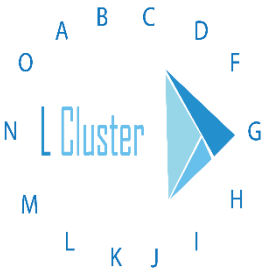


L Cluster

Information System Is

Learning-Oriented Social Media Service

Learning Cluster

Professor Supervisor: Dr. Ayman Mustafa

Assistant Supervisor: Eng. Mahmoud El Mahdy

Prepared by:

1. Mohamed Soliman Gomaa
2. Ahmed Abdel Salam Saad
3. Ahmed Abdel Aziz Abdel Azim
4. Mohamed Tarek Hussini
5. Hagar Mohamed Ali
6. Mai Ashraf Mohamed

Abstract:

- It’s a work & educational-oriented social networking service that operates via web app.
- Allow users with same career to communicate by publishing posts and interacting with them or by messages.
- We need to development the system of communication for increase the cooperation
  rate.
- L Cluster is a web project helps to interest by the life of people, it keeps the people at the same specialization up to date with each other and with each other all the time.
- Cluster can easily find people of their field any time easily and also can help each other easily.
- The system helps Doctors, engineers, pharmacists and people at all fields over the world to communicate with each other, talk each other, share their experience together.

- The project facilitate the communication, cooperation, web development, information gaining, and proximity of distances.

Table of Contents

Table of Figures

Chapter 6

6. Project design

Chapter7

7. Project implementation

Chapter 1

Introduction

1.1 purpose:

We will begin by defining **Cluster:**

- Group of independent servers (usually in close proximity to one another).
- Interconnected through a dedicated network to work as one centralized data processing resource.
- Clusters capable of performing multiple complex instructions by distributed workload across all connected servers.
- Clustering improves the system's availability to users, its aggregate performance, and overall tolerance to faults and component failures.
- The system requirement specification document is specifically designed to delineate the boundaries of the communication Information System design and functionality. Parties interested in this documentation would include but not be limited to the system owners, the system users,
- The project manager and the design team.
- A free system for departments to facilitate communication.

1.2 Problem Definition:

- There are many people over the world at the same specialization, they can't communicate directly or like face to face. One can search to find his coworker and it will take time.
- People can communicate, but people at the same department over the world can hardly intercommunicate with other.

1.3 Project Solution:

- The best way for communication currently is online like chat online or publish posts or publish on groups.
- This will be available for people of the same departments or specialization.
- Over the world; many people of Cluster can communicate easily and in cooperative way, it could be by a system or project like a web page.

1.4 Time Schedule:

The time schedule of tasks that are finished successfully, and the duration to finish.

Task	Duration(weeks)
Gathering Information	3
Planning	3
Analysis	4
Design	3
Implementation	4
Testing	3
Maintenance	2

1.5 Time line of the project schedule:

Figure 1 shows the time line of the project schedule. Figure 2 shows the completed and in progress milestones. Finally, Table 1 shows the detailed project tasks.

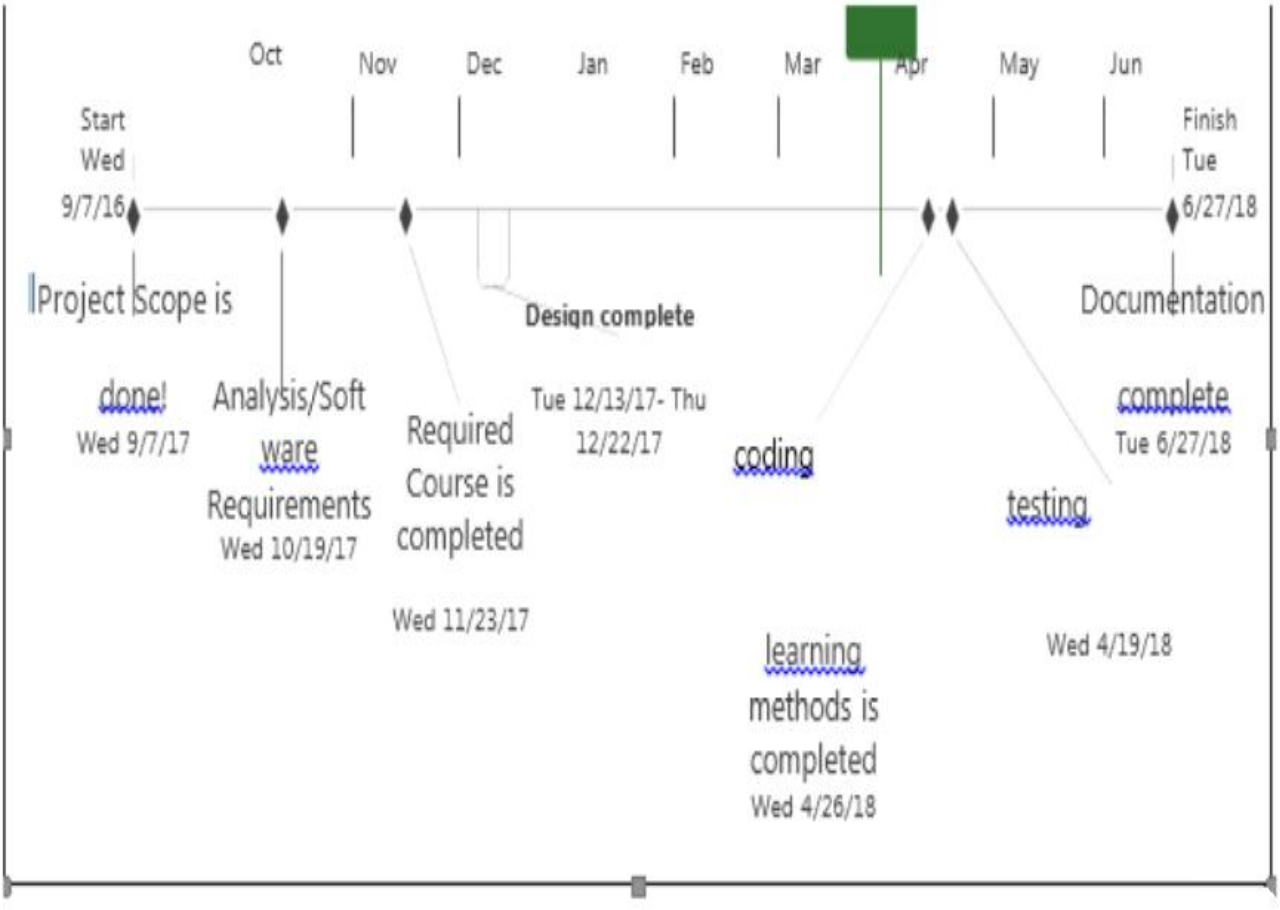


Figure 1

1.6 Project Plane:

- L Cluster is an environment on the internet for coworkers and shared departments.
- It helps to communication between people of special departments and common organization to develop fields like medicine, engineering, science and computer and information.

1.7 Project objective:

- Easy communication between clusters and common organization.
- Publish and display scientific researches.
- Special page for each specialization for easy search and access to the target.
- Provident most of scientific books.
- Display all scholarships.
- Display all medical news.
- Provides teamwork.
- Displays the conference online.
- Saving time and power.

1.8 Project Team:

This screen shows the team of the project.

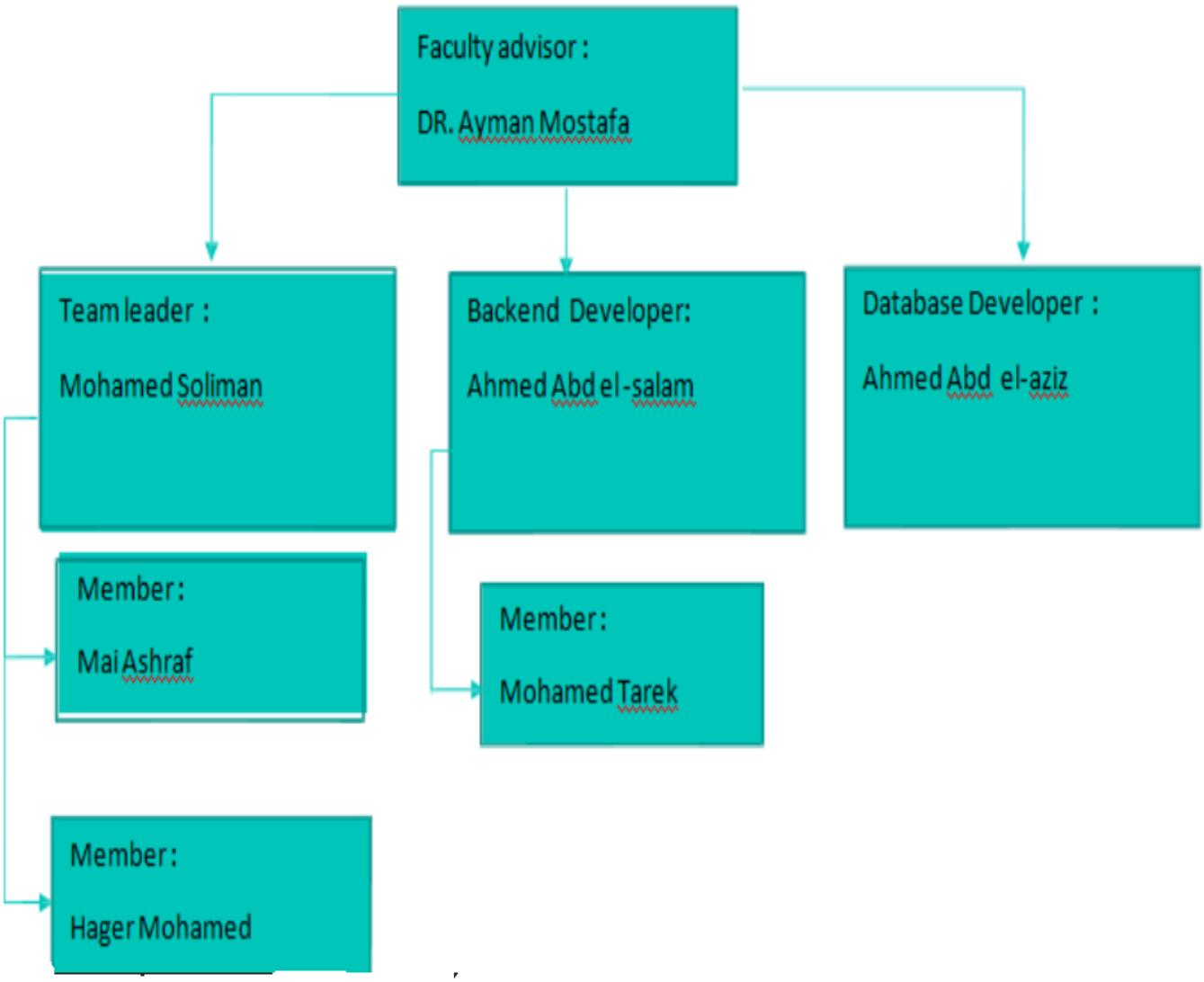


Figure 2

Chapter 2

2. Description of Project

2.1 Business model:

The screen of Business Model Canvas.

The Business Model Canvas

Team or Company Name:	Date:

Key Partners	Key Activities	Value Proposition	Customer Relationships	Customer Segments
• Content provider	• Platform development. • Data (Bigger data) management.	• Connect with others they have the same career. • Learning (easier and more enjoyable). • Share professional (vocational) live. • Provide teamwork environment. • Find help in solving technical problems. • Take advantage of the experiences or even the problems of others.	• Same-side network effects. • Cross-side network effects.	• Students or employees who believe that collaborative learning, discussion and teamwork are effective ways of learning. • Advertisers
	Key Resources • Platform. • Technology infrastructure.		Channels • Website. • Adds. • Mobile app.	

Cost Structure	Revenue Streams
• Cloud (or data centers) cost. • Development cost. • Marketing cost.	• Free. • Ads revenues.

Figure 3

Chapter 3

3. Feasibility Analysis

3.1 Technical Feasibility:

- The risk regarding familiarity with applications (as a social media web app) is moderately high.
- The Marketing Department has little experience.
- Numerous social media sites exist on the Internet.
- The risk regarding familiarity with the technology is moderately low.
- Back-end developers have knowledge of the current Web-based order entry system and the databases and Internet technology it uses.
- Front end developers have knowledge of the current web design technology to make the interface pleasing to the eye and simple to use.
- The project size is considered medium risk.
- The project team will likely consist of only 6 people.
- The project time frame is somewhat critical.

3.2 Economic Feasibility (cost-benefit analysis):

Development Costs:

1. Development training.
2. Data conversion costs.
3. Hardware and software.
4. Consultant fees.

Operational Costs:

1. Software upgrades.
2. Software licensing fees.
3. Hardware repairs.
4. Hardware upgrades.
5. Operational team cost.

- The main source of profit in this system is Advertisements. At the beginning, a small number of users may be missing this source.

3.3 Organizational Feasibility:

Users register by the university from which they graduate or are still a student,

But there may be more than one risk:

- There may be some difficulties in reaching all universities worldwide.

- How to convince users at the beginning of the application.

3.4 Requirements Definition:

3.4.1 Functional Requirements:

Edit his data:

- The system will allow users to edit your personal data.
- The system will allow users to edit your career data.

Publish and edit posts:

- The system will allow users to publish Post s contain text, image(s), and/or video(s).
- The system will allow users to determine privacy for each post.
- The system will allow users to edit your posts.
- Show and interact with other posts.
- The system will allow users to see posts that recommended by the system.
- The system will allow users to make like, share, comment, and reply on comments.
- The system will allow users to report posts.
- Define relations among user.
- The system will allow users to follow /unfollow other users.
- The system will allow users to block/unblock other users.
- The system provides chat between two or more users.
- Create or join to groups (team works).
- The system will allow users to create groups and determine the group privacy, objectives and roles.

- The system will allow group owner to employee group admins.
- The system will allow group admins send and receive (accept or revise) group requests.
- The system will allow group member to interact in group as group rules.

3.4.2 Nonfunctional Requirements:

Operational:

- The system database will be constructed to facilitate setting and getting data.
- The system will run on any Web browser.

Performance:

- System designed in way that Minimize user effort and facilities user's communication.

Security:

- User information will be secured.
- User messages, following data will be secured.

Cultural and political:

- Make sure the all members are doctors, medical organizations or are related to medicine.

3.5 USECASE Diagram:

The screen shows USE CASE Diagram.

