**Zola**

**Development’s Guide**

**Faculty: Ho Hoang Kiem**

**Student:**

|  |  |  |
| --- | --- | --- |
| **No** | **Full Name** | **Roll No** |
| **1** | **Duong Anh Tuan** | **SE62310** |

**Table of contents**

Table of contents 2

**Review 1** 3

* 1. [Introduction](#_Introduction) 4
  2. [Existing Scenario](#_Existing_Scenario) 4
  3. [Customer Requirement Specification](#_Existing_Scenario) 4
  4. [Functional Requirement Specification](#_Functional_Requirement_Specificatio) 4
  5. [System Requirements](#_System_Requirements_1) 7
  6. [Development Software](#_Development_Software) 7

[Technology](#_Techonology) 7

**Review 2** 9

* 1. [Use Case and Sequence Diagram](#_Use_Case_and) .10
  2. Entity Relationship Diagram (ERD) xx

**Review 3** xx

* 1. Database Relationship Diagram xx
  2. Table Definition xx
  3. Other Constraint [xx](#_Table_Definition_1)
  4. Site map xx
  5. GUI Design xx
  6. Report xx

**Task sheet** xx

Task sheet xx

**Monitoring Report** xx

Unit Test xx

REVIEW 1

# Introduction

Zola is an free chat application in windows. It allow you to communicate with your friends.

# Existing Scenario

# Customer Requirement Specification

# Functional Requirement Specification

## Guest can register

### Description

The application allows guest create their account with the following information:

* Username
* Password
* Name
* Gender

Username must satisfy:

* Contain only lower case and number characters.
* If user input upper cases in username, system automatically change them to lower case characters.

Password must satisfy:

* Upper case and lower case characters
* Number characters
* Special characters such as: $#!
* Password length between 6-20 chars

Genders are:

* Male
* Female

Username, password, name, gender are required.

### Functional Requirements`

1. User must provide Username, Password, Name, Birthday, Gender to create an account.
2. Username is unique.

## Guest can login

### Description

The application allows guest login into their account with the following information:

* Username
* Password

Username must satisfy:

* Contain only lower case and number characters.
* If user input upper cases in username, system automatically change them to lower case characters.

Password must satisfy:

* Upper case and lower case characters
* Number characters
* Special characters such as: $#!
* Password length between 6-20 chars

Username, password are required.

### Functional Requirements

1. User must provide Username, Password to login.

## User can update their account

### Description

The application provide a window allow user to change following account’s information:

* Name
* Password
* Birthdate
* Gender
* Avatar

Password must satisfy:

* Upper case and lower case characters
* Number characters
* Special characters such as: $#!
* Password length between 6-20 chars

### Functional Requirements

1. User can update thei own account’s information.

## User can search orther accounts

### Description

The application provide a textbox allow user inputs orther account’s username. After receiving search result, application must show top 20 results, which is not friend of user’s account into a list, each item contains:

* Name
* Username
* Avatar

### Functional Requirements

1. User can search orther accounts by account’s username.

## User can make friend with each orther.

### Description

After searching accounts, application list top 20 results and check if user and that account is already friend.

* If not, show button “Make Friend” in the list searching result.
* If user sent “Make friend” invitation, but orther side account haven’t accepted yet, show label “Pending”.

### Functional Requirements

1. User can make friend with orther accounts after searching accounts.

## User can receive “make friend” invitation.

### Description

When orther account sends “make friend” request, application show 2 buttons:

* Accept.
* Not Accept.

When User press “Accept” Button, friend request disappears, user and order account’s relationship become friend. Then notify requested order account message: “Request Accepted”.

When User press “Not Accept” Button, friend request disappears. Then notify requested account message: “Request NOT Accepted”.

### Functional Requirements

1. User can respone “Make friend” request.
2. Notify result “make friend” request for requested account.

## User can chat with friends .

### Description

Application shows a list containts Friend’s account. Each item containts:

* Name
* Avatar
* Status

When user clicks an item, application will show a Chat area containts:

* Message of two accounts sent.
* Send Message button.
* Input Message Textbox.

