|  |
| --- |
|  |
| Project Initiation Document |
| Newsfish Content Management system |
|  |
| **Client: Downsview Advocate** |
|  |

|  |
| --- |
|  |

**Contents**

1. Document Information …………………………………………………………………………………………………………………
   1. Document Authors ……………………………………………………………………………………………………………......
   2. Revision History ………………………………………………………………………………………………………………………
   3. Document Approvals ………………………………………………………………………………………………………………
   4. Document Purpose …………………………………………………………………………………………………………………
2. Project Overview ………………………………………………………………………………………………………………………….
   1. Project Description …………………………………………………………………………………………………………………
   2. Positioning ……………………………………………………………………………………………………………………………..
      1. The Business Opportunity ………………………………………………………………………………………………
      2. Problem Statement ………………………………………………………………………………………………………..
   3. Stakeholders and Users …………………………………………………………………………………………………………..
      1. Stakeholders …………………………………………………………………………………………………………………..
      2. Stakeholder and User Needs …………………………………………………………………………………………..
      3. Product Position Statement ……………………………………………………………………………………………
      4. Constraints ……………………………………………………………………………………………………………………..
      5. Operating Environment ………………………………………………………………………………………………….
3. Process and Data Modeling ….…………..…………………………………………………………………………………………
   1. Data Flow Diagram (As Is System) .………………………………………………………………………………………….
   2. Data flow Diagram (To Be System) ……………………… …………………………………………………………………
4. Architecture …………………………………………………………………………………………………………………………………
   1. Technologies Used ………………………………………………………………………………………………………………….
   2. Operational, Performance, Security Requirements …………………………………………………………………
5. Design …………………………………………………………………………………………………………………………………………..
   1. Business Rules ………………………………………………………………………………………………………………………..
   2. Business Use Case Diagram …………………………………………………………………………………………………….
   3. Use Case Specifications …………………………………………………………………………………………………………..
      1. Use Case Specification: ………..…………………………………………………………………………………………
      2. Use Case Specification: …………………………………………………………………………………………………..
   4. Domain Class Diagram …………………………………………………………………………………………………………….
6. Database Design …………………………………………………………………………………………………………………………..
7. Interface Mock-ups ………………………………………………………………………………………………………………………
8. Acceptance Criteria ………………………………………………………………………………………………………………………
9. Implementation schedule for PRJ 666 …………………………………………………………………………………………..
10. **Document Information**
    1. **Document Authors**

|  |
| --- |
| Author’s Name |
| Peng Qu |
| Yong Fu Liu |
| Bo Qu |
|  |
| Professor’s Name |
| Mark Buchner, Mohamed Kassim |

* 1. **Revision History**

|  |  |  |
| --- | --- | --- |
| Version | Date | Note |
| 1.0 | June 23, 2017 | First Draft |
| 2.0 | August 4, 2017 | Final Version |
|  |  |  |
|  |  |  |

* 1. **Document Approvals**

**Date:**

X .

Name of Client/Rep

Company Name

* 1. **Document Purpose**

The purpose of this document is to provide the specifications for the system. It will describe what the system will look like, as well as define the features the system will be able to perform.

1. **Project Overview**
   1. **Project Description (including Project Background)**

This project is for the Downsview Advocate newspaper. It is a content management system. Right now, the workflow is like this: the writers submit articles to the editor in various forms. The editor arranges all the articles to the realtimeboard web app and blocks off. The sales manager puts all the ads on and blocks off too. The editor and sales manager upload the articles and ads to google drive. The designer checks the space arrangement from realtimeboard and grabs the contents from the google drive and designs. There are a lot of human labours and cost a lot of time.

The new system will place all the contents into one database management system. This system allows the writers to upload the articles, the photographers to upload the photos, the editor to grab and modify the contents, and the sales manager to upload the advertisements via browser. This system guarantees the latest version of the content. Content edit histories will be stored and can be traced. The system also displays the articles online, schedules options, publishes the content to social media and embeds in processing the Metadata entry and optimization tools.

* 1. **Positioning**
     1. **The Business Opportunity**

This system has given the opportunity to develop a computerized system for the Downsview Advocate, whose workflow is currently based on google drive.