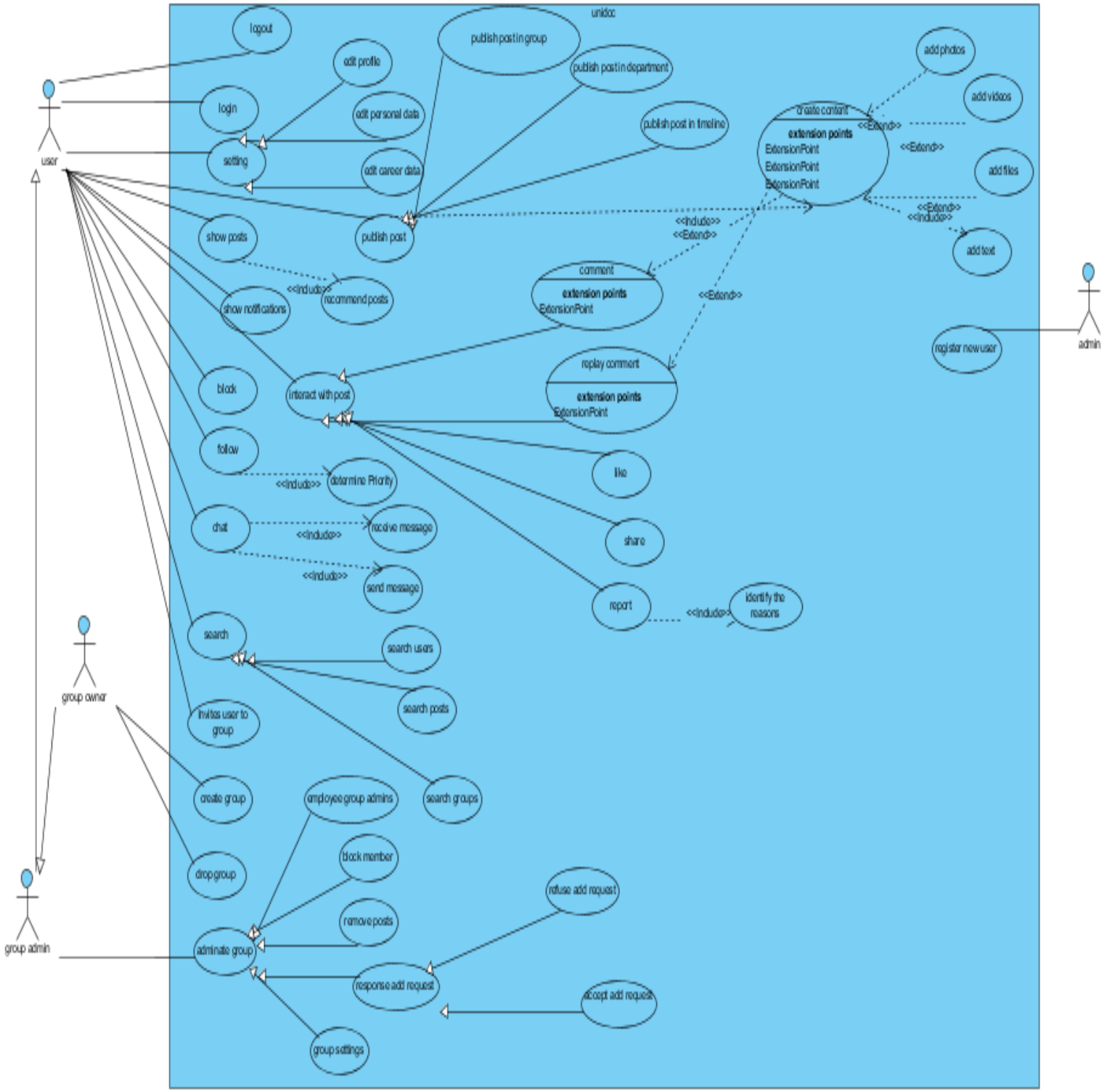


Figure 4

3.6 Sequence Diagram:

The screen of sequence Diagram.

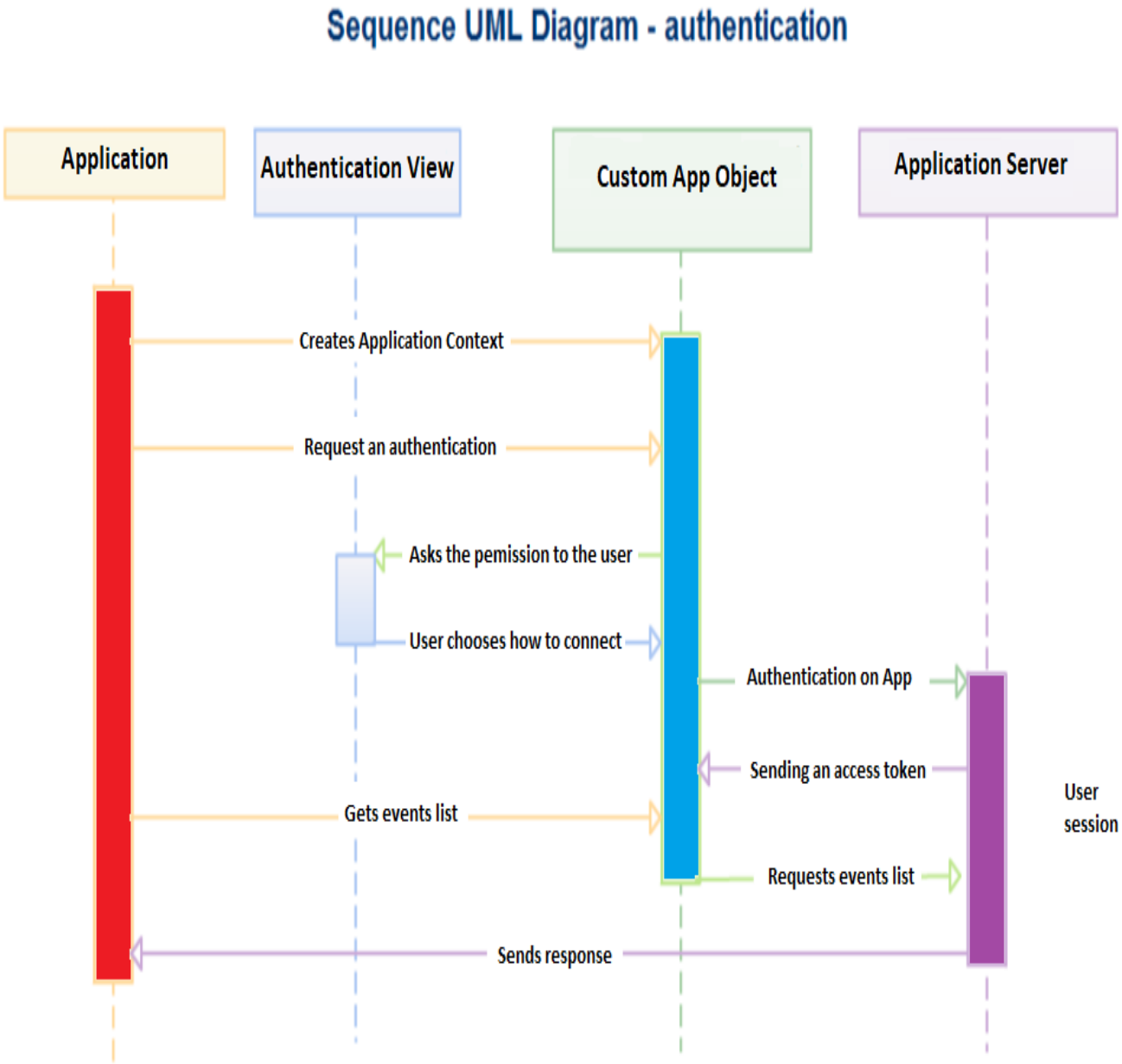


Figure 5

3.7 login and registration:

The picture shows the login and registration and how it work.

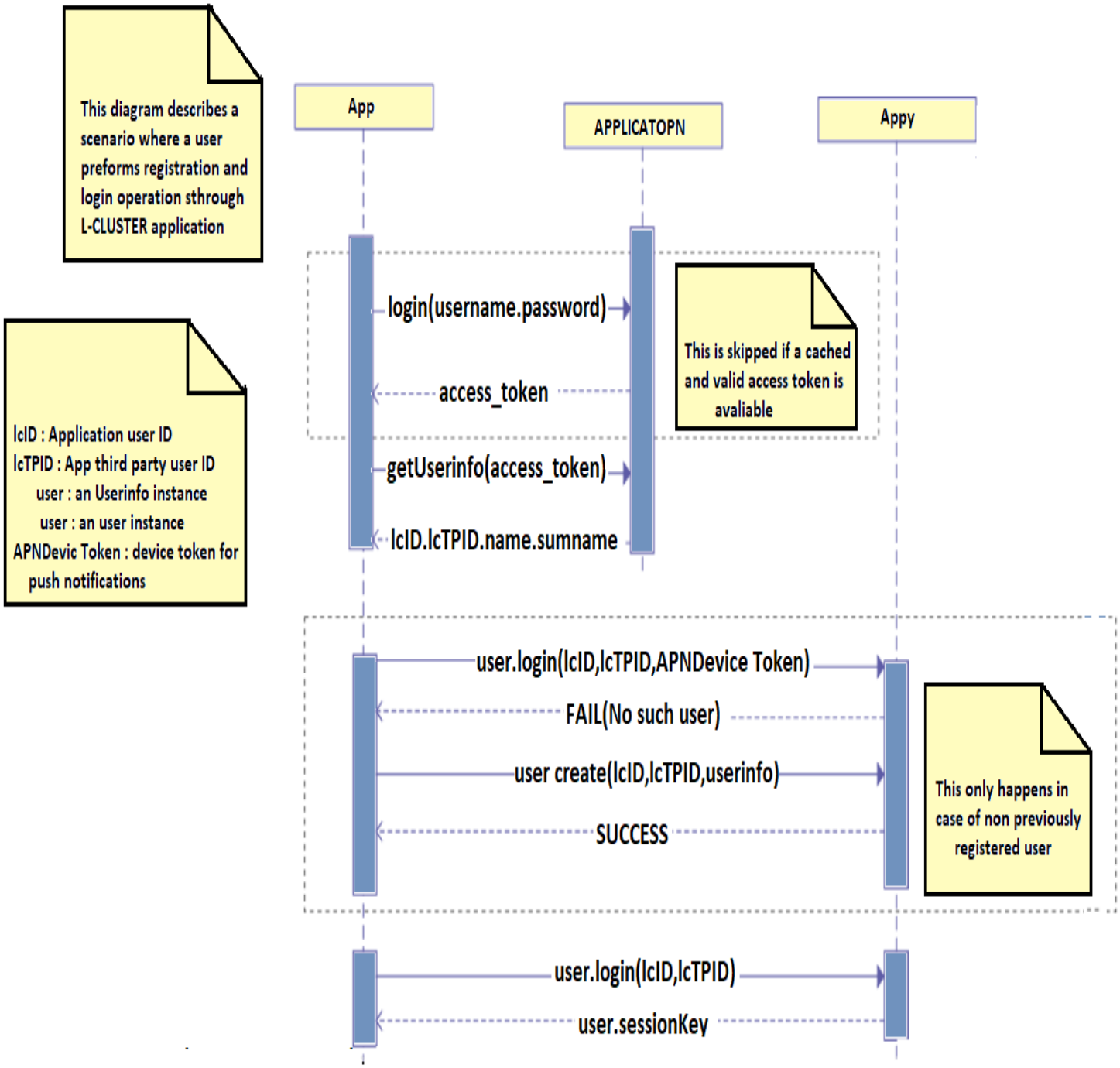


Figure 6

Chapter 4

4. Data Flow Diagram:

4.1 Context Diagram: The first diagram of data flow diagram shows the context diagram.

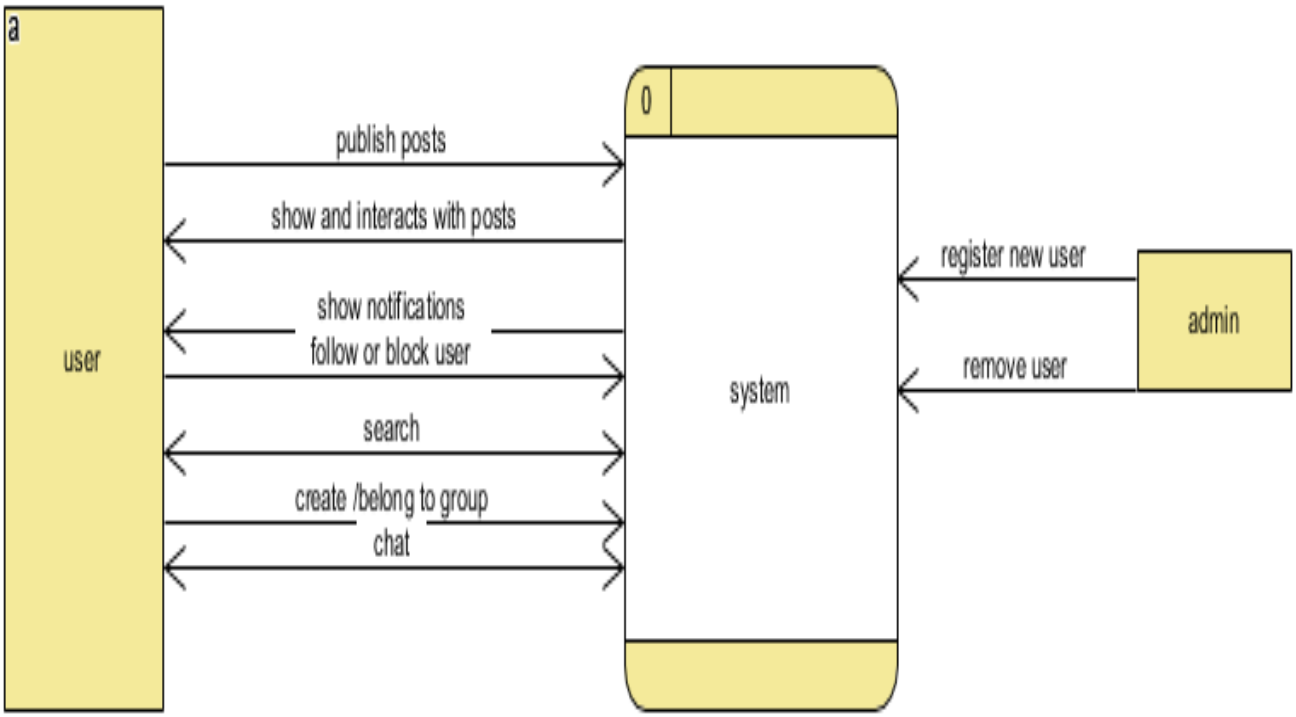


Figure 7

4.2 Level (0):

The picture shows level 0 of the project's data flow diagram (first level).

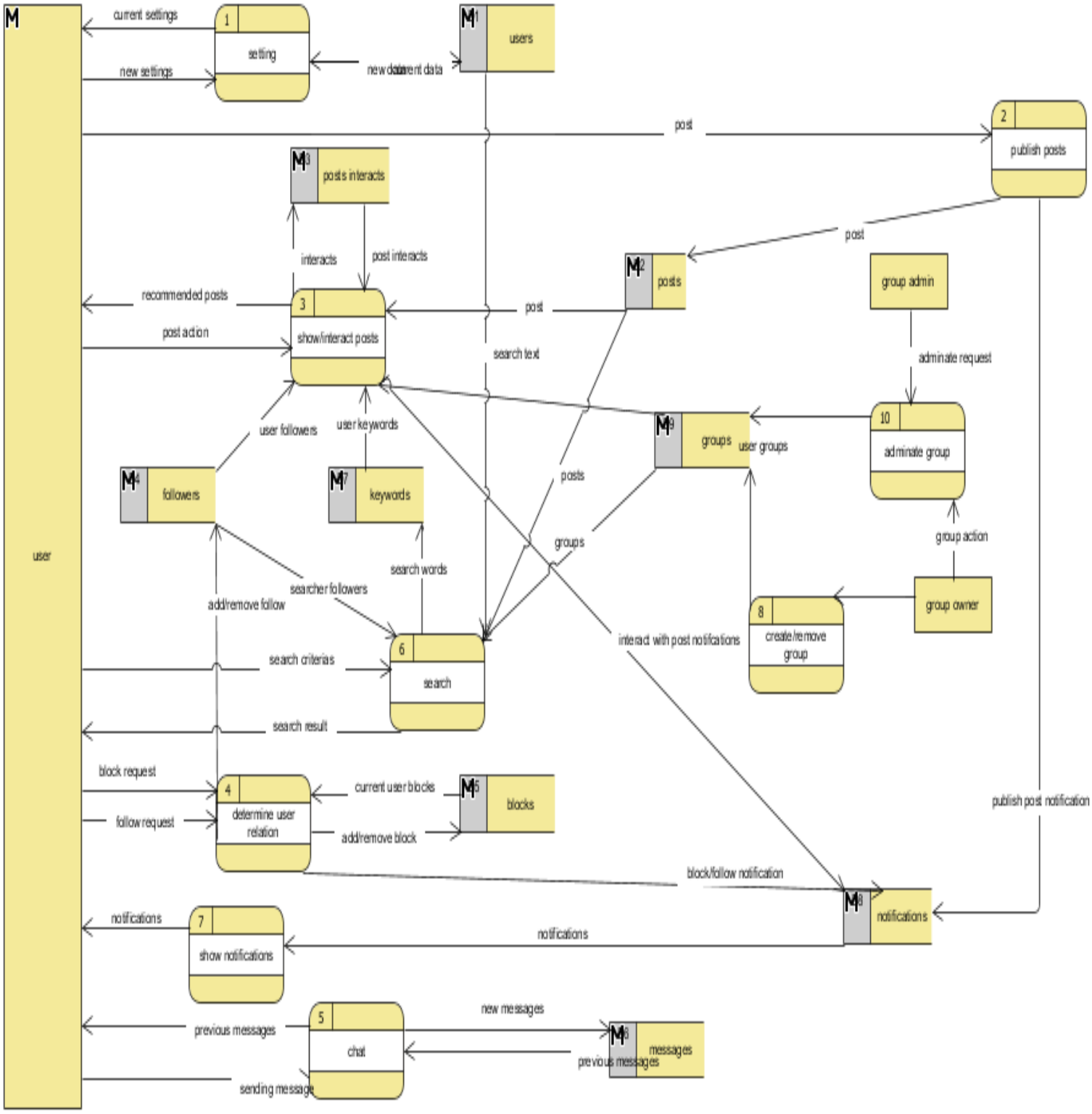


Figure 8

4.3 Level (1):

The picture shows level 0 of the project's data flow diagram (second level).

