall allow admin to delete content. |

**3.2 Kastle 2.0** Non-Functional Requirements

This Section collects all the Project-Acronym Non-Functional Requirements. All non-functional requirements are numbered “NF – n” where “n” indicates the nth requirement.

NF - 1 Kastle 2.0 must be operable by persons of the KIPPLA organization with ease.

NF - 2 Kastle 2.0 shall display information in real time.

NF - 3 Kastle 2.0 shall be able to go through the whole process without crashing.

NF - 4 Kastle 2.0 shall be able to handle the load of all users using the website at the same time.

**3.3 Kastle 2.0** Hardware Requirements

This Section collects all the Project-Acronym Hardware Requirements. All hardware requirements are numbered “H – n” where “n” indicates the nth requirement.

H - 1 KASTLE 2.0 will run on a separate web server currently provided by SiteGround.

H - 2 Kastle 2.0 will require a display device for supertile display. Each mobile device will have a touch screen display for simple input and screen capability, suitable for text and image displays. Otherwise each computer will have a navigation tool (mouse, touchpad, or keyboard) to use in order to input information.

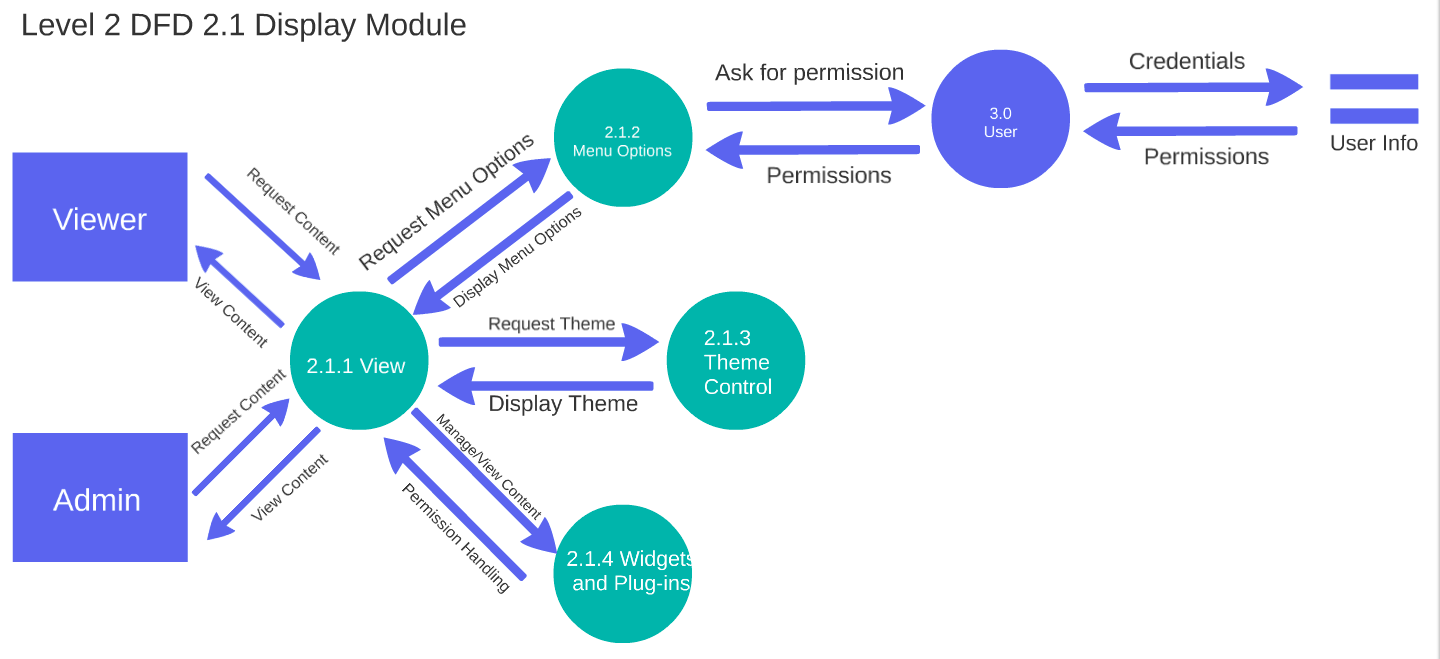
4.0 KIPPLA DETAILED DEISIGN

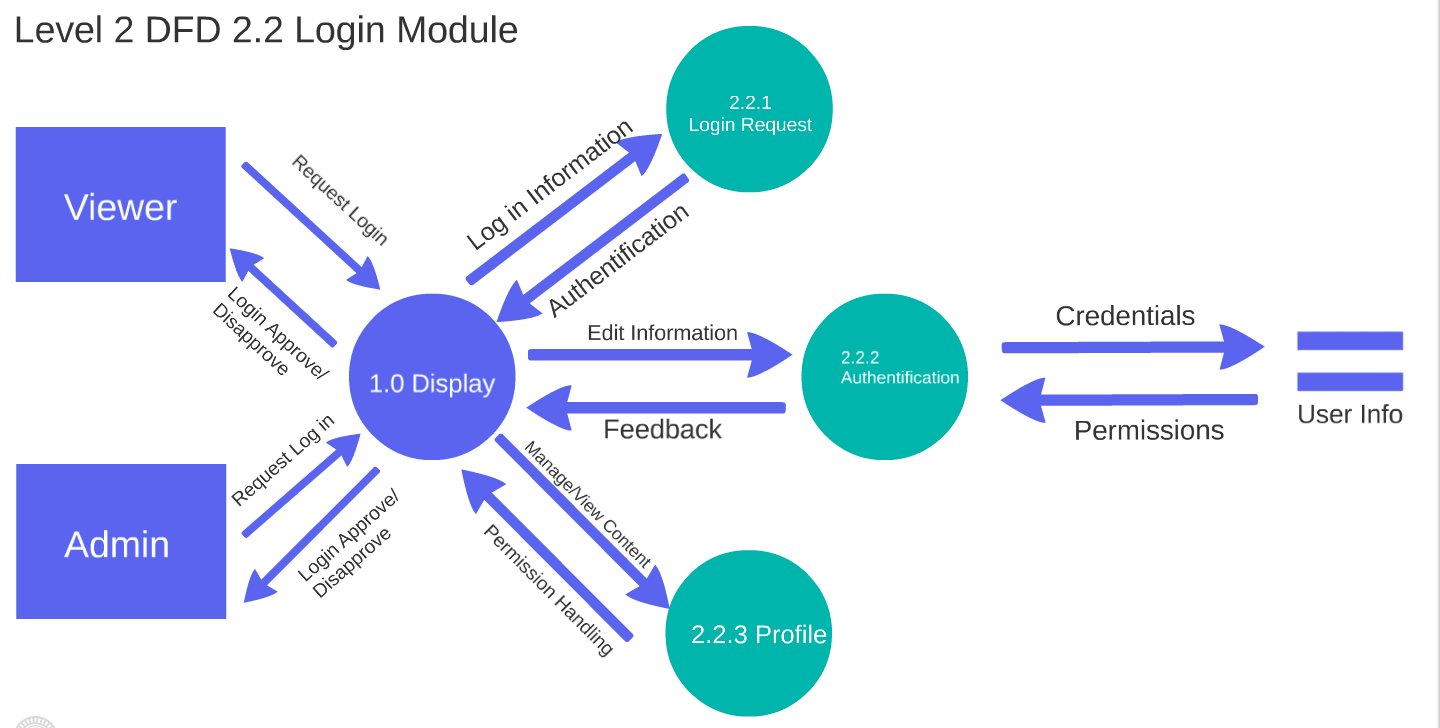
In this section the KIPPLA described in Section 2 with requirements listed in Section 3 will be designed