* + 1. **Problem Statement**

|  |  |
| --- | --- |
| **Problem of** | Current contents managements and web publication are inefficient, error-prone and cause unnecessary labor cost |
| **Affects** | Editor, sales manager, and designer |
| **The Impact of which is** | Limit the team to improve the quality of publication |
| **A successful solution would** | Perform a content management system that allow editors to plan the print publication content, manage print page space and manage web content publication in an automated way |

* 1. **Stakeholders and Users**
     1. **Stakeholders**

|  |  |
| --- | --- |
| Stakeholder | Type of User |
| Manager | User, Sponsor |
| Editor | User |
| \*Administrator | User |
| Sales Manager | User |
| Writers | User |
| Photographers | User |
| Designer | User |
| Developers | Developer |

**\*NOTE: Administrator is the system administrator, not management administrator**

* + 1. **Stakeholder and User Needs**

|  |  |
| --- | --- |
| Actor | Description |
| Manager | Oversees all the operations |
| Editor | Oversees all the content in terms of the articles |
| Administrator | Manages the system |
| Sales manager | Submits all of the advertisements |
| Writers | Contribute contents |
| Photographers | Contribute photos |
| Designer | Designs ads and layout |
| Developers | Develop the system |

* + 1. **Product Position Statement**

|  |  |
| --- | --- |
| For: | All staff members in Downsview Advocate |
| Who: | Requires one easier way to perform the content management |
| The: | “Newsfish” is a content management system |
| That: | Provides improved print content management and automated web content publication functions |
| Unlike: | Current manual management solution |
| Our Project | Will make complex processes simple and automated, greatly improve the efficiency of the current management system |

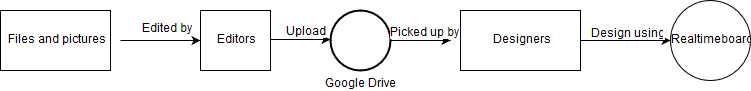
* + 1. **Constraints**

|  |  |  |
| --- | --- | --- |
| Constraint | Description | Workaround or Solution |
| Limited time to implement | Due to the eight month window for PRJ566 and PRJ666, developing the contents management system with a small team will require a lot of hard, dedicated work to create all the product components (database design, database implementation, web application UI design). | Due to our team members all have some experience on PHP and MySQL. We will choose matured LAMP technology to implements our design. |

