

# **CSEDU - Onusondhan**

## **Software Design Document**

### **Submitted by:**

1. Golam Mourshid (51)
2. Sree Sowmik Kumar Sarker (54)



## TABLE OF CONTENTS

<b>1.</b>	<b>INTRODUCTION</b>	<b>Page No</b>
1.1	<i>Purpose</i>	4
1.2	<i>Scope</i>	4
1.3	<i>Overview</i>	5
1.4	<i>Reference Material</i>	5
1.5	<i>Definitions and Acronyms</i>	5
<b>2.</b>	<b>SYSTEM OVERVIEW</b>	6
<b>3.</b>	<b>SYSTEM ARCHITECTURE</b>	6
3.1	<i>Architectural Design</i>	6
3.2	<i>Design Rationale</i>	7
3.3	<i>Architectural Goals and Constraint</i>	8
3.3.1	<i>Security</i>	8
3.3.2	<i>Persistence</i>	8
3.3.3	<i>Reliability/Availability</i>	8
3.3.4	<i>Performance</i>	8
3.4	<i>System Logical View</i>	9
3.5	<i>System Process View</i>	9
3.6	<i>Modular Decomposition View</i>	9
<b>4.</b>	<b>DATA DESIGN</b>	14
4.1	<i>Data Description</i>	14
4.1.1	<i>Logical Data Model</i>	14
4.1.2	<i>Physical Data Model</i>	14
4.2	<i>Data Flow</i>	14
<b>5.</b>	<b>COMPONENT DESIGN</b>	15
5.1	<i>Analysis Classes</i>	15
5.1.1	<i>Class diagram</i>	15

5.1.2	<i>Collaboration diagram</i>	15
5.1.3	<i>Sequence Diagram</i>	15
5.1.4	<i>State diagram</i>	15
5.1.5	<i>Activity Diagram</i>	15
5.2	<i>Design classes (GUI, OS, Data)</i>	15
5.2.1	<i>Class Diagram</i>	15
5.2.2	<i>Interface</i>	15
5.2.3	<i>Resource</i>	15
<b>6.</b>	<b>HUMAN INTERFACE DESIGN</b>	15
6.1	<i>Overview of User Interface</i>	15
6.2	<i>Screen Images</i>	15
6.3	<i>Screen Objects and Actions</i>	16
<b>7.</b>	<b>USER REQUIREMENT VS COMPONENT TRACEABILITY MATRIX</b>	16
7.1	References	16

## Details

### 1. INTRODUCTION

#### 1.1 Purpose

The Software design document is intended to describe the architecture and system design of the project “CSEDU – Onusondhan”. This SDD will take care of the aspects of how to implement the software for its designated platform. This will also cover up the points on how to implement each functions to make the software usage and also the user-human interface that will make the software usable to the target audience.

#### 1.2 Scope

In the CSEDU department website a person or a student only can know which courses are taught here and where it is located or what is the number of professor but doesn't know which courses are most favourite / which courses are so analytical / the number of teacher currently stayed here etc. But by CSEDU –

Onusondhan all these types questions answer will get a user. The SRS specifies the requirements of the CSEDU – Onusondhan product. . The goal of the project is to create a platform to make a full right consciousness about CSEDU. By using this all the users can know all about CSEDU and also it's goal is to do that. CSEDU – Onusondhan will be a Web app built for the Web using the Flask framework. It will be platform independent and accessible on all devices with a web browser supporting Flask .

### **1.3 Overview**

We are willing to implement a website for the students which will contain all type of questions with answers about CSEDU students or enthusiasts. There will be questoins about CSEDU with answer, forum to ask questions as well as some other features like user's list and so on.

This document will be talking about the points given below.

1. UI design
2. Option placement on menu
3. Communication between different layouts
4. Input method from the user
5. Implementing firebase database system
6. Possible bug fixing

### **1.4 Reference Material**

1.4.1 Functional Requirements Document on CSEDU - Onusondhan written by Golam Mourshid and Sree Sowmik Kumar Sarker.

1.4.2 Non Functional Requirements Document on CSEDU - Onusondhan written by Golam Mourshid and Sree Sowmik Kumar Sarker.

1.4.3 Software Requirements Specification Document on CSEDU – Onusondhan written by Golam Mourshid and Sree Sowmik Kumar Sarker.