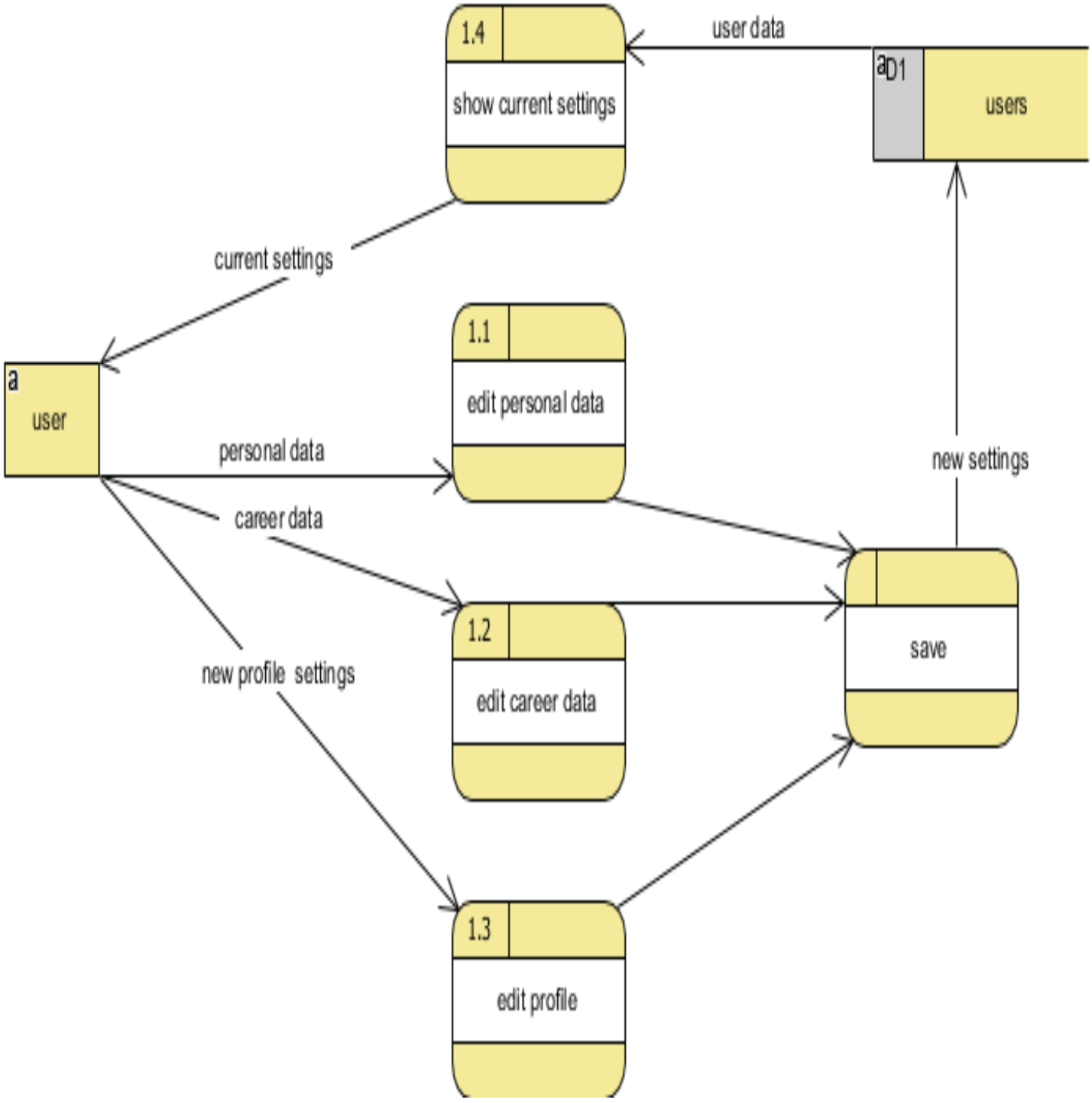


Figure 9

Level (1) cont.:

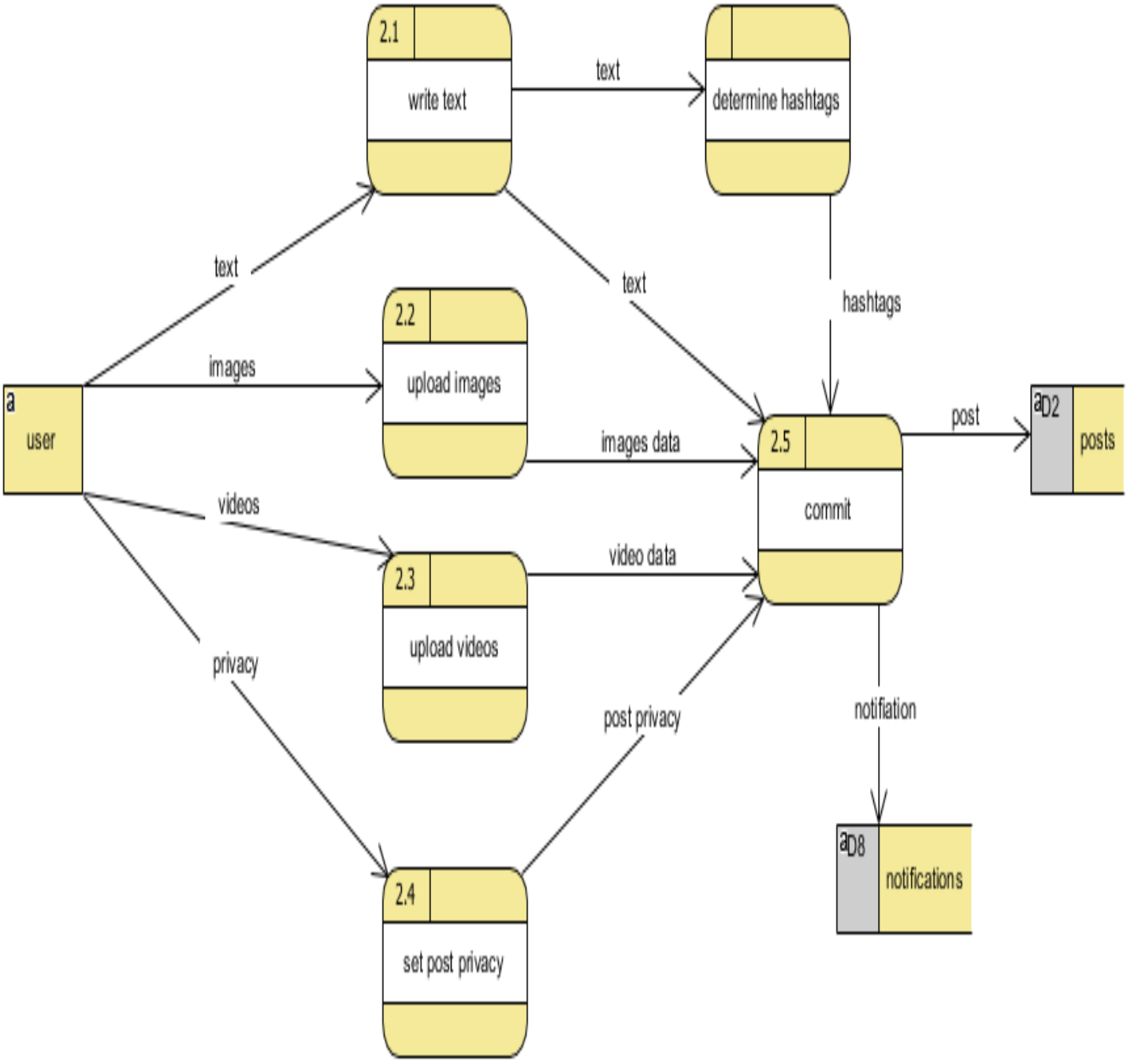


Figure 10

Level (1) cont.:

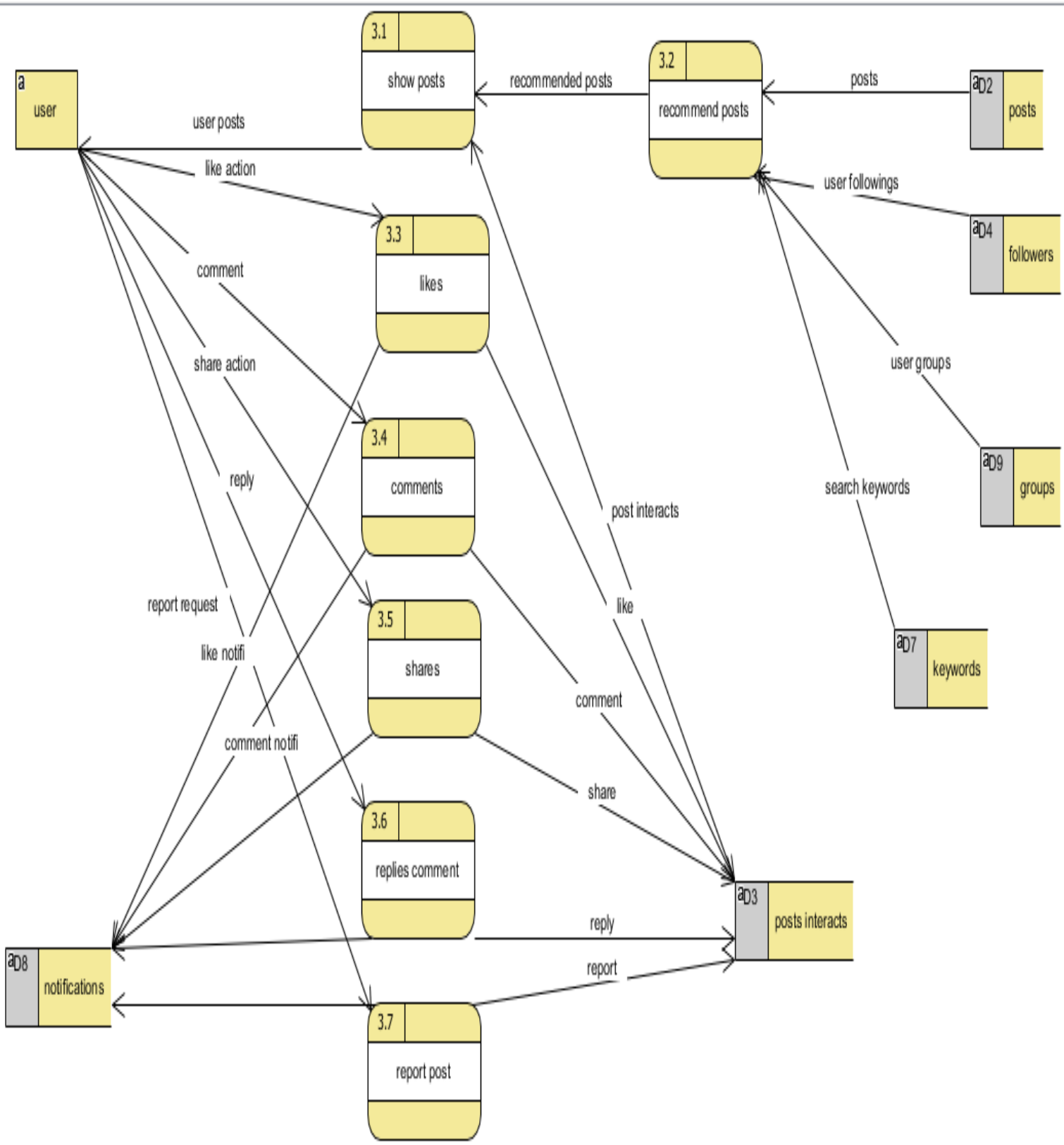


Figure11

Level (1) cont.:

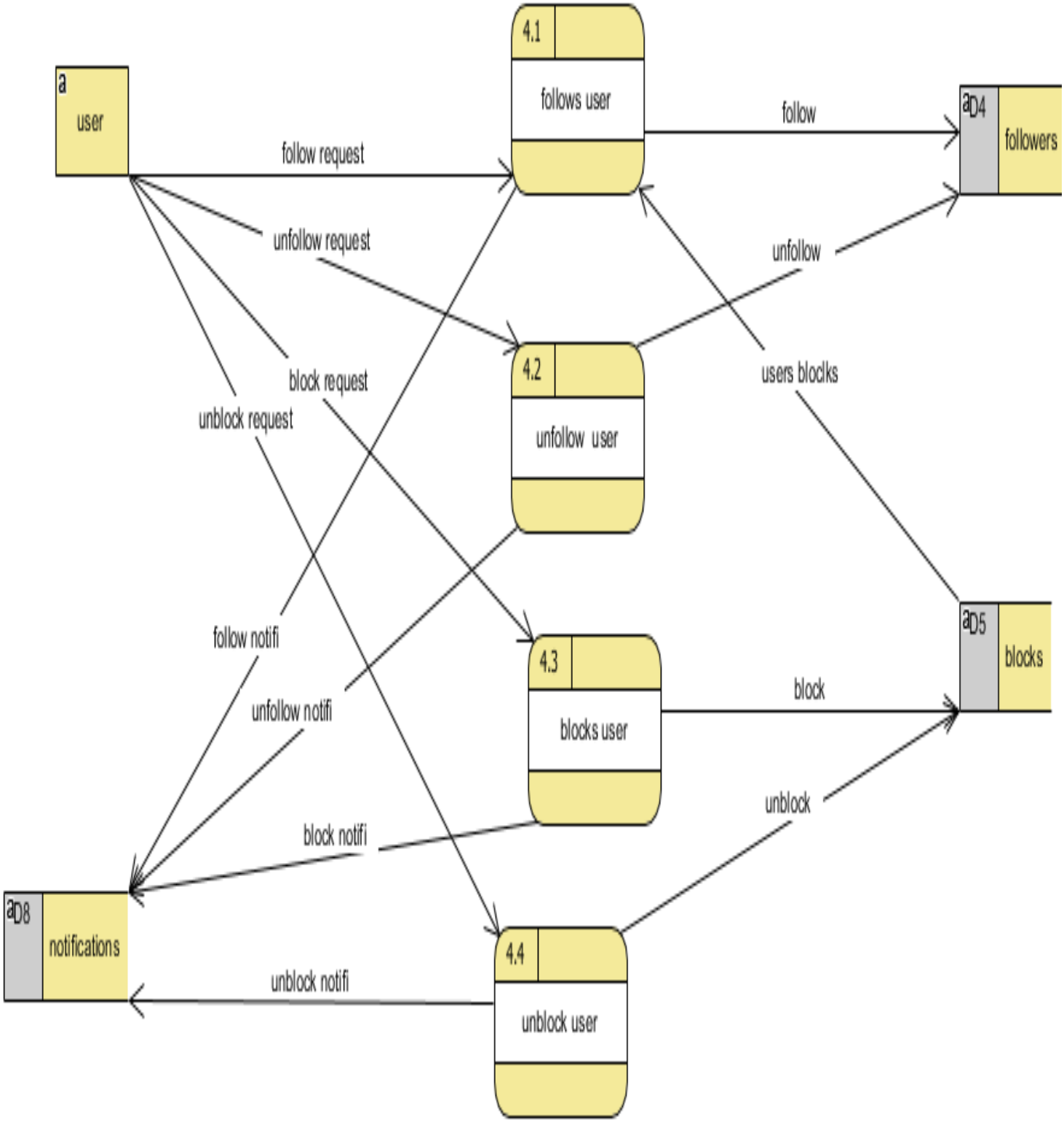


Figure 12

Level (1) cont.:

Figure 13

level(1)cont.