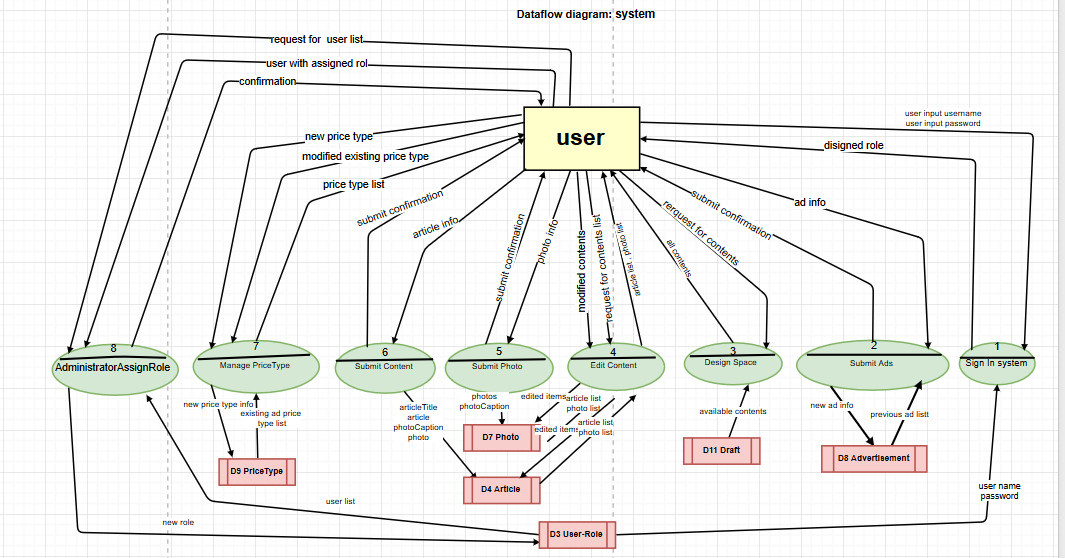
* + 1. **Operating Environment**

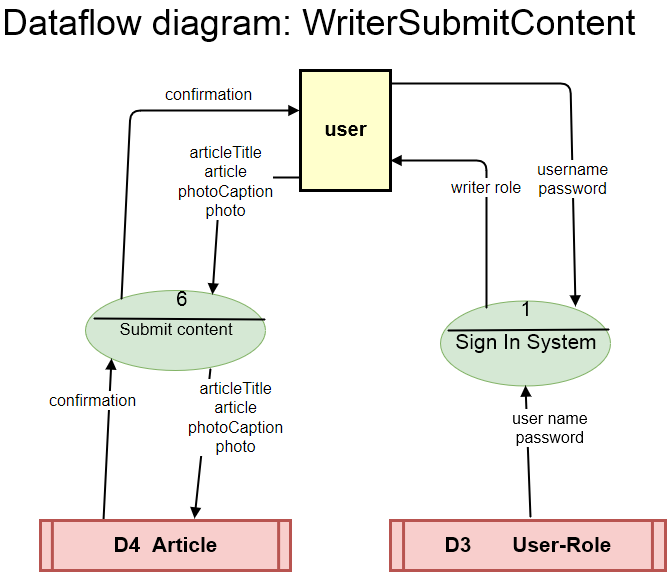
• LAMP –Linux, Apache, MySQL, and PHP( need to verify current system environment)

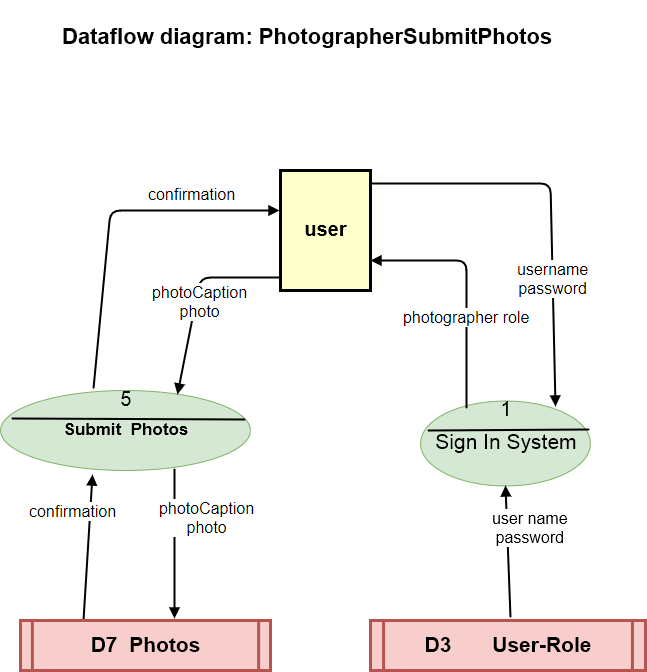
1. **Process and Data Modeling**
   1. **Data Flow Diagram (As Is System)**