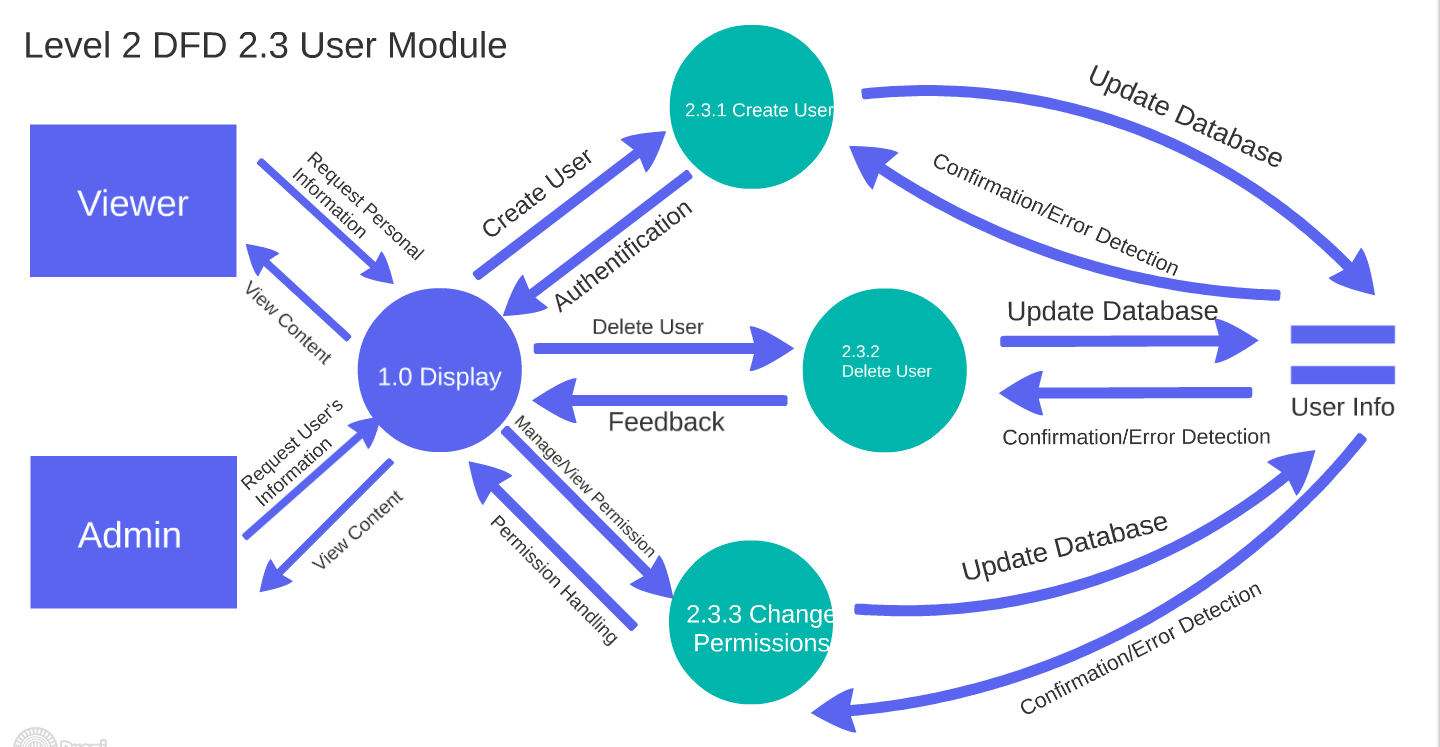
in detail possibly including higher level DFDs/ Each major module detailed design is included in