1.4.5 The websites that are need for theme and UI design of the projects are:

- i) <https://material.io/develop/web/>
- ii) <https://balsamiq.cloud/>
- iii) <https://www.draw.io/>

### **1.5 Definitions and Acronyms**

#### **Definitions:**

Flask : A micro-framework for Python based on Werkzeug, Jinja 2 and good intentions. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Onusondhan : To find something that is/are necessary.

**Acronyms:**

CSEDU : Department of Computer Science and Engineering , University of Dhaka.

## **2. SYSTEM OVERVIEW**

This website for the students which will contain all type of questions with answers about CSEDU students or enthusiasts. Here user's can be a member of it by doing registration and filling the necessary condition. Only member's can ask question and can give the answer . But all type of user's can view answers. The write answer will be marked true by the admin after judge. Here a member can also see the list of another members.

The design will be simple enough to be able to navigate through all the options without any explanation. User will know right away what option does what by their naming or placements.

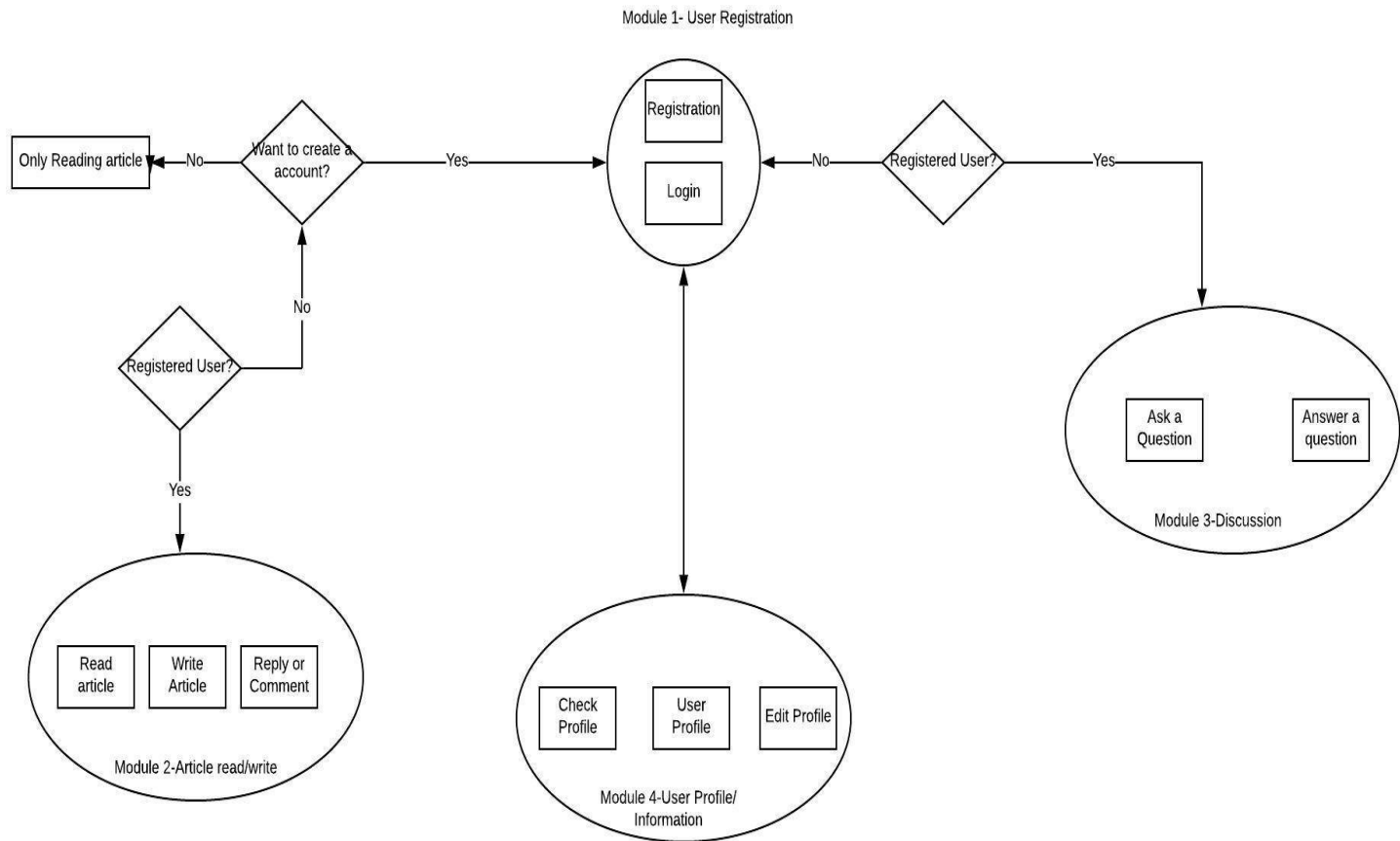
## **3. SYSTEM ARCHITECTURE**

### **3.1 Architectural Design**

Architectural design represents the structure of data and program and components that are required to build a computer-based system. It considers the architectural style that the system will take, the structure and properties of the components that constitute the system and the interrelationships that occur among all architectural components of a system. We follow the following steps in our architectural design process of CSEDU – Onusondhan.

1. Represent the system in context.
2. Define archetypes
3. Refine the architecture into components
4. Describe instantiations of the system.

In following sections, we will represent these steps.



In module 1 by registration a user can be a member of this system. If already register then he/she can login otherwise not.

In module 2 for writing an article or asking questions or for comment an user need to be logged in. Although all type of user can see the articles and can read the questions and answers.

In module 3 only valid user's can ask questions and reply answer for a question. But for this the user need to logged in.

### 3.2 Design Rationale

The architecture above is chosen because it's linear, And the project doesn't need any complex design structure in order to be built. This architecture is also