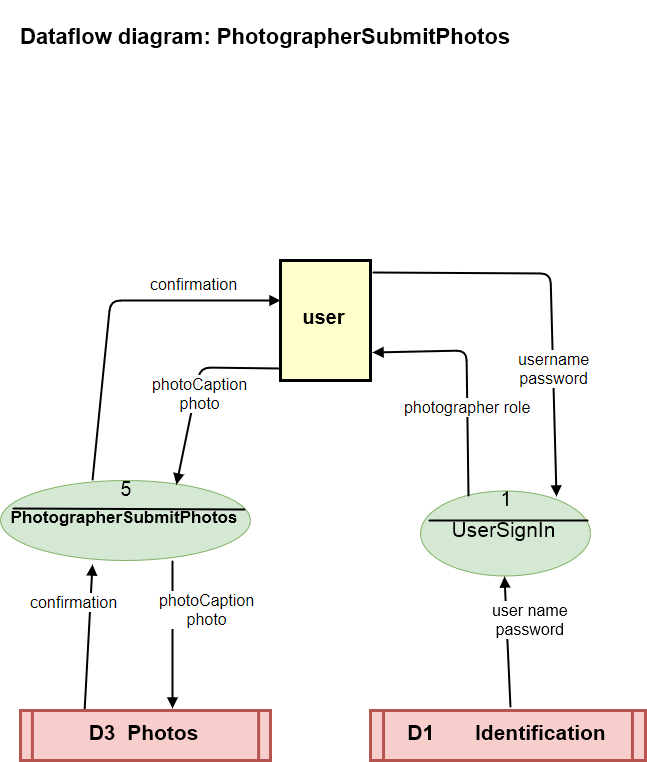
****

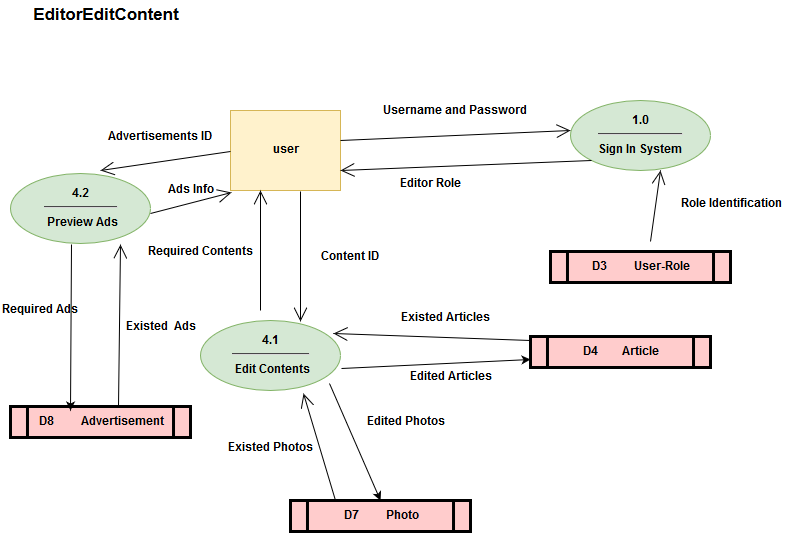
* 1. **Data Flow Diagram (To Be System)**