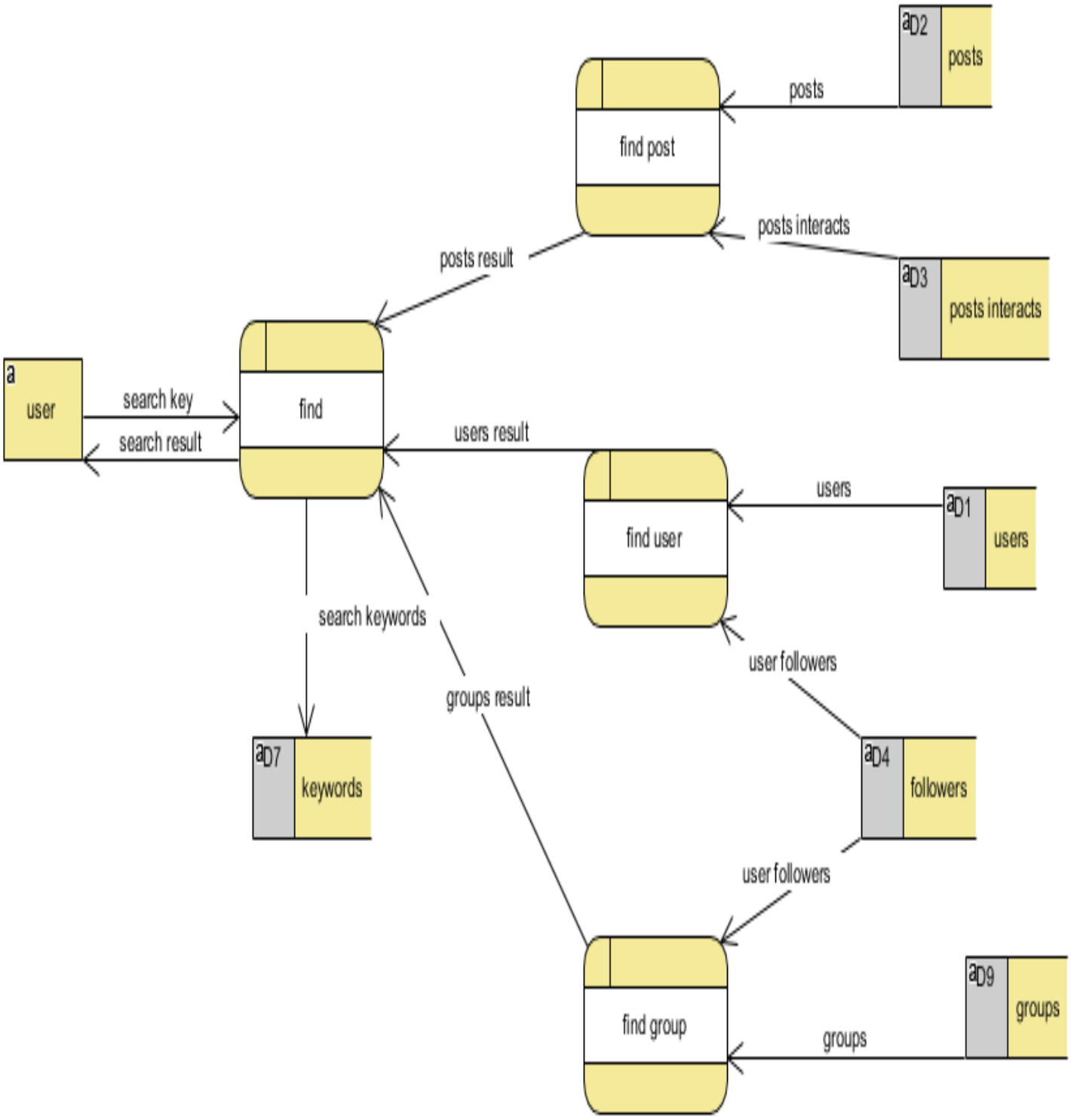


Figure 14

4 .4 Logical Database Diagram:

The logical Database Diagram shows entities and their relationships.

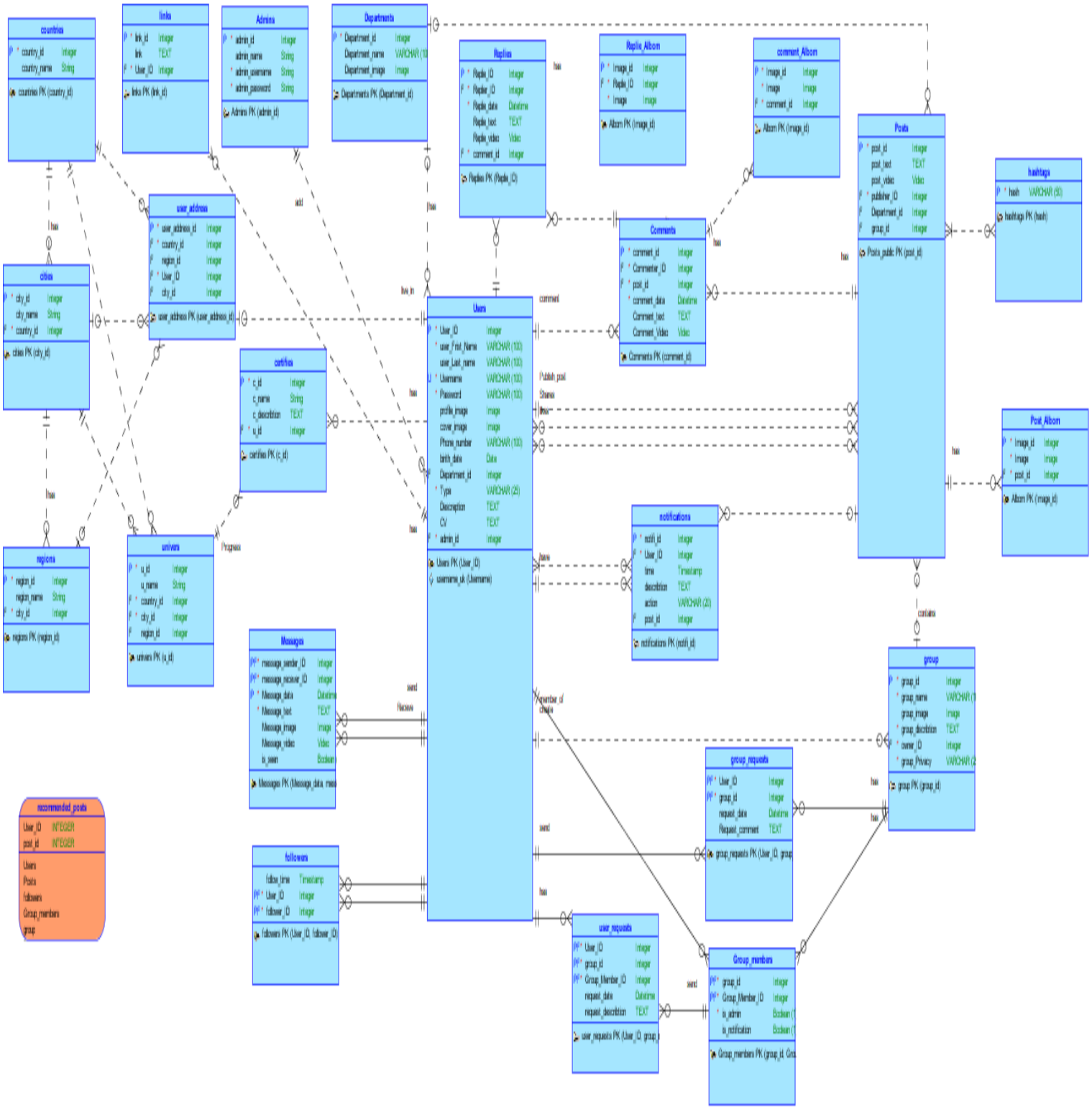


Figure 15

4.5Physical Database Diagram:

The picture shows the physical Database diagram that include information about sequence, index, primary and foreign keys between attributes.

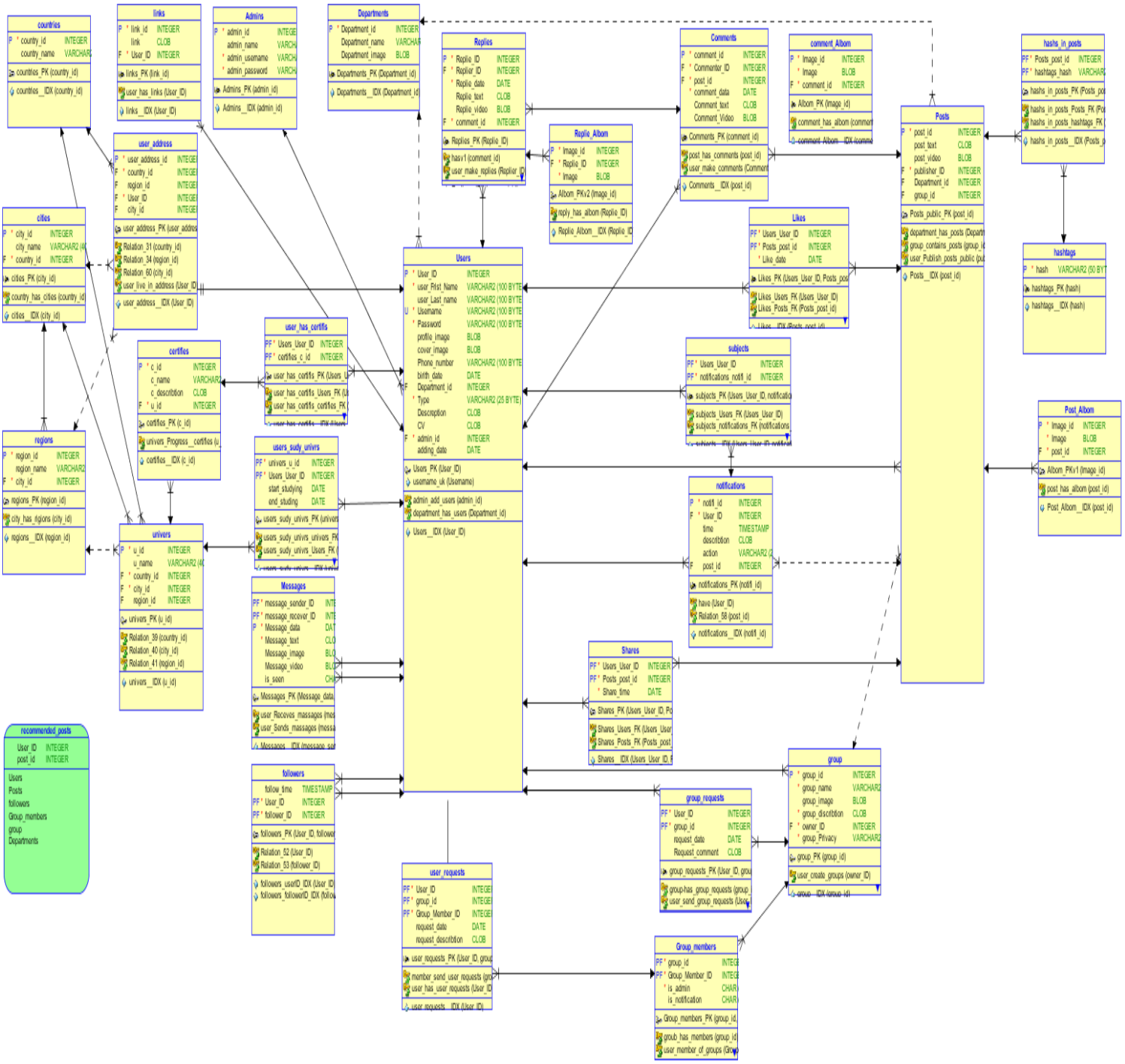


Figure 16

4.6 Create user and schema:

CREATE USER c## l cluster IDENTIFIED BY root DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp

QUOTA UNLIMITED ON users;

GRANT create session TO c##l cluster;

GRANT alter session TO c## l cluster;

GRANT create table TO c## l cluster;

GRANT create trigger TO c## l cluster;

GRANT create view TO c## l cluster;

GRANT create sequence TO c## l cluster;

GRANT create synonym TO c## l cluster;

GRANT create type TO c## l cluster;

GRANT create procedure TO c## l cluster;

Chapter 5

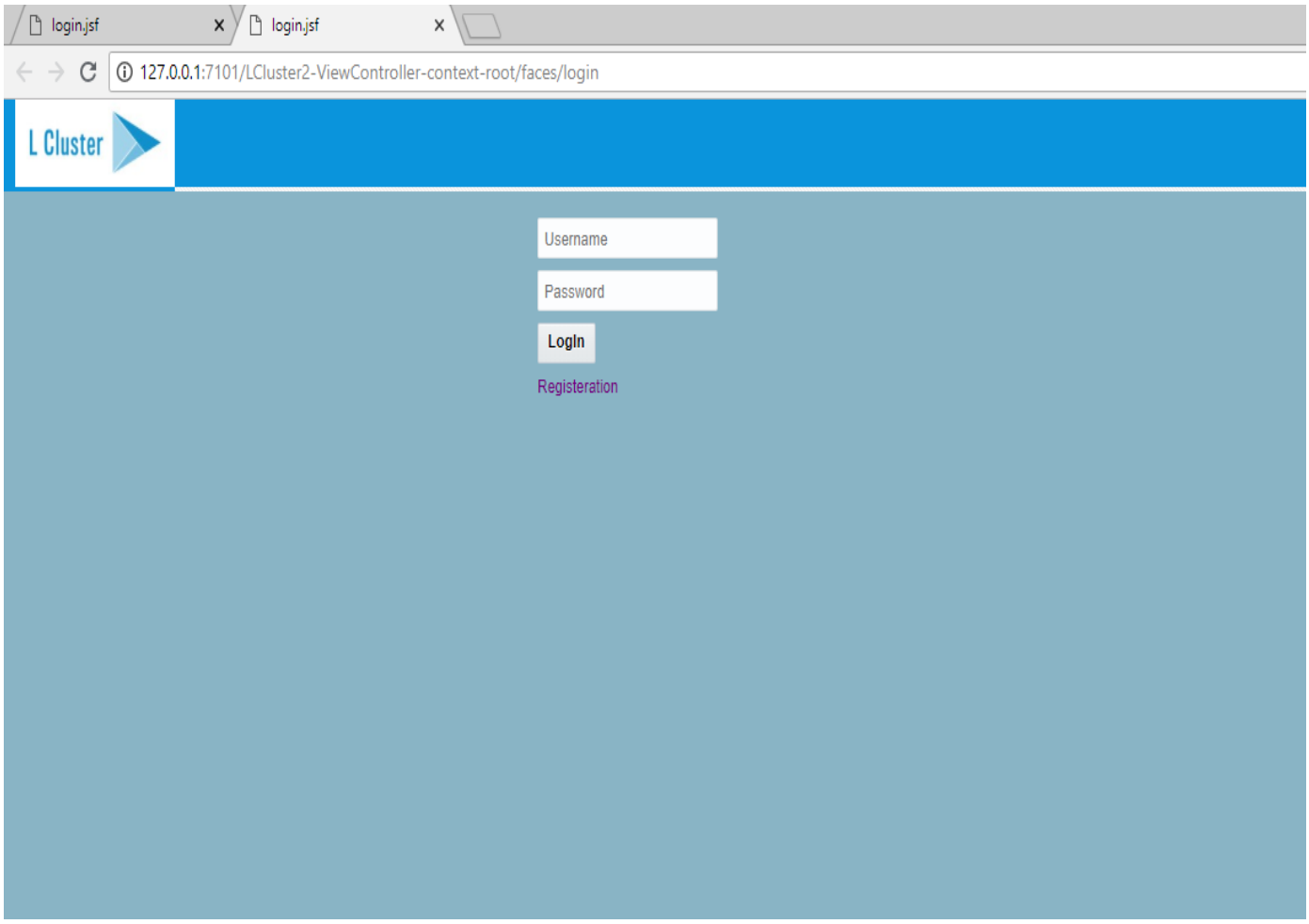
5. Project Tools:

- Oracle Database 12c.
- Oracle JDeveloper 12c.
- SQL Developer.
- SQL Developer Data Modeler.
- Visual Paradigm 15.0.
- Photoshop.