### Functional Requirements

1. User can see friend list.
2. User can chat with friends.

## User can logout.

### Description

Application proovides a “logout” Button.

When user clicks on that button, application redirect into login GUI.

### Functional Requirements

1. User can logout.

# System Requirements

## Hardware

### Host Server

|  |  |
| --- | --- |
| Processor | Core i5-4200H or higher. |
| Memory | 12GB RAM or greater. |
| Modem/ADSL | Connecting to internet 24/24 |

### Client

|  |  |
| --- | --- |
| Processor | Pentium III or higher. |
| Memory | 512MB RAM or greater. |
| Monitor | Super VGA (800x600) or higher resolution with 256 colors. |
| Pointing Device | Microsoft Mouse, Microsoft IntelliMouse®, or compatible pointing device. |
| Modem/ADSL | Connecting to internet |

## Software

### Server

|  |  |
| --- | --- |
| Operation System | Windows 10. |
| Database | Microsoft SQL Server 2017 |
| Software | .NET Framwork 4.6.1 |

### Client

|  |  |
| --- | --- |
| Operation System | Windows 10. |
| Software | .NET Framwork 4.6.1 |

# Development Software

* Microsoft Windows 10 Education
* Visual studio 2017 Community.
* .NET Framwork 4.6.1
* Microsoft SQL Server 2017

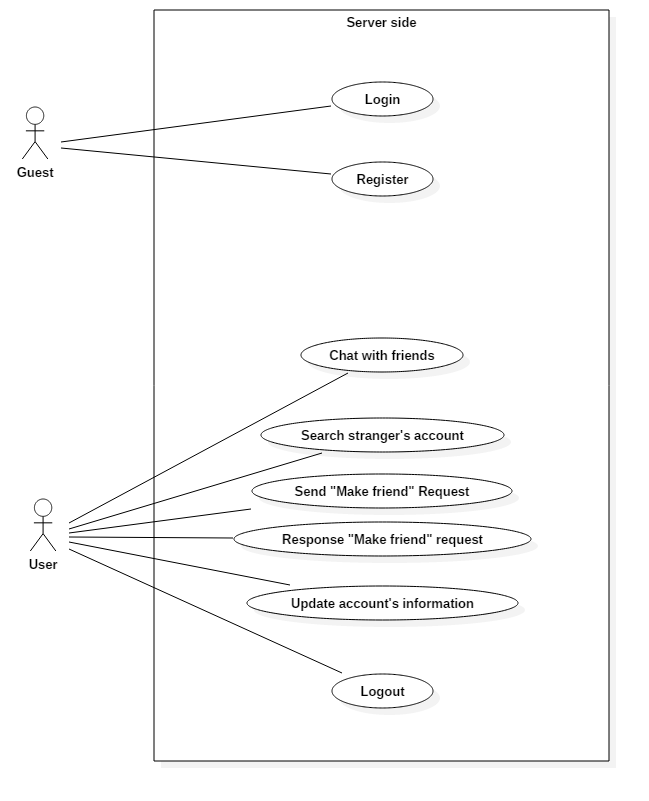
# Technology

* WCF
* WPF

|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader  **Duong Anh Tuan** | **Ho Hoang Kiem** |
| **Date** | **12/10/2017** |  |