selected to reduce complexity in implementation. We removed some dependencies according to their lowest priority as they don't possess much importance.

### **3.3 Architectural Goals and Constraint**

Main goal for this architecture was to undergo the simplest relations between modules and sub-modules. So that developers find it easy to implement as well as users can navigate easily without thinking to figure out how a feature can be used.

#### **3.3.1 Security**

There are sign up & sign in page for the user authentication. For posting anything in the page or interact with the admin, user must be registered. Valid email address must be used, because there is a account validation issue. User data is saved in firebase database . For reading/accessing in the site, no authentication is required. So , user info is not editable by others and anything saved on database can't be modified other than the admins.

#### **3.3.2 Persistence**

Every data input by user in the html form / other input labels are stored in and collected from firebase database by using default and built-in methods. So there is no inconsistency in the stored and showed information.

#### **3.3.3 Reliability/Availability**

There is no precision error in the information showing as we used built in methods to get the information stored in database. There is no other custom encryption process, so no error can occur while retrieving data from database and load on website. Data can only be corrupted in transmission layer or data link layer of the process if user has bad network connectivity or corrupted browser which doesn't support our character standard (UTF-8)/architecture standard.

#### **3.3.4 Performance**

Though this is an online app. Major performance is dependent on the network bandwidth/speed. For profile image upload, there will be at most 1-2 seconds delay if the bandwidth speed is relatively high. This App is responsive. Any size of device can hold this app (Android/ipad/laptop/desktop) as this is a web application. CSS will work beautifully in any kind of supported device and browser (Mozilla/ Google



Chrome/ Opera/ Internet Explorer etc.).

### **3.4 System Logical View**

The whole system was designed with python and Flask framework. There are no classes or class diagram for the system. But we have different packages and blueprints for different modules/ sub-modules. The system architecture design was illustrated on 3.1 where we find 4 different modules. Now module 1 is considered level 1 and rest of them are on level 2 as most of the modules depends on user info and registration. Only module 2.1 is independent from others for generalization.