Chapter 6

6. Project design:

6.1 Login page:



The first page of the project web pages, one can login by entering his/her name and password.

Figure 17

6.2 Home Page:

The Home page of the user that include his picture and posts of his/her followers.

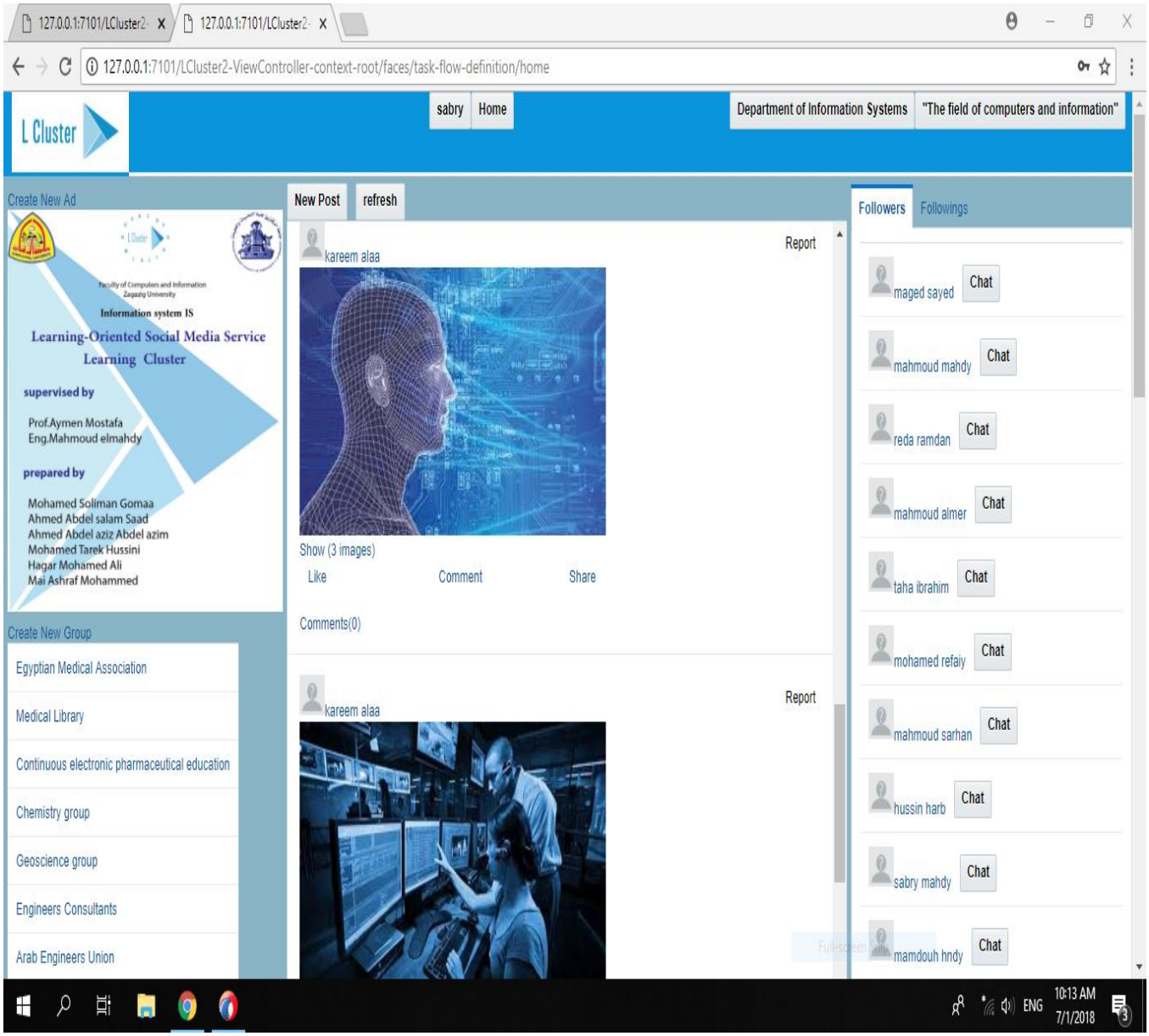


Figure 18

6.3 Creating posts:

User can create posts contain text, images and videos.

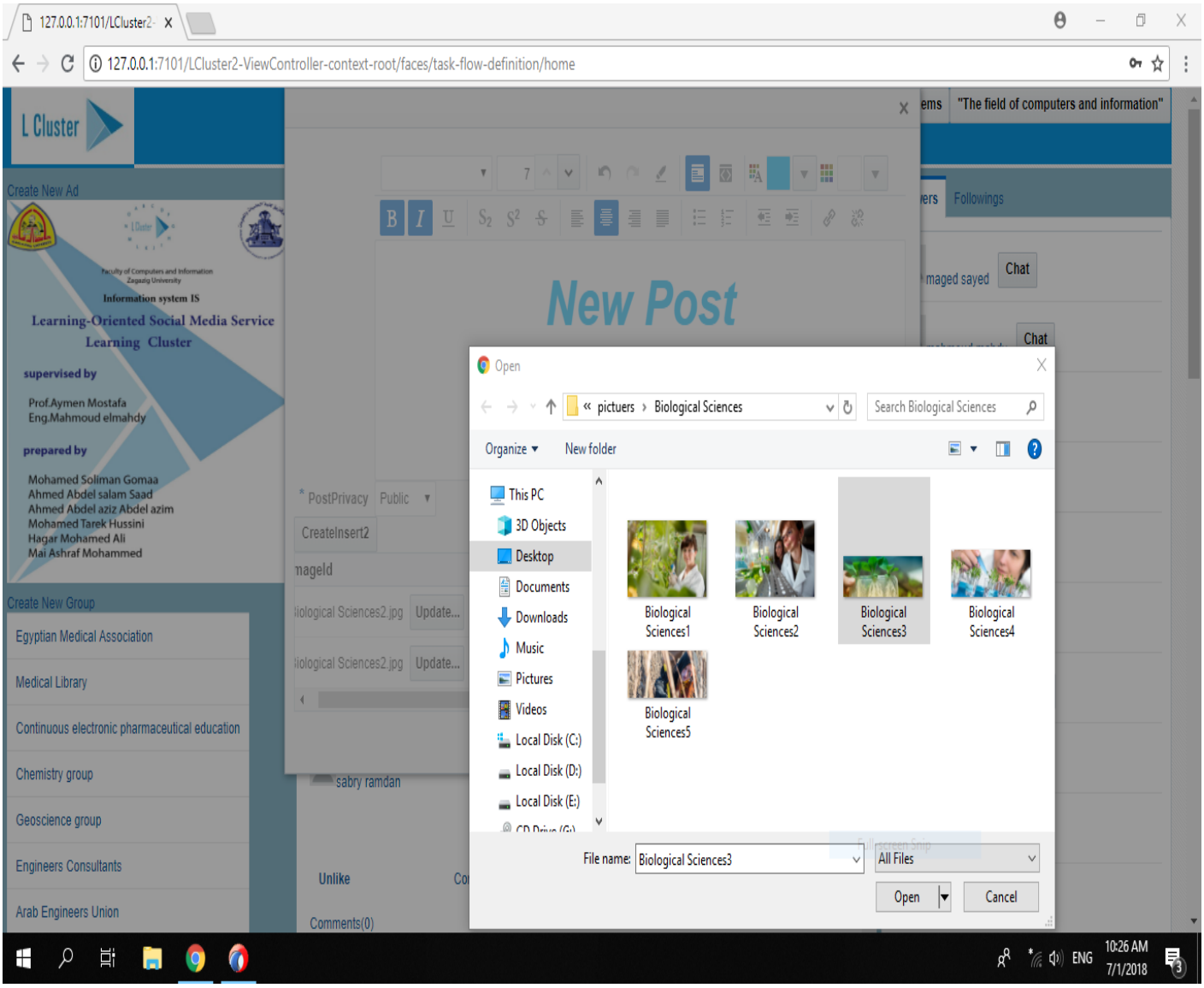


Figure 19

6.4 Publishing Posts:

User can publish posts that include images, text and videos.

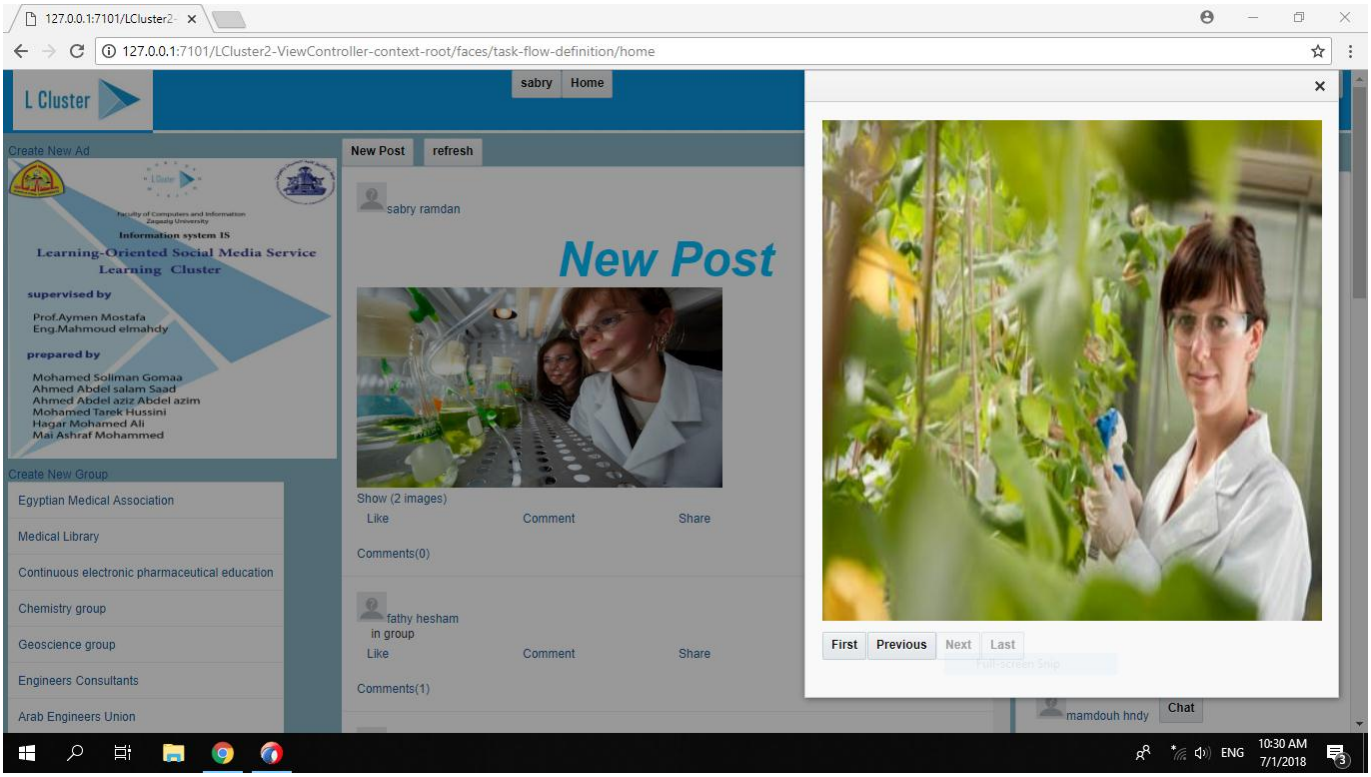


Figure 20

6.5 Writing Comments:

User can write comments and replies also with his/her followers.

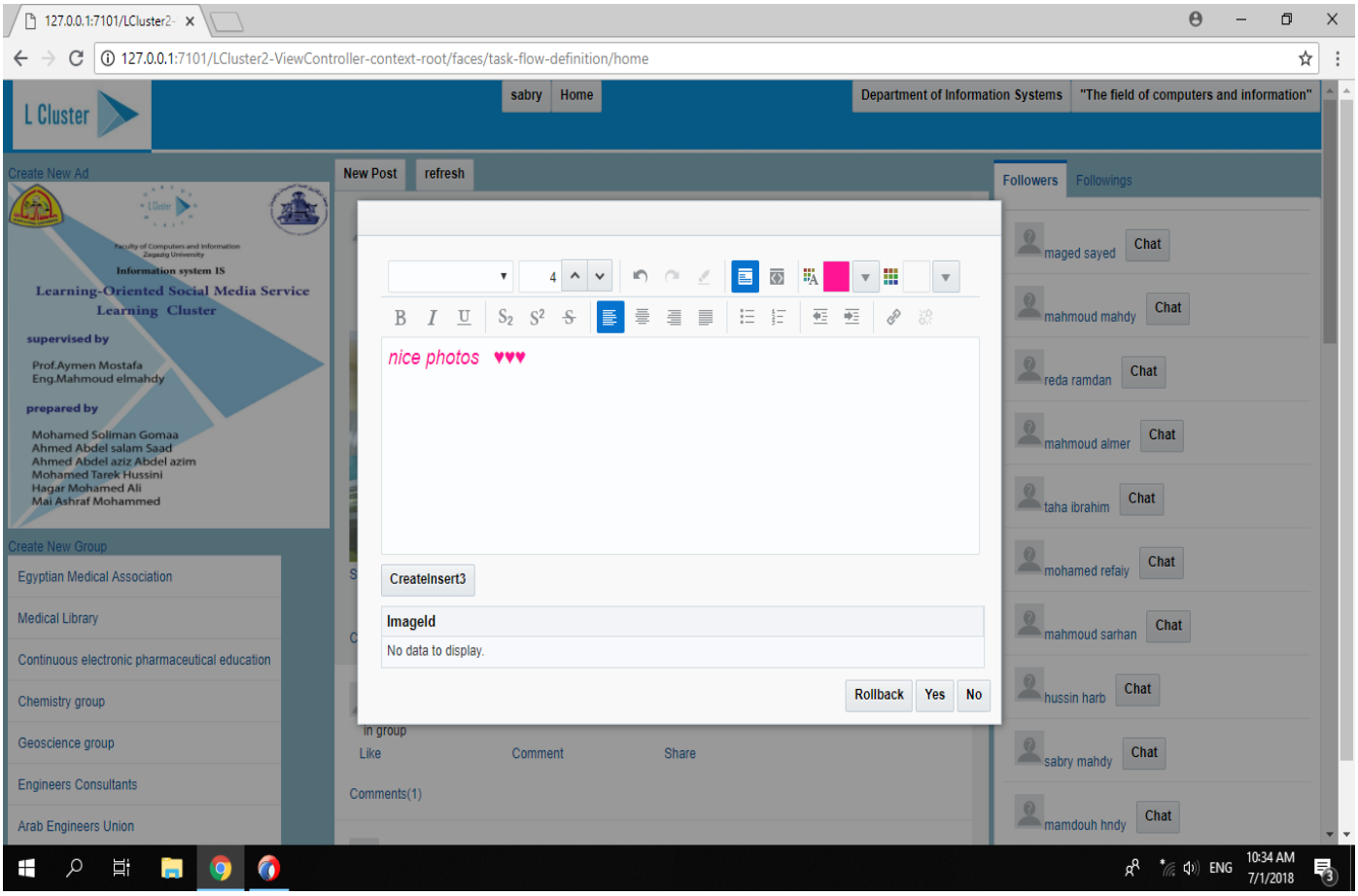


Figure 21

6.6 Publishing comments:

After creating comments, user can publish them and see other comments of followers. User can share his opinion.