REVIEW 2

# Use Case Diagram.

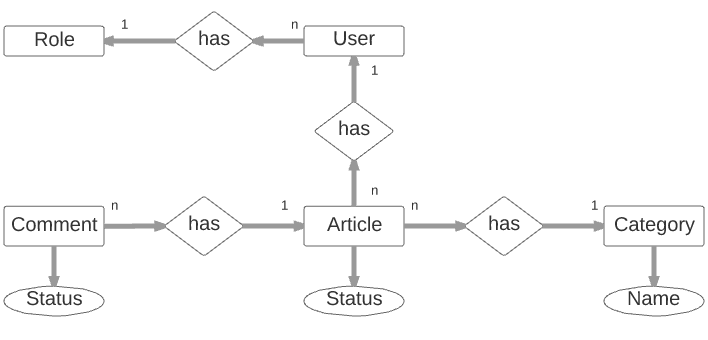


# Entity Relationship Diagram (ERD)

## Conceptual

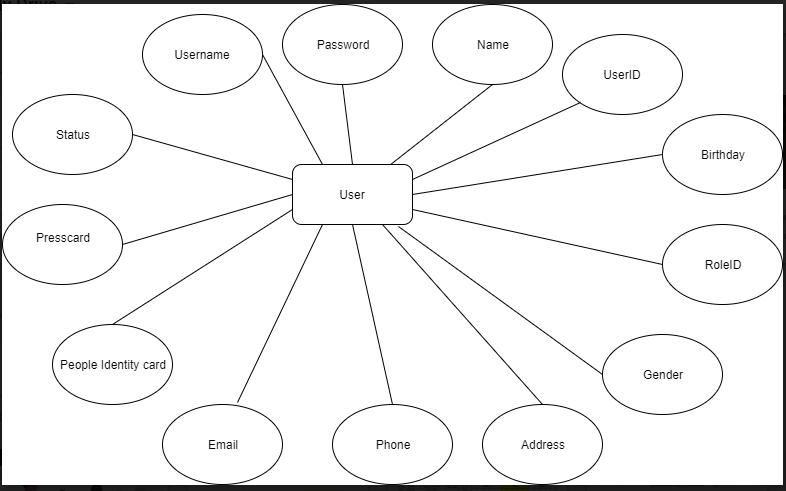
## 

## 9.2. Logical



***9.3. Table***

**9.3.1. User**

****

REVIEW 3

# 10. Database Relationship Diagram

# 22140183_1876889965670709_1219535866_o

# 11. Table Definition

# *11.1. User*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| PK | **UserID** | int |  | No |  |  | User’s ID in database |
| Unique | Username | varchar | 45 | No |  |  | Username |
|  | Password | varchar | 200 | No |  |  | Password |
|  | Name | nvarchar | 45 | No |  |  | Name of the user |
|  | Birthday | date |  | No |  |  | Birthday of the user |
|  | Gender | int |  | No |  |  | Gender of the user:  - 0 for Female  - 1 for Male  - 2 for Others |
|  | Address | nvarchar | 200 | No |  |  | Address of the user |
|  | Phone | varchar | 20 | No |  |  | Phone of the user |
|  | Email | varchar | 45 | No |  |  | E-mail of the user |
|  | PeopleIdentityCard | varchar | 45 | No |  |  | People Identity Card Number of the user |
|  | PressCard | varchar | 45 | Yes |  |  | Press Card number of the user |
|  | RoleID | int |  | No | Role’s ID of “Reader” | Role.RoleID | Role’s ID of user’s role |
|  | Status | varchar | 45 | No |  |  | Status of the user’s account:  - Available  - Unavailable |

**Constraint**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| User | +  - Role ID cannot be RoleID of “Admin” (the system has only 1 administrator)  - Email must be in e-mail’s form ([something@something](mailto:something@something))  - Phone must be string of numeric and have length at least 9 | - | +  - Cannot update UserID, Username |

# *11.3. Comment*

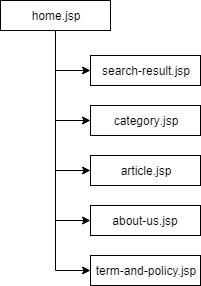
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| PK | **CommentID** | int |  | No |  |  | Comment’s ID in database |
|  | ArticleID | int |  | No |  | Article.ArticleID | Article’s ID of the article contains comment |
|  | UserID | int |  | No |  | User.UserID | User’s ID of comment’s creator |
|  | CreatedTime | datetime |  | No |  |  | Time the comment created |
|  | Content | nvarchar | 200 | No |  |  | Category’s ID of article’s category |
|  | Status | varchar | 45 | No |  |  | Comment’s status :  - show  - hide |
|  | LastModifedUserID | int |  | No |  | User.UserID | User’s ID of the user last modifying comment |
|  | LastModifiedTime | datetime |  | No |  |  | Time of last modifying comment |

**Constraint**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Add** | **Delete** | **Update** |
| Comment | + | - | +  - Cannot update  CommentID, ArticleID, CreatorID, CreatedTime |