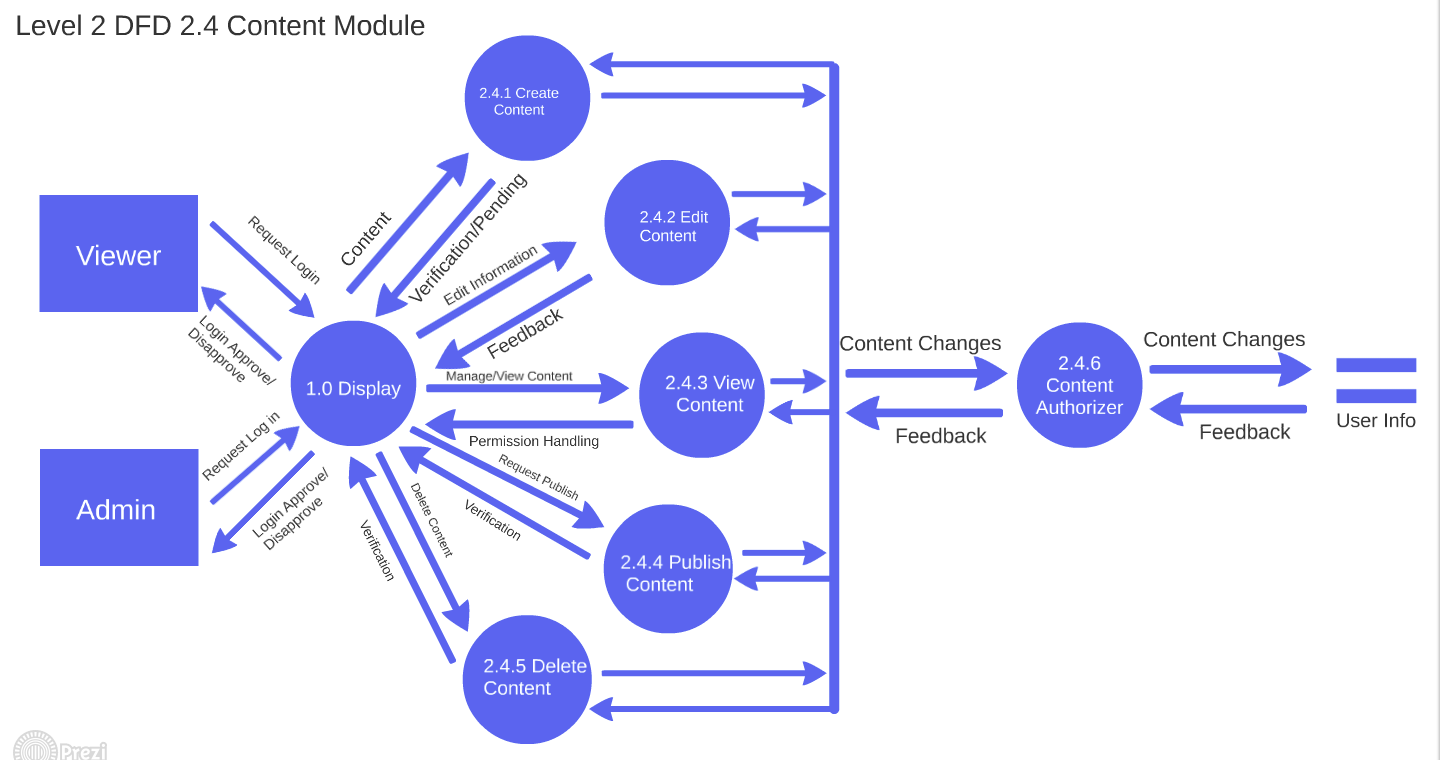
correspondence with the design sections defined in Section 2 and responding to the requirements listed

in its correlated sub-section in Chapter 3.









**5.0 *KIPPLA* ELEMENTS OF IMPLEMENTATION**

In this section (some of) the modules designed in Section 4 with requirements listed in Section 3 will be

implemented initially at least at the level of pseudo code. Where possible, actual code will be provided. Each

module is implemented in correspondence with the design sections defined in chapter 2 and responding to the

requirements listed in its correlated sub-section in chapter 3.

**A. ACRONYMS**

**B. DATA DICTIONARY**