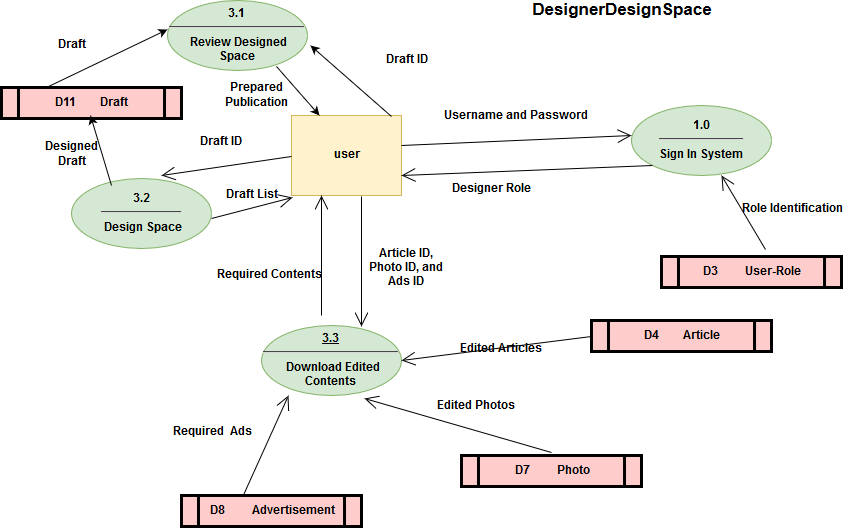
****

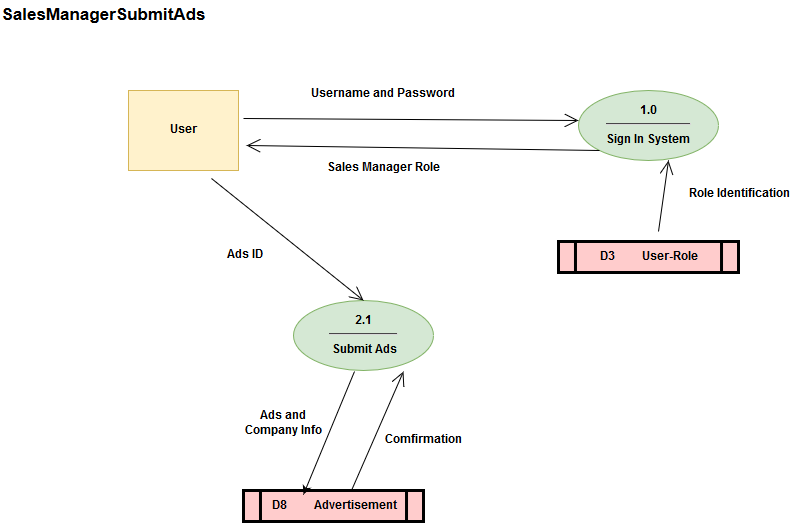
****

****

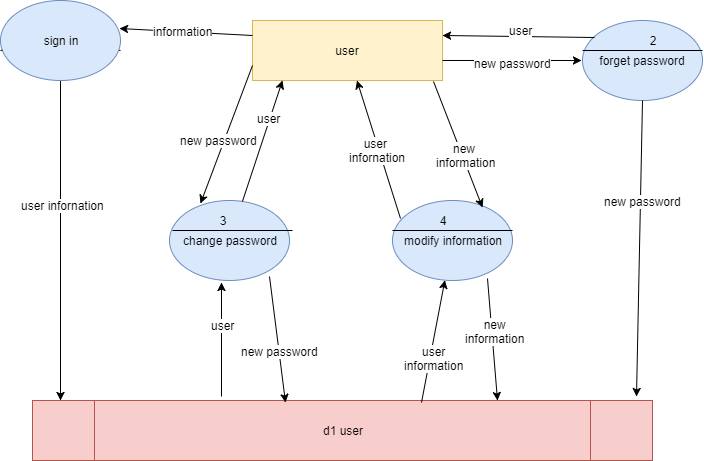
****

****

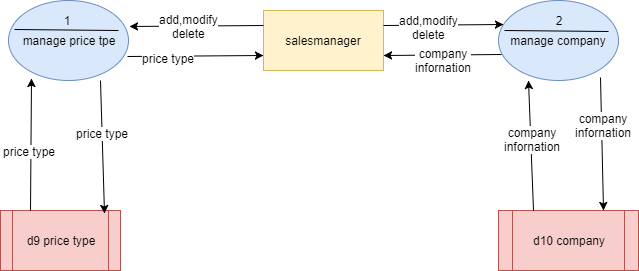
****

****

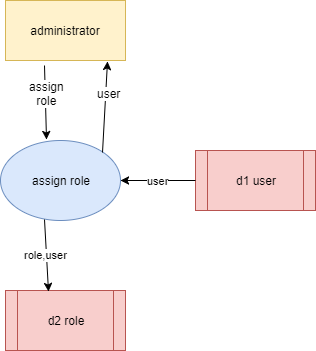
**UserManagerInformation**

****

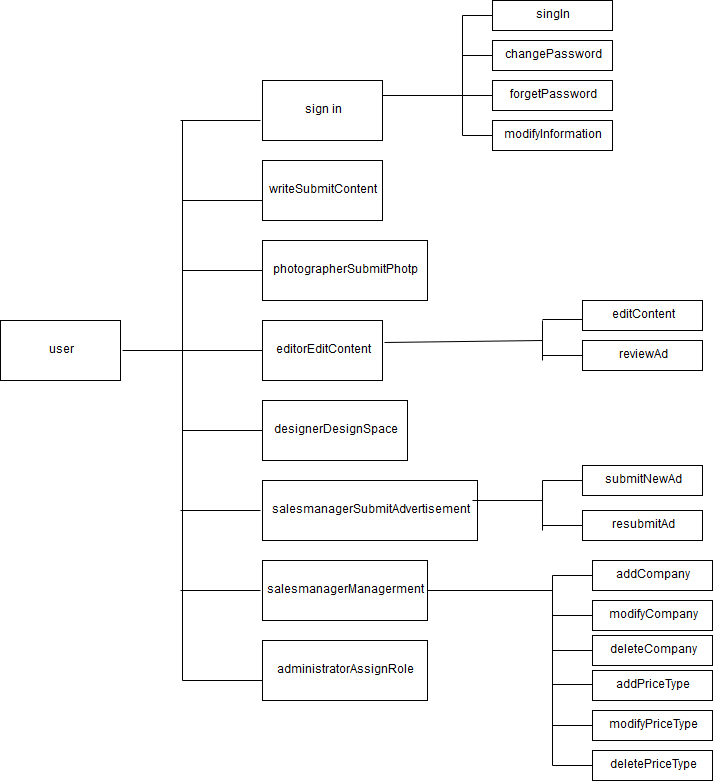
**SalesmanagerManagement**

****

**AdministratorAssignRole**

****

**Work Breakdown Structure:**

****

1. **Architecture** 
   1. **Technologies Used**

**LAMP - Linux Operation System**

**Apache HTTP Server**

**MySQL relational database management system**

**PHP**

* 1. **Operational Requirements**

**N/A**

* 1. **Performance Requirements**

**No requirement**

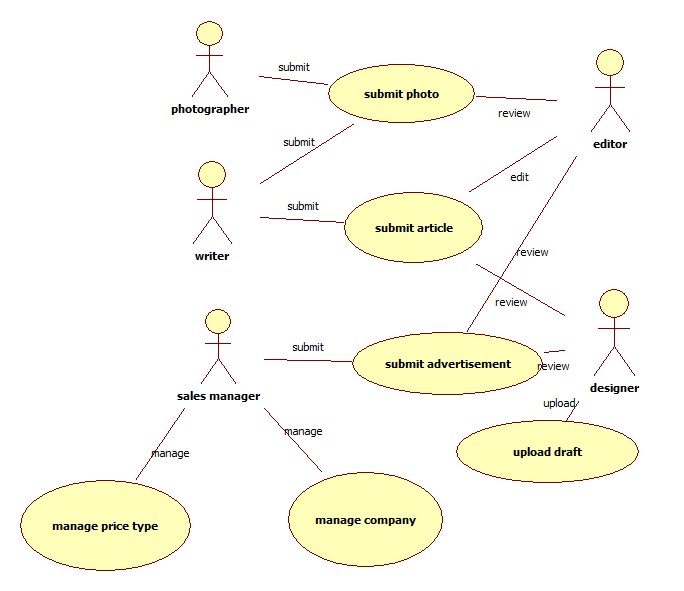
* 1. **Security Requirements**

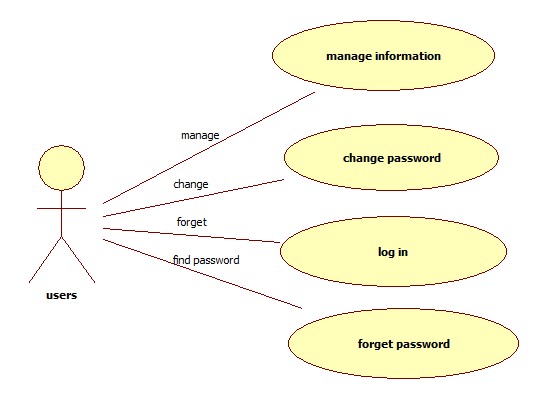
**User account protected security system**

1. **Design**
   1. **Business Rules**

|  |  |  |
| --- | --- | --- |
| Rule # | Description | Reference |
| 1 | **The file format that writer submit must be MS word document or plain text format** |  |
| 2 | **The contents must be edited before space design** |  |
| 3 | **All the advertisement must be submitted only by the sales manager.** |  |
| 4 | **Web version must be published after print version** |  |
| 5 | **Content must has a title and photos must have captains** |  |
| 6 | **Edit must be completed before deadline so the Designer have time to do space design** |  |
| 7 | **User’s email must be unique** |  |
| 8 | **Company name is not empty** |  |
| 9 | **Password and re-password must match** |  |
| 10 | **Telephone number must be 10 digits** |  |
| 11 | **PriceMin is less than priceMax, both of them are double** |  |
| 12 | **Edit must be completed before deadline so the Designer have time to do space design** |  |
|  |  |  |

* 1. **Business Use Case Diagram**

****

****

* 1. **Use Case Specifications**
     1. **Use Case Specification: writer submit contents**