### **3.5 System Process View**

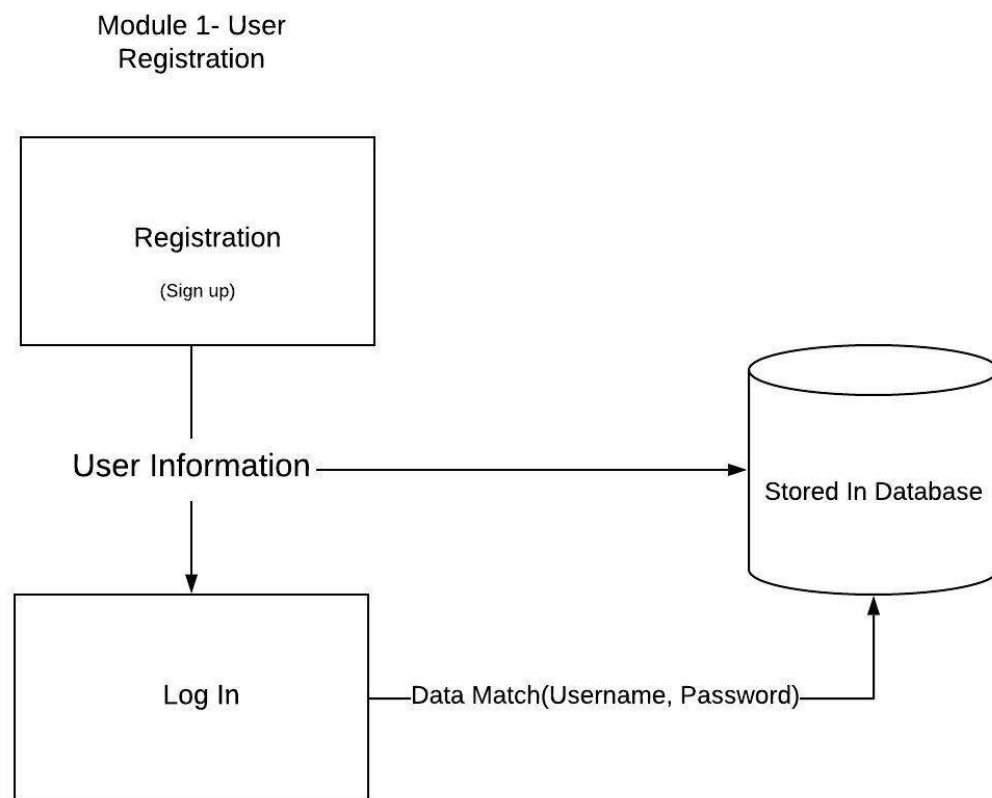
A description of the process view of the architecture. Describes the tasks (processes and threads) involved in the system's execution, their interactions and configurations. Also describes the allocation of objects and classes to tasks.

On our system architecture most of the processes are independent. But the processes in sub-modules that are dependent on module 1 keeps some data from process of module 1. A session key is stored for the logged in user and used while writing a new article in module 2 or ask a question/reply on module 3. Information shown on user profile process(module 4) directly depends on the data from module one login process. But no thread or two different processes are run concurrently in this system.

### **3.6 Modular Decomposition View**

Provide a decomposition of the sub systems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object oriented description. For a functional description, put top level data flow diagram (DFD) and structural decomposition diagrams. Also include state transition diagram, sequence diagram, activity diagram whenever appropriate. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

All the modules are illustrated below with the data flow (DFD)-



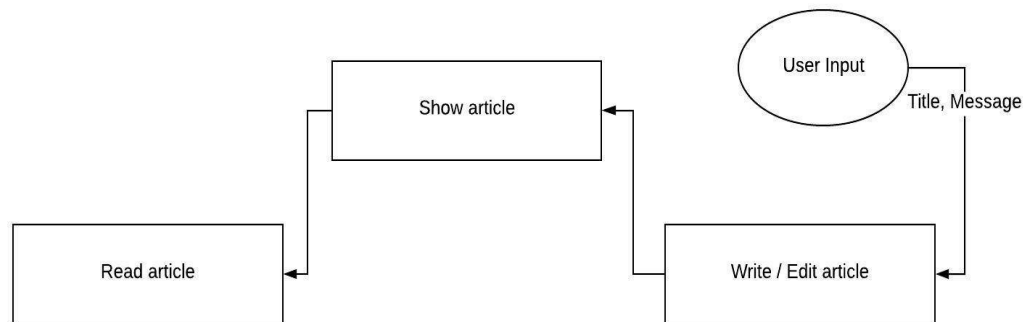
#### **Module-1:**

##### **Registration:**

User prompts input in the register form and submit. User data is stored in database and redirected to login page.