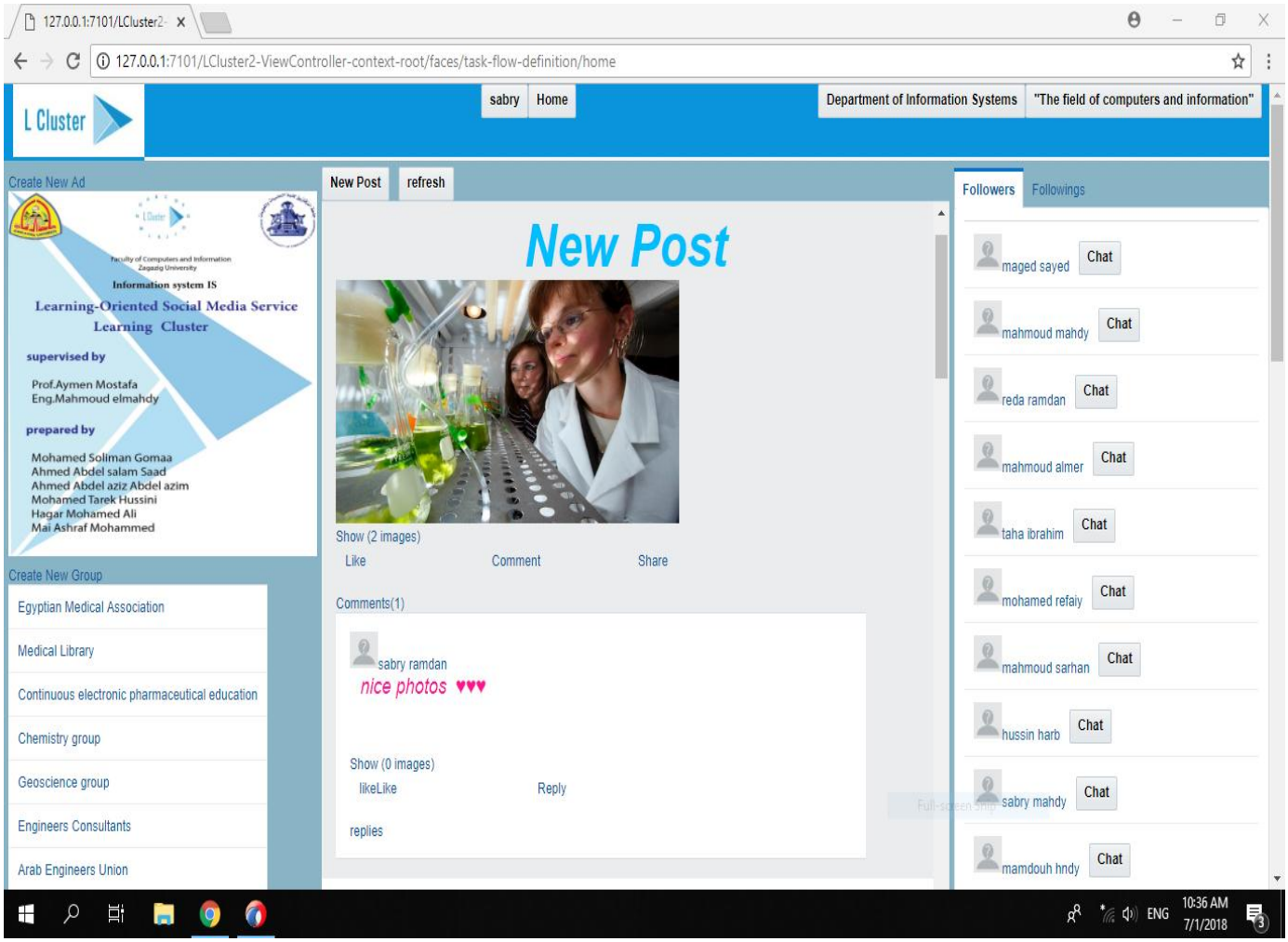
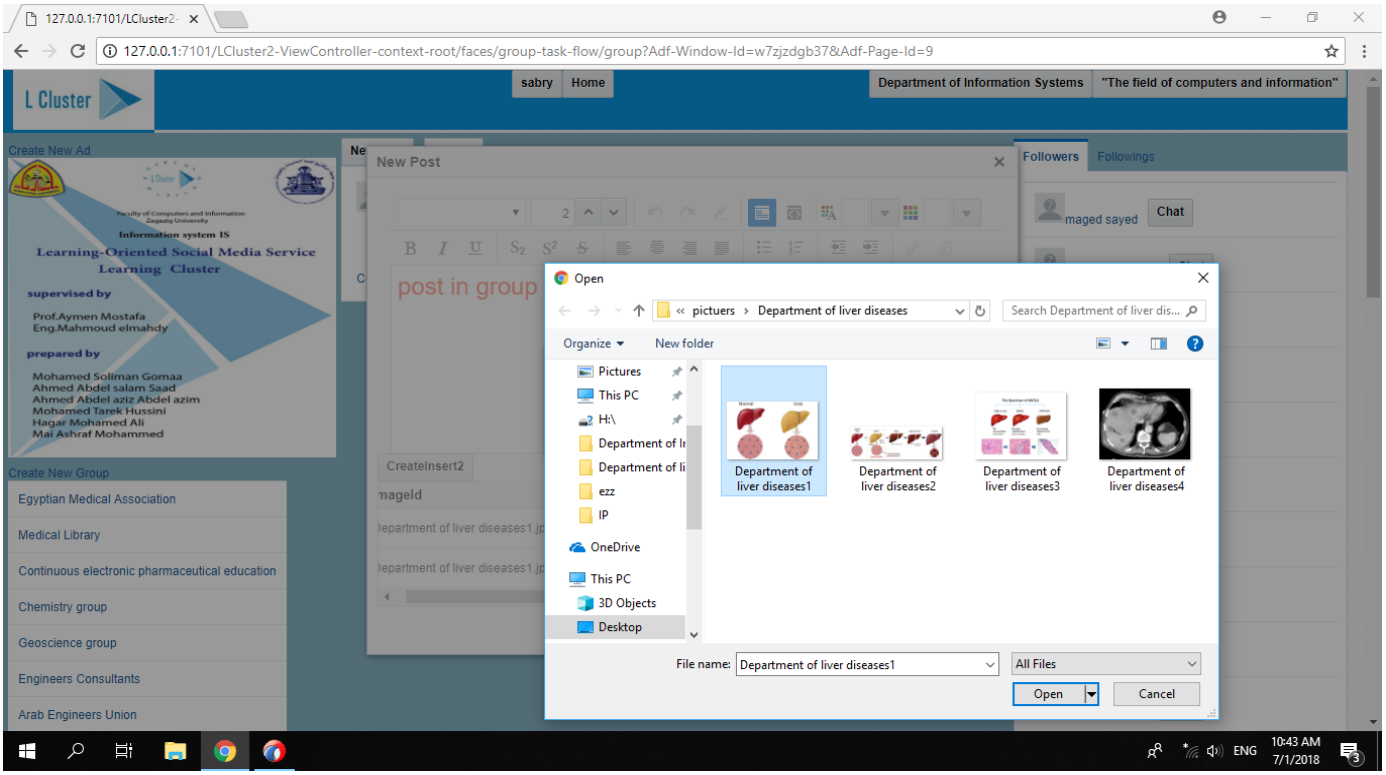


Figure 22

6.7 Creating and Publishing Posts in Groups:

User can write comment and publish it in groups.



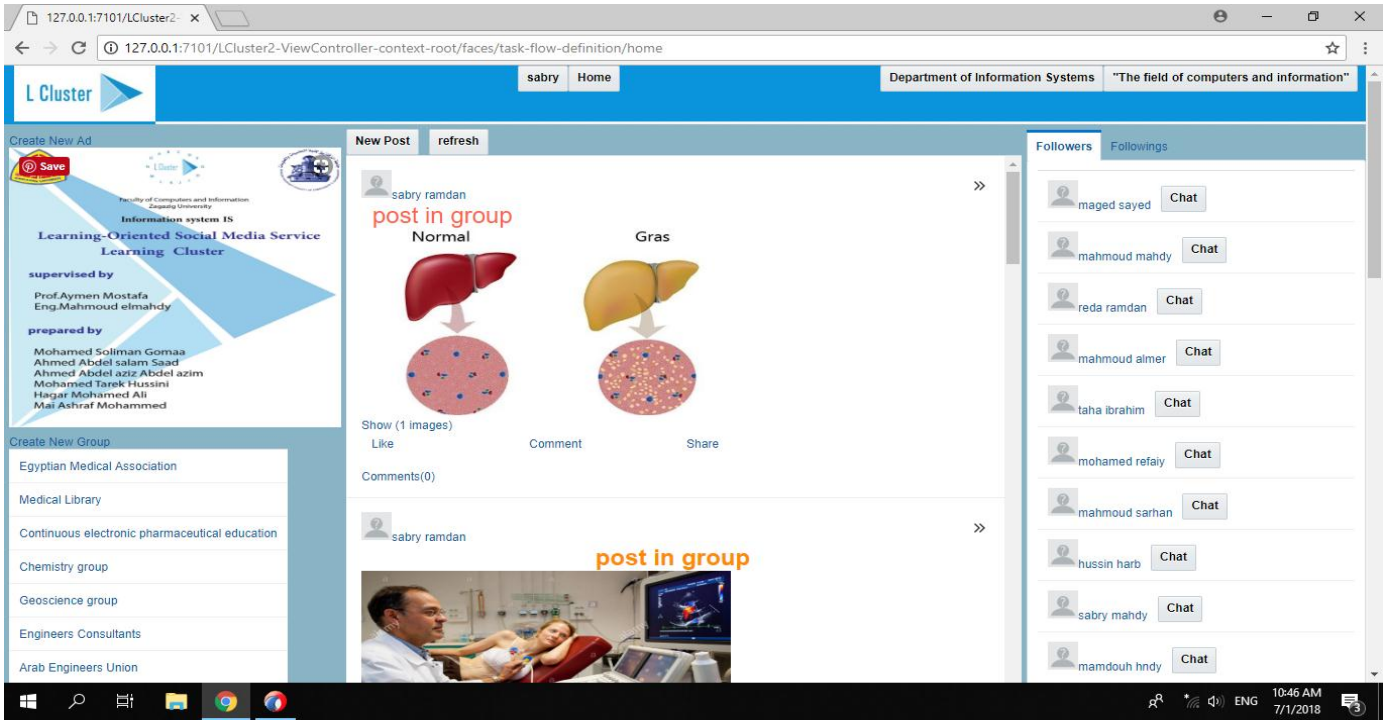


Figure 23

6.8 Editing and Deleting Posts:

User can also edit and delete posts that he/she published.

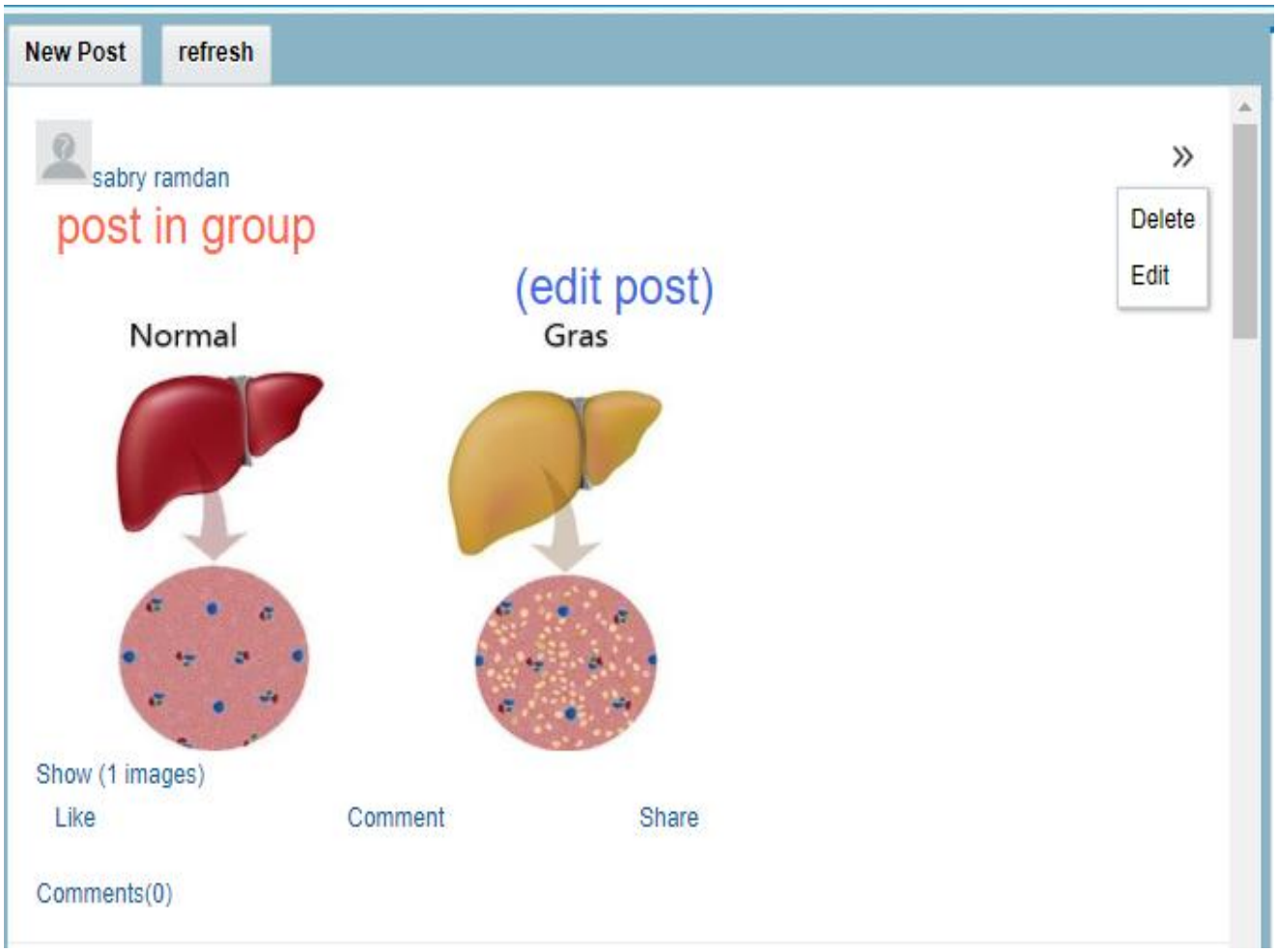


Figure 24

6.9 Chat:

User can also chat with his/her coworker, follower or peoples in his/her specialization.