# 12. Site map

## Common Side

****

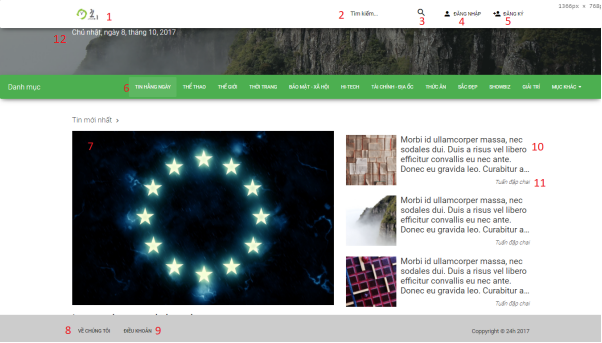
# GUI Design



## Common side

### Home page:

This is Home page of the website, the users. readers, journalist, editors will go to this page first.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Img24hLogo | Image |  | Click | Redirect to Home page |  |
| 2 | txtSearch | Text field | 0<length<50 |  | Search article by title |  |
| 3 | btnSearch | Button |  | Click | Redirect to Search result page |  |
| 4 | btnLogin | Button |  | Click | Show Login pop-up |  |
| 5 | btnRegister | Button |  | Click | Show Register pop-up |  |
| 6 | btnCategory | Button |  | Click | Redirect to Category page |  |
| 7 | imgPreviewArticle | Image |  | Click | Redirect to Article page |  |
| 8 | btnAboutUs | Button |  | Click | Redirect to About Us Page |  |
| 9 | btnTermAndPolicy | Button |  | Click | Redirect to Term and policy page |  |
| 10 | lbTitle | Label |  |  | Show title of the article |  |
| 11 | lbAuthor | Label |  |  | Show author of this article |  |
| 12 | lbDate | Label |  |  | Show current date |  |

TASK SHEET

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No 1** | | **Project Title** | | **Activity Plan Prepared By** | | **Date of Preparation of Activity Plan** | | | | | |
| **Sr. No** | **Task** | |  | |  | | **Start Date** | **End Date** | **Actual Days** | **Member’s name** | **Status** |
| **CRS document** | | | | | | | | | | | |
| 1 | Introduction | | 24H Website | | V.X.M. Tuấn | | 09/17/2017 | 09/22/2017 | 5 | All of Team | FINISH |
| 2 | Existing Scenario | | 24H Website | | V.X.M. Tuấn | | 09/17/2017 | 09/22/2017 | 5 | All of Team | FINISH |
| 3 | Customer Requirement Specification | | 24H Website | | V.X.M. Tuấn | | 09/17/2017 | 09/22/2017 | 5 | All of Team | FINISH |
| 4 | Functional Requirement Specification | | 24H Website | | V.X.M. Tuấn | | 09/17/2017 | 09/22/2017 | 5 | All of Team | FINISH |
| 5 | System Requirements | | 24H Website | | V.X.M. Tuấn | | 09/17/2017 | 09/22/2017 | 5 | All of Team | FINISH |
| 6 | Development Software | | 24H Website | | V.X.M. Tuấn | | 09/17/2017 | 09/22/2017 | 5 | All of Team | FINISH |
| **Use Case and Sequence Diagrams** | | | | | | | | | | | |
| 7 | Guest register as Reader | | 24H Website | | V.X.M. Tuấn | | 09/22/2017 | 09/22/2017 | 1 | V.X.M. Tuấn | FINISH |
| 8 | Guest Login as Reader | | 24H Website | | V.X.M. Tuấn | | 09/22/2017 | 09/22/2017 | 1 | V.X.M. Tuấn | FINISH |
| **ERD** | | | | | | | | | | | |
| **GUI** | | | | | | | | | | | |
| **Report** | | | | | | | | | | | |
| **Permission Security** | | | | | | | | | | | |
| **Task sheet** | | | | | | | | | | | |
| 112 | Task sheet | | 24H Website | | V.X.M. Tuấn | | 09/17/2017 |  | 1 | V.X.M. Tuấn | FINISH |