**Identifier: WriterSubmitContent**

**Brief Description:**  This use case enables the writer to submit articles and photos

**Section 1: Business Rule(s):**

* Content must has a title
* The articles format must be MS word documents or plain text

**Section 2: All Scenarios:**

**Scenario 1: writer submit contents**

Precondition: writers must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor( writer)** | **System** | **Data Used** |
| **1** | **Request upload article** | **Display submit form, prompting input title, photo caption, photo upload, file upload, article** | **userId** |
| **2** | **Fill the form and submit** | **Validate format, title, if success;**  **If file is not MS document, parse article to photos and plain text, save the article’s text and photo to database. If not, go to \*A1** | **articleId, articleTitle, timestamp, article, photoId, file, photoCaption, photo** |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor( writer)** | **System** | **Data Used** |
|  |  | **Highlight invalid items, Hint actor to re-input items.** |  |
| **1** | **Modify input items and resubmit** | **Go to main flow step 2** |  |

Post Conditions: articles or photographs submitted and saved to the database

* + 1. **Use Case Specification: photographer submit photos**

**Identifier: PhotographerSubmitPhotos**

**Brief Description:**  This use case enables the photographers to submit photos

**Section 1: Business Rule(s):**

* photos must have captains

**Section 2: All Scenarios:**

**Scenario 1: photographer submit photos**

Precondition: photographer must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(photographer)** | **System** | **Data Used** |
| **1** | **Request upload photos** | **Display upload form, prompting upload photo caption, photo.** | **userId** |
| **2** | **Select photos and upload** | **Validate form, if success, save the content to database. If not, go to \*A1** | **photoCaption, photo** |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(photographer)** | **System** | **Data Used** |
|  |  | **Highlight invalid items, Hint actor to re-input items.** |  |
| **1** | **Modify input items and resubmit** | **Go to main flow step 2** |  |

Post Conditions: articles or photographs submitted and saved to the database

* + 1. **Use Case Specification: Editor edit contents**

**Identifier: EditorEditContents**

**Brief Description:**  This use case enables editor to edit contents, set editing status, set content other attributes such as location (page number of the print news page), web version location.

**Section 1: Business Rule(s):**

* Edit must be completed before deadline so the Designer have time to do space design

**Section 2: All Scenarios:**

**Scenario 1: Editor edit the content**

Precondition: Editor must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor( editor )** | **System** | **Data Used** |
| **1** | **Request edit content** | **Display available articles list and photo list** | **articleId, articleTitle,**  **photoId, photoCaption, photo,** |
| **2** | **Select a content to edit** | **Display article ( if it is article allow edit text), list and allow available attributes to be editable**  **If content is photo: editable attributes include: articleTitle, caption, photoEditFinishOrNot**  **If contents is article: editable attributes includes: location, category, articleTitle, photos, articleEditFinishOrNot** | **photoEditFinishOrNot**  **location, category, articleTitle, photos**  **articleEditFinishOrNot** |
| **3** | **Edit and submit** | **If content is article, move original article to history table and save edited article and attributes settings to database** | **articleTitle,**  **photoCaption, article, category, location, articleId, historyTimeStamp** |

Post Conditions: contents and content attributes setting are edited and saved to the database

**Scenario 2: editor review advertisement**

Precondition: At the moment when the use case begins, the system is displaying the list of advertisement with a prompt of review

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| **request to review** | **display the advertisement detail.** | **advertisementId,companyName, companyAddress, name, email, phone, designOption, contentRevision, printOption, printSizeW, printSizeH,webSizeW, webSizeH, type** |
| **choose exit** | **display the list of advertisement** |  |

* + 1. **Use Case Specification: Designer design space**

**Identifier: DesignerDesignSpace**

**Brief Description:**  This use case enables designer to pull out all final contents currently available in database and save design back to database

**Section 1: Business Rule(s):**

* All content must be edited by the Editor before space designing

**Section 2: All Scenarios:**

**Scenario 1: Designer grab available contents**

Precondition: Designer must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(designer )** | **System** | **Data Used** |
| **1** | **Request available contents for space design** | **Display all available edited contents list** |  |
| **2** | **Select contents to download and request download** | **Pull out content from database and send to the designer** |  |