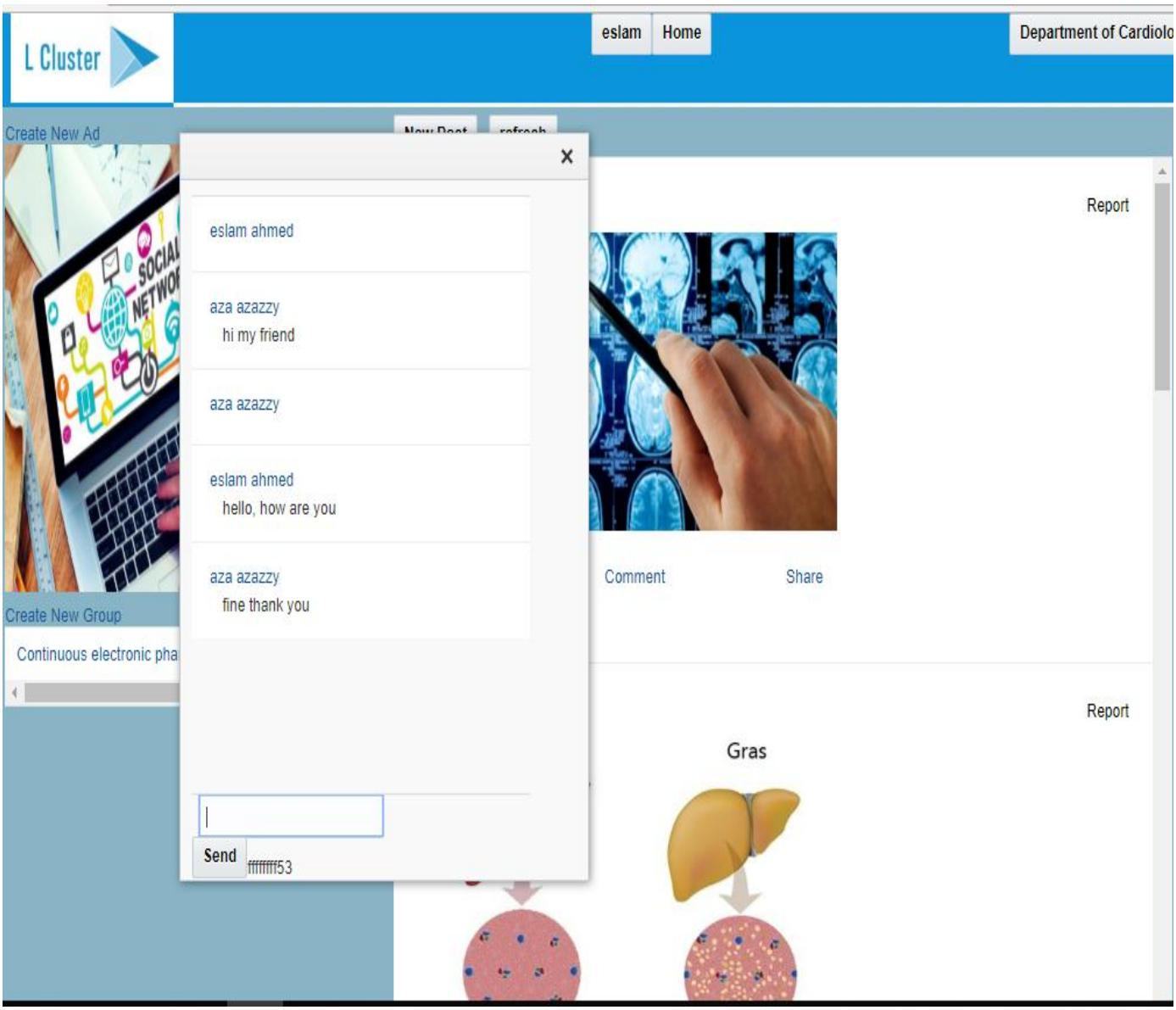


Figure 25

Chapter 7

7. Project implementation:

7.1 Context Flow:

The picture of context diagram between login page and home page.

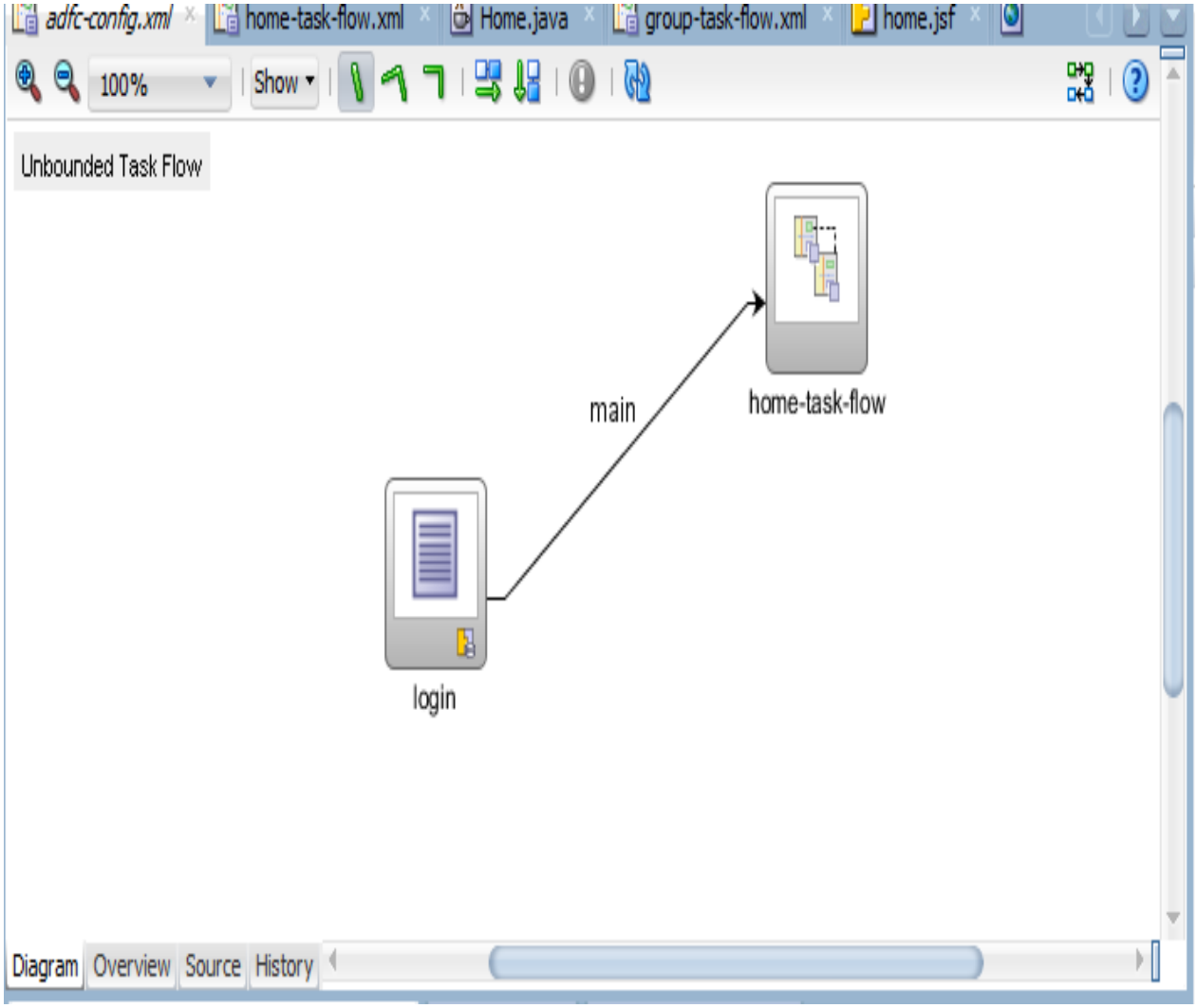


Figure 26

7.2 Context diagram of home page:

The picture shows pages of home page like group page and department page.

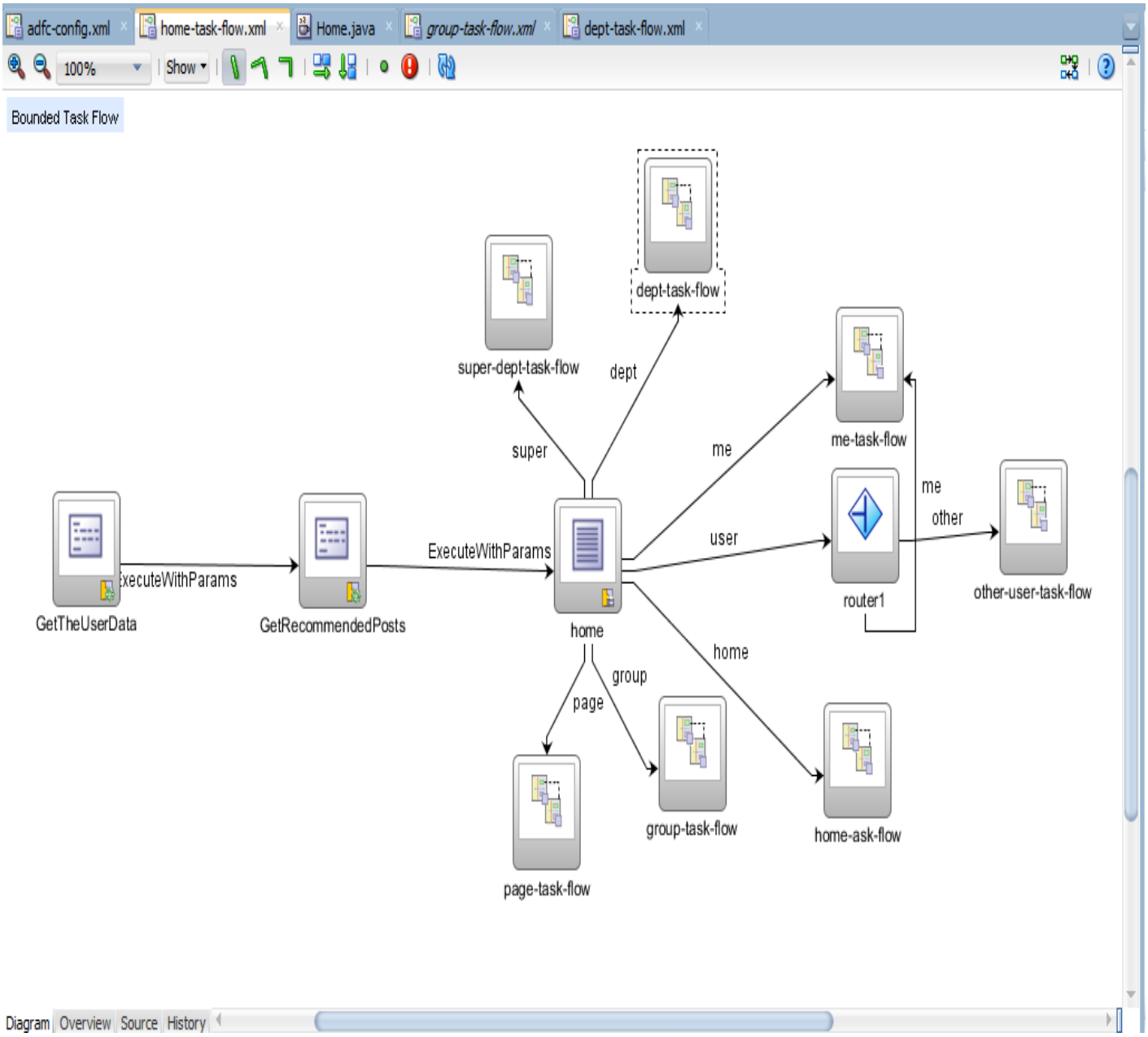


Figure 27

7.3 Context diagram of group page:

The picture shows page of user's groups.

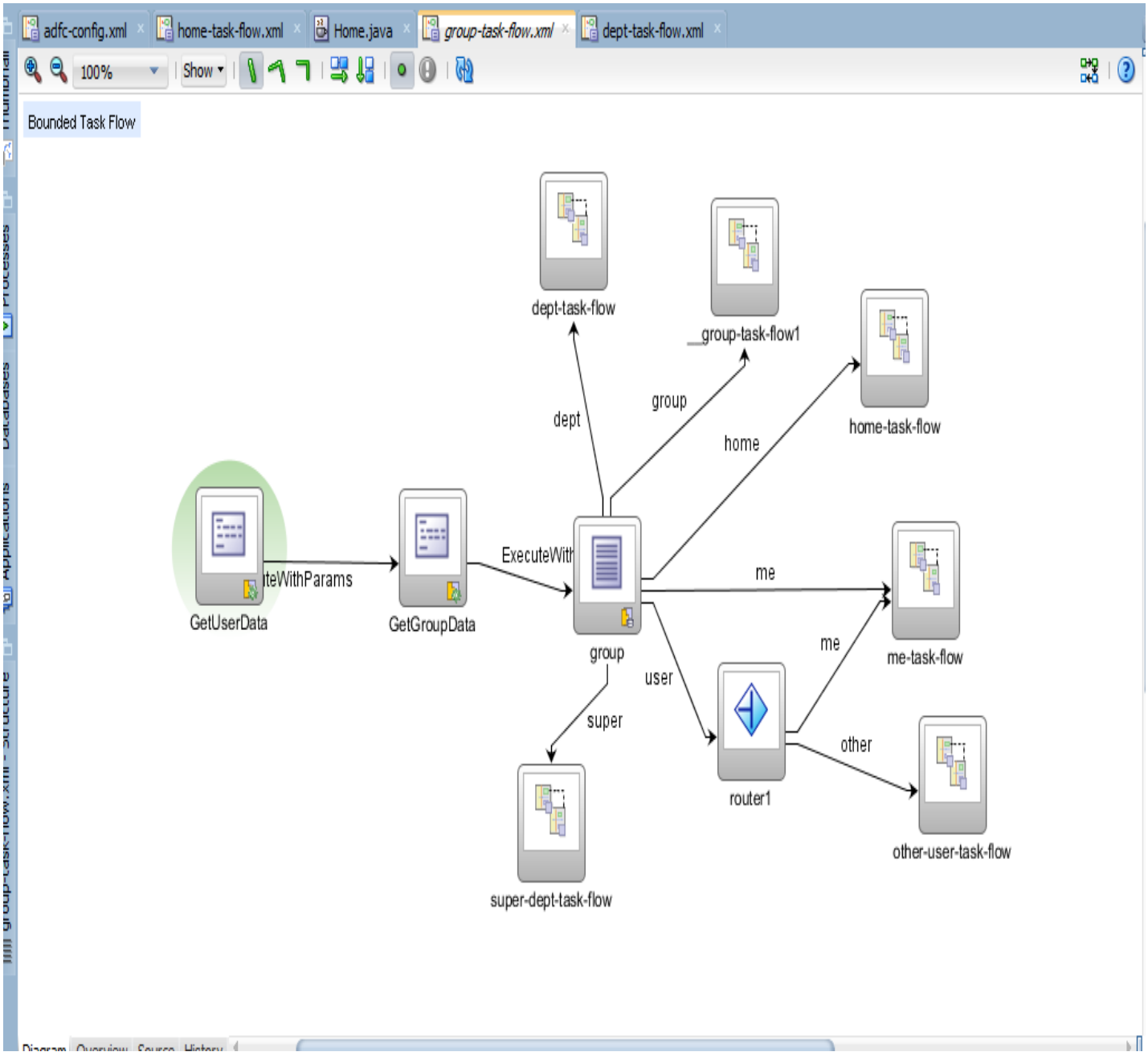


Figure 28

7.4 Context diagram of department page:

The picture shows department page. User should be included in one department not more.