##### **Log in:**

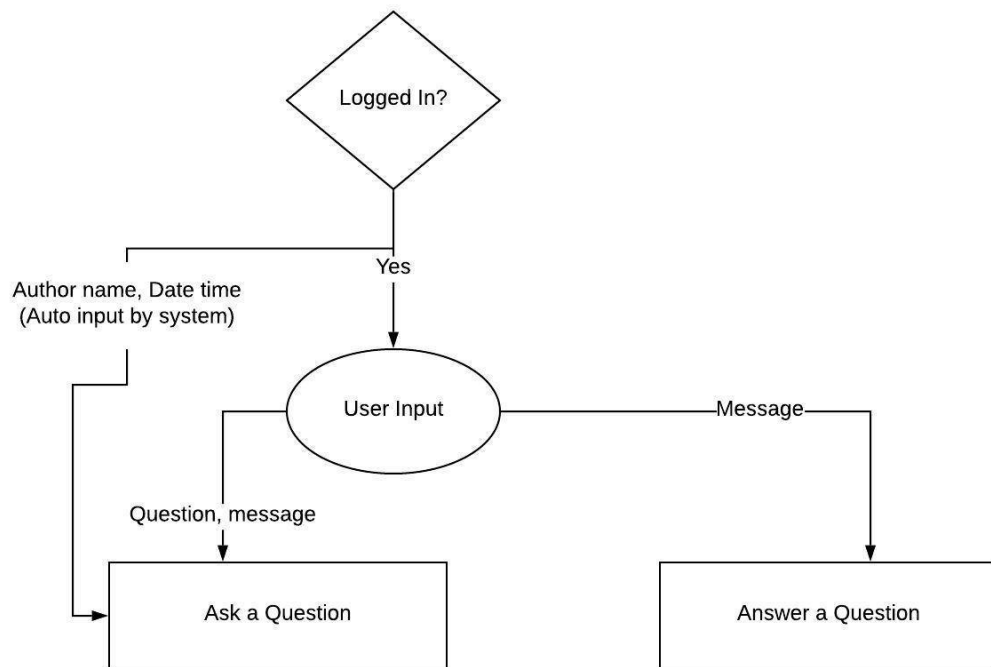
User inputs the credentials again. User input is compared with the stored data in database and grant/revoke access to next level modules.



Module 2- Article  
read/write

### Module-2:

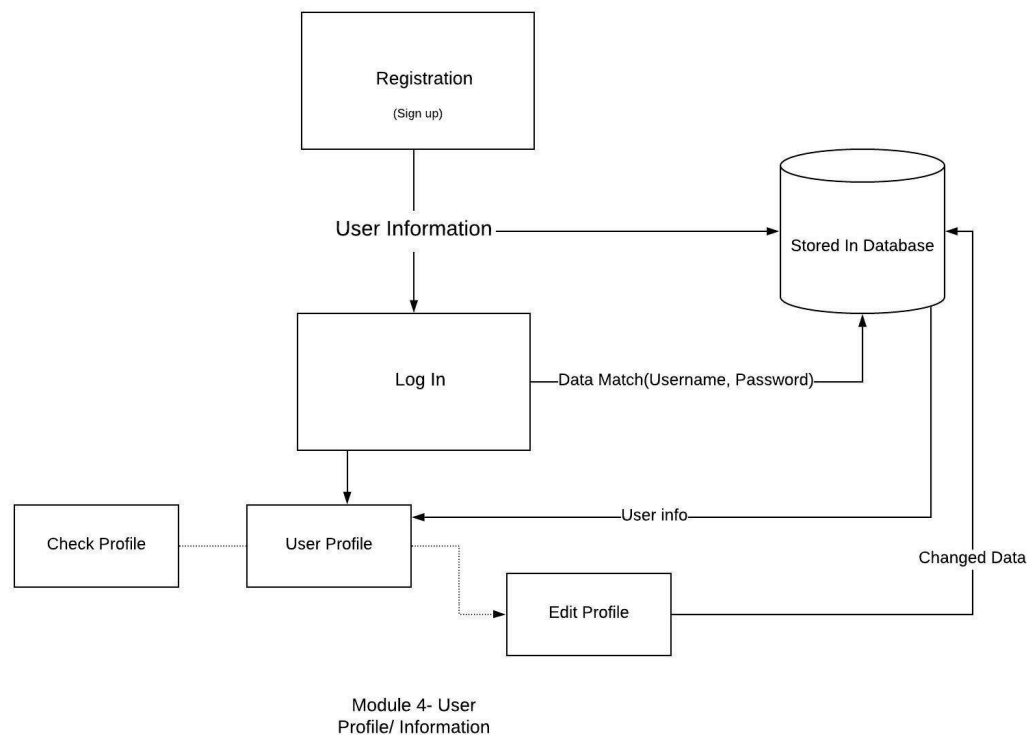
User prompts to create a new article(if signed in) Gives input data as necessary for the writing (title, images, description, ...) Then, data is stored in database as string/ embedded format. Shown in homepage/ articles page section. Users can read all the articles from that section even without being registered or logging in. But editing/deleting must be done by author himself/admin while logged in.



Module 3-  
Discussion Forum

### Module-3:

If user is logged in he can put questions in forum and reply a question. Can read responses of questions without logging in can search or read questions and responses. For question user has to put question title and message manually. User name and date time is automatically retrieved by system.



### Module-4:

If user is registered his/her information is stored in database as given input. Login process check credential in matches or not. Then profile process shows information from database according to matched credentials. Information of profile can be edited by the user himself while logged in.

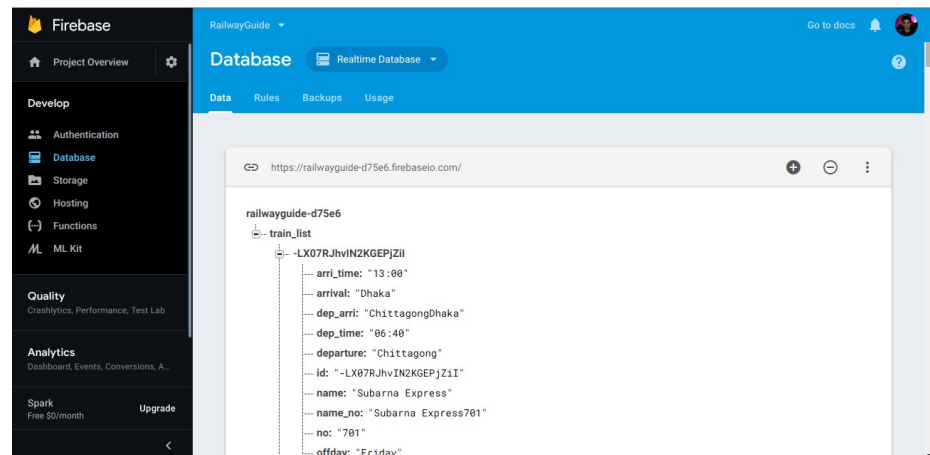
## 4. DATA DESIGN

### 4.1 Data Description

#### 4.1.1 Logical Data Model

As we are using Firebase database and it keeps data in Json tree format . The database is not structured as traditional DBMS for flexibility and scalability.

Example: (mlab DB storage)



#### 4.1.2 Physical Data Model

The databases for this system follows NoSQL standard so informations are stored as collections. Only Username and password combination are unique as can be considered as primary key.