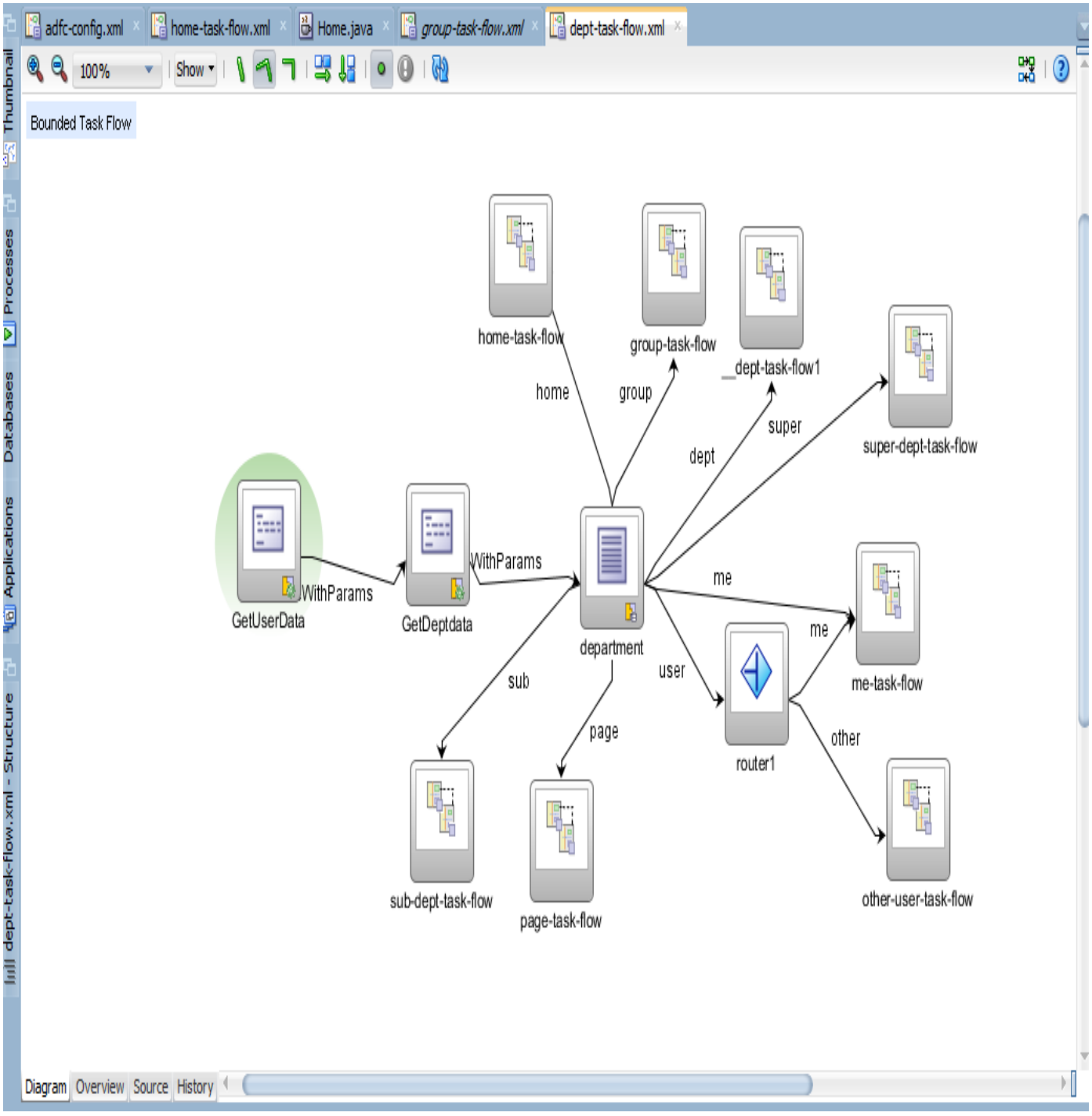


Figure 29

7.5 Showing images using servlet (code):

- **Using http servlet to get image (of blod data type) from database and show it ：**

```
Package view. Servlets;

import java.io.BufferedInputStream;

import java.io.IOException;

import java.io.OutputStream;

import java.io.PrintWriter;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.servlet.*;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.*;

import javax.sql.DataSource;
```

```
@WebServlet(name = "ImageServlet", urlPatterns = { "/imageservlet" })

public class ImageServlet extends HttpServlet {

private static final String CONTENT_TYPE = "image/gif; charset=utf-8";

public void init(ServletConfig config) throws ServletException { super.init(config);

  }

public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {


response.setContentType(CONTENT_TYPE);

String imageId = request.getParameter("id");

OutputStream os = response.getOutputStream();

    Connection conn = null;

    try {

       Context ctx = new InitialContext();

       DataSource ds= (DataSource) ctx.lookup("java:comp/env/jdbc/Lcluster3DS");

       conn = ds.getConnection();

       PreparedStatement statement =
```

```
        conn.prepareStatement("SELECT image_id,image " +
        "FROM images " + "WHERE image_id =?");

        statement.setInt(1, new Integer(imageId));

        ResultSet rs = statement.executeQuery();


        if (rs.next()) {

          Blob blob = rs.getBlob("Image");

          BufferedInputStream in = new
                  BufferedInputStream(blob.getBinaryStream());
int b;

byte[] buffer = new byte[10240];

while ((b = in.read(buffer, 0, 10240)) != -1) {
os.write(buffer, 0, b);

          }

os.close();

        }

    } catch (Exception e) {

    System.out.println(e);

      } finally {

        try {

          if (conn != null) {

            conn.close();
```

```
 }
} catch (SQLException sqle) {
System.out.println("SQLException error");
 }
```

7.6 Upload images (code):

- **The java class that used to upload image to blob data type on oracle data base :**

```
package view;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.SQLException;
import oracle.adf.model.BindingContext;
import oracle.adf.model.binding.DCBindingContainer;
import oracle.adf.model.binding.DCIteratorBinding;
import oracle.binding.BindingContainer;
import oracle.binding.OperationBinding;
import oracle.jbo.Row;
import oracle.jbo.domain.BlobDomain;
import org.apache.myfaces.trinidad.model.UploadedFile;
```

```
public class UploadPostImage {
public UploadPostImage() {
        super();
    }
public BindingContainer getBindings() {
  return
BindingContext.getCurrent().getCurrentBindingsEntry();
    }
private UploadedFile _file;
public void setFile(UploadedFile _file) {
        this._file = _file;
       // uploadImage();
       // BindingContainer bindings = getBindings();
     // OperationBinding operationBinding =
bindings.getOperationBinding("ExecuteWithParams");
      // Object result = operationBinding.execute();
        uploadImage();
           }
    public UploadedFile getFile() {
        return _file;
           }
```

```
public String uploadImage(){
UploadedFile myfile = (UploadedFile)this.getFile();
BindingContext bindingctx = BindingContext.getCurrent();
BindingContainer bindings = bindingctx.getCurrentBindingsEntry();
DCBindingContainer bindingsImpl = (DCBindingContainer)bindings;
DCIteratorBinding iter = bindingsImpl.findIteratorBinding("Images3Iterator");

Row row = iter.getCurrentRow();
row.setAttribute("Image", createBlobDomain(myfile));
return null;
}

private BlobDomain createBlobDomain(UploadedFile file) {
InputStream in = null;
BlobDomain blobDomain = null;
OutputStream out = null;
try {
in = file.getInputStream();
blobDomain = new BlobDomain();
```

```
out = blobDomain.getBinaryOutputStream();
  byte[] buffer = new byte[8192];
        int bytesRead = 0;
while ((bytesRead = in.read(buffer, 0, 8192)) != -1) {
    out.write(buffer, 0, bytesRead);
                }
in.close();
     } catch (IOException e)
e.printStackTrace();
    } catch (SQLException e) {
e.fillInStackTrace();
              }
return blobDomain;
            }
                                                                    }
```

7.7 Creating New Post (code):

- **The design of adf button to create new post by calling “newPost_b_action” java method**

```
<af:button text="New Post" id="newPost_b"
disabled="#{!bindings.CreateWithParams1.enabled}
"
action="#{viewScope.homeBean.newPost_b_action}
"/>
```

7.8 Showing New Post:

- **the “newPost_b_action” java method, that show “**NewPost_popup**” popup to get post data from yser anduse “CreateWithParams1” iterator to set data in database:**

```
public String newPost_b_action() {
    BindingContainer bindings = getBindings();
    OperationBinding operationBinding =
bindings.getOperationBinding("CreateWithParams1"
);
    Object result = operationBinding.execute();
    if (!operationBinding.getErrors().isEmpty()) {
      return null;
    }
```

```
        RichPopup.PopupHints hints = new
RichPopup.PopupHints().add(RichPopup.PopupHints
.HintTypes.HINT_ALIGN,RichPopup.PopupHints.Alig
nTypes.ALIGN_AFTER_END);


        getNewPost_popup().show(hints);

        return null;

    }
```

7.9 Get and Set data:

- **The “CreateWithParams1” iterator that used to get and set data in data base :**

```
<action IterBinding="Posts2Iterator"
id="CreateWithParams1"
InstanceName="AppModuleAMDataControl.Posts2"
          DataControl="AppModuleAMDataControl"
RequiresUpdateModel="true"
Action="createWithParams">
    <NamedData NDName="PublisherId"
NDValue="#{sessionBean.userID1}"
NDType="java.lang.String" NDOption="1"/>
    <NamedData NDName="PostPrivacy"
NDValue="public" NDType="java.lang.String"
NDOption="2"/>
  </action>
```

7.10 showing new post:

- **The design of “NewPost_popup” popup :**

```
<af:popup childCreation="deferred" autoCancel="disabled"
id="NewPostPop"

binding="#{viewScope.homeBean.newPost_popup}">
        <af:dialog id="d4" type="yesNo"
dialogListener="#{viewScope.homeBean.newPostDialog_listene
r}"

binding="#{viewScope.homeBean.newPost_dialog}">
          <af:panelFormLayout id="pfl7">
            <af:richTextEditor
value="#{bindings.PostText.inputValue}"

required="#{bindings.PostText.hints.mandatory}"

columns="#{bindings.PostText.hints.displayWidth}"

shortDesc="#{bindings.PostText.hints.tooltip}" id="it2"/>
```

```
                    <af:selectOneChoice
value="#{bindings.PostPrivacy1.inputValue}"
                                        label="#{bindings.PostPrivacy1.label}"

required="#{bindings.PostPrivacy1.hints.mandatory}"

shortDesc="#{bindings.PostPrivacy1.hints.tooltip}" id="soc1">
                      <f:selectItems
value="#{bindings.PostPrivacy1.items}" id="si1"/>
                      <f:validator
binding="#{bindings.PostPrivacy1.validator}"/>
                    </af:selectOneChoice>
                  </af:panelFormLayout>
                  <af:button
actionListener="#{bindings.CreateInsert2.execute}"
text="CreateInsert2"
                       disabled="#{!bindings.CreateInsert2.enabled}"
id="b21"/>
                  <af:table
value="#{bindings.Images3.collectionModel}" var="row"
rows="#{bindings.Images3.rangeSize}"
                       emptyText="#{bindings.Images3.viewable ?
'No data to display.' : 'Access Denied.'}"
                       rowBandingInterval="0"
fetchSize="#{bindings.Images3.rangeSize}" id="t2"
```

```
                    partialTriggers="::b21"
columnStretching="last" width="400">
                <af:column
headerText="#{bindings.Images3.hints.ImageId.label}" id="c2"
width="400">
                    <af:panelGroupLayout id="pgl13">
                        <af:inputFile
value="#{uloadPostImageBean.file}"

label="#{bindings.Images3.hints.Image.label}"

required="#{bindings.Images3.hints.Image.mandatory}"

columns="#{bindings.Images3.hints.Image.displayWidth}"

shortDesc="#{bindings.Images3.hints.Image.tooltip}" id="it14">
                            <f:validator
binding="#{row.bindings.Image.validator}"/>
                        </af:inputFile>
                        <af:button text="OK" id="b23"
action="#{uloadPostImageBean.uploadImage}"/>
                        <af:inputText
value="#{row.bindings.ImageId.inputValue}"

label="#{bindings.Images3.hints.ImageId.label}"
```

```
required="#{bindings.Images3.hints.ImageId.mandatory}"

columns="#{bindings.Images3.hints.ImageId.displayWidth}"

maximumLength="#{bindings.Images3.hints.ImageId.precision}
"

shortDesc="#{bindings.Images3.hints.ImageId.tooltip}"
id="it10">
                        <f:validator
binding="#{row.bindings.ImageId.validator}"/>
                        <af:convertNumber groupingUsed="false"

pattern="#{bindings.Images3.hints.ImageId.format}"/>
                      </af:inputText>
                    </af:panelGroupLayout>
                  </af:column>
                </af:table>
                <f:facet name="buttonBar"/>
              </af:dialog>
            </af:popup>
```

7.11 Making Buttons of (Yes= commit) and (No= rollback) on:

- **We customized popup by “newPostDialog_listener” java method**

```
public void newPostDialog_listener(DialogEvent dialogEvent)
{
    // Add event code here...
    DialogEvent.Outcome outcome =
dialogEvent.getOutcome();
    if(outcome == DialogEvent.Outcome.yes)
    {
      BindingContainer bindings = getBindings();
     OperationBinding operationBinding=
bindings.getOperationBinding("Commit");
        Object result = operationBinding.execute();
      }
     if(outcome == DialogEvent.Outcome.no)
    {
      BindingContainer bindings = getBindings();
```

```
        OperationBinding operationBinding=
bindings.getOperationBinding("Rollback");
            Object result = operationBinding.execute(); }  }
```

7.12 Creating Database view:

- **Create database view that used to get posts recommended for user included in home page**

```
CREATE OR REPLACE VIEW recommended_posts  AS
SELECT
   posts.post_id,
   users.user_id,
   posts.post_date
FROM
   users
   INNER JOIN posts ON users.user_id = posts.publisher_id
UNION
SELECT
   posts.post_id,
   users.user_id,
   posts.post_date
FROM
   users
```

```
    INNER JOIN followers ON users.user_id =
followers.follower_id
    INNER JOIN users users1 ON users1.user_id =
followers.user_id
    INNER JOIN posts ON users1.user_id = posts.publisher_id
UNION
SELECT
    posts.post_id,
    users.user_id,
    posts.post_date
FROM
    users
    INNER JOIN group_members ON users.user_id =
group_members.group_member_id
    INNER JOIN "group" ON "group".group_id =
group_members.group_id
    INNER JOIN posts ON "group".group_id = posts.group_id
UNION
SELECT
    posts.post_id,
    users.user_id,
    posts.post_date
```

```
FROM
    users
    INNER JOIN departments ON departments.department_id =
users.department_id
    INNER JOIN posts ON departments.department_id =
posts.department_id
;
```

Chapter 8

8. Business plan:

- Customer segments
- value propositions
- channels
- Customer relationships
- Revenue streams
- Key resources
- Key activities
- Key partners
- Cost structure

8.1 Customer segments:

We are creating L Cluster for some departments such as medicine, engineering, computer and information science and science,

In the development of the project; all people and all organization in the world.

8.2 Value propositions:

- Easy communication between people and many organizations.
- Publish and display scientific researches.
- Special page for each specialization for easy search and access to the target.
- Provident most of scientific books.
- Display all scholarships for coworkers.
- Display all special news.
- Provides teamwork.
- Displays the conference online.
- Saving time and power.

8.3 Channels:

- Doctors , Engineers and clustered organization can communication by :

- www.l clusterd.com

8.4 Customer relationships:

- Coworkers and friends in career can communicate with each other and by communities or automated services.

8.5 Revenue streams:

- L Cluster is very important to people around the world so we don't think about money.
- L Cluster is free to all coworkers, Common departments and organizations, it will not be limited for some fields in the future.

8.6 Key resources:

- **L Cluster resources are :**

1) Database: Oracle database 12c (sql-plsql)
2) Web design: (html5-css3-jqurey-bootstrap)
3) Developer :Oracle ADF
4) Cloud to save data

8.7 Key activities:

- **L Cluster activities are :**

1) Web designer
2) Web development
3) Database administrator

8.8 Key partners

- **L Cluster has relationship with :**

1) All specialists
2) All Departments
3) All syndicates
4) All faculties and all expertise.

8.9 Cost structure

- **All cost will be in :**

1) Develop the site
2) Save data in cloud

Chapter 9

9. Conclusion:

9.1 In The Future:

We can develop this project by:

1. **Improving its design**
   - We can develop its shape and pages.
   - We can develop the buttons.
   - We can Using new tools and functions

2. **Improving its security**
   Improve security to avoid hacking and preventing it, security is very important for sensitive data.
3. **Using Gaps system:**
   - Which is a network of orbiting satellites in the space that sends precise and accurate position?
   - Gaps has the ability to pinpoint locations up to three dimensions like the latitude, longitude, and altitude.
   - Global positioning system enables us to determine the positions easily.

4. **Using Data mining algorithms :**
   - Is a set of heuristics and calculations that creates a data mining model from data?
   - It can be a challenge to choose the appropriate or best suited algorithm to apply to solve a certain problem.

- Data mining algorithms **using** negative and positive feedback.

9.2 Negative and positive feedback:

Means:

If comments containing good or bad words , images and videos this can determine if posts are good or bad using data mining algorithms.