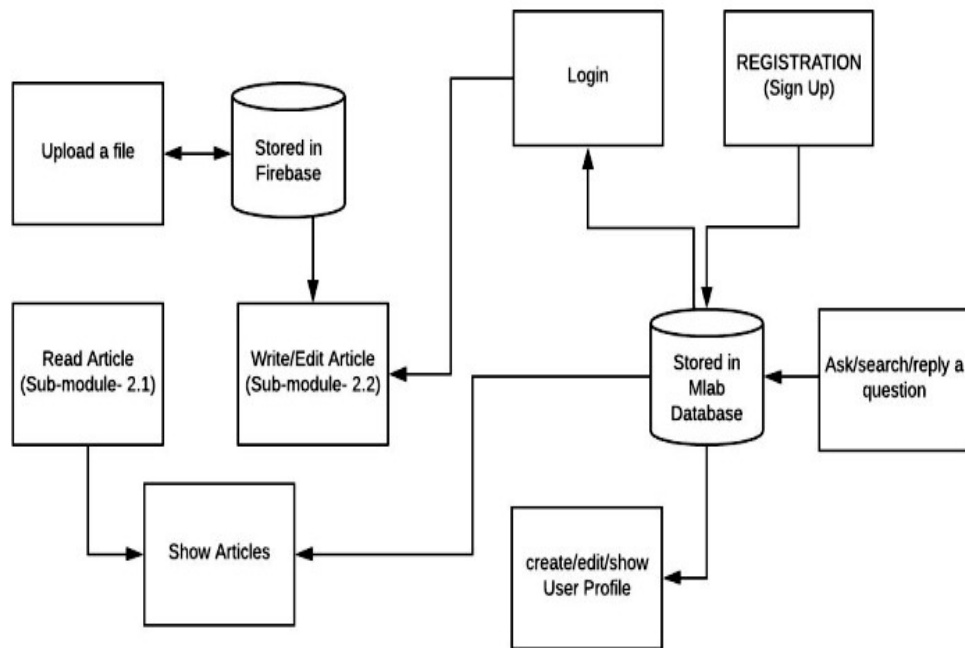
The relationship between data is already described via DFD on section 3.5.

### 4.2 Data Flow

As already explained on section 3.5 via DFD and system flow diagram how data is stored , loaded and used from database, module 1,2,3,4 that requires user input stores data into Mlab database and module 5 stores data into firebase database. And the flow is illustrated on section 3.5.

## 5. COMPONENT DESIGN

### 5.1 Activity Diagram



## 6. HUMAN INTERFACE DESIGN

### 6.1 Overview of User Interface

The user interface will be designed to be similar to the standard web interface, a white background with black text is chosen as our standard because the ease of visibility and ability to read. The flow of interface will be has been designed around the idea of searching for items, updating information, and displaying the returned . After displaying the making new questions information it will also give opportunity to edit this questions answer. After selecting post button the information will be posted . Also can delete any king of data. Deleting data will not appear in font of the user.

### 6.2 Screen Images

### 6.3 Screen Objects and Actions

In the homepage, There are some buttons such as Home, Ask Question, Articles, Tags, Login, Registration. By Clicking Articles, user can read and create articles. By Clicking in the ask question user can ask different types of questions and in reply button user can answers some others questions answer. In the user list option user can see the other member's in this site. Besides every questions and answers there is a tag that defines the questions type. If the answer of the question if valid then after judging the admin of the system will give a accepted answer sign.

## 7. USER REQUIREMENT VS COMPONENT TRACEABILITY MATRIX

A matrix comparison that shows completion/ coverage status different user requirements stated in SRS-

Requirement ID- User Requirement	Functional Requireme nt	Non- Functional Requirement	Testing Requirement	Performance requirements
R001- Sign Up	Pass	Pass	Pass	Pass
R002- Sign In	Pass	Partial	Pass	Pass
R003-Ask Question	Partial	X	Pass	X
R004-Search Question/tag	Pass	Partial	Pass	Pass
R005-Reply or answer	Partial	X	X	Pass
R006- Users	Pass	Pass	Pass	X

References :

- i) <https://www.slideshare.net/NadialIT/software-design-document-42181210>
- ii) <https://www.guru99.com/traceability-matrix.html>
- iii) [Templates - Journals, CVs, Presentations, Reports and More - Overleaf, Online LaTeX Editor](#)