Post Conditions: available contents are downloaded to the local computer

**Scenario 2: Designer upload designed result to database**

Precondition: Designer must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(designer )** | **System** | **Data Used** |
| **1** | **Request upload designed result** | **Display upload form** |  |
| **2** | **Select the space designed file to upload**  **And submit** | **Save the file to the database** |  |

Post Conditions: space design file is uploaded to the database

**Scenario 3: Designer Review Space Design Result**

Precondition: Designer must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(designer )** | **System** | **Data Used** |
| **1** | **Request review space design file** | **Display space design result file link, and allow download to local computer** |  |
| **2** | **choose download file** | **Send space design result file to designer.** |  |

Post Conditions: space design file is downloaded to the local computer

* + 1. **Use Case Specification: Sales manager submit advertisements**

**Identifier: SalesManagerSubmitAdvertisements**

**Brief Description:**  This use case enables sales manager to submit ads to database

**Section 1: Business Rule(s):**

* Only sales manager can submit the advertisements
* Company name is not empty

**Section 2: All Scenarios:**

**Scenario 1: Sales manager submit new advertisement**

Precondition: sales manager must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(Sales manager)** | **System** | **Data Used** |
| **1** | **Request submit new ads** | **Display submit form which must specify companyName, companyAddress, name, email, phone, designOption, content, printOption, printSizeW, printSizeH, webSizeW, webSizeH, priceType** |  |
| **2** | **Upload ads, enter the companyName, companyAddress, name, email, phone designOption, content, printOption, printSizeW, printSizeH, webSizeW, webSizeH, priceType** | **Company name is not empty, print size is int, store the content into the system, display list of advertisement.** | **advertisementId,companyName, companyAddress, name, email, phone, designOption, content, printOption, printSizeW, printSizeH,webSizeW, webSizeH, priceType, timestamp** |
| **3** | **Choose exit** | **Display the list of advertisements** |  |

Post Conditions: New ads are submitted and saved to the database

**Scenario 2: Sales manager resubmit previous advertisements**

Precondition: sales manager must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(** **Sales manager)** | **System** | **Data Used** |
| **1** | **Request submit previous ads** | **Display the advertisement details with a prompt of resubmit** |  |
| **2** | **Choose resubmit** | **Company name is not empty, print size is int, store the content into the system, display list of advertisement.** | **advertisementId, companyName, companyAddress, name, email, phone, designOption, contentRevision, printOption, printSizeW, printSizeH,webSizeW, webSizeH, priceType, timestamp** |
| **3** | **Choose exit** | **Display the list of articles** |  |

Post Conditions: Previous ads are re-submitted and saved to the database

* + 1. **Use Case Specification: User sign in**

**Identifier: UserSignIn**

**Brief Description:**  This use case enables user regist to the system.

**Section 1: Business Rule(s):**

● User email must be unique

● Password & re-password must match

● Telephone number must be 10 numbers

● PriceMin is less than priceMax

● PriceMin and priceMax are both double

**Section 2: All Scenarios:**

**Scenario 1: UserSignIn**

Precondition: At the moment when the use case begins, the system displays home page

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
| **1** | **Request to sign in** | **Display the sign in page** |  |
| **2** | **Enter the email, password,**  **re-password, first name, last**  **name, address, telephone,**  **facebook, twitter, role** | **Validate email address, if not go to \*A1.**  **Validate password , if not go to \*A2.**  **If all inputs are valid ,save to database, confirm new account is created, display log in page.** | **email, password, firstName, lastName, address, phone, faceBook, twitter, role** |
|  |  |  |  |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
|  |  | **Display the account already exited. Please use another email address.** |  |
| **1** | **Modify input email address and resubmit** | **Go to main flow step 2** |  |

**Alternative flow: A2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
|  |  | **Display invalidation message** |  |
| **1** | **Enter password and repassword** | **Go to main flow step 2** |  |

Post Conditions: The user successfully signed in the system

**Scenario 2: User Change Password**

Precondition: User must log in

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
| **1** | **Request to change the password** | **Display the change password page** |  |
| **2** | **Enter the origin password, new password, re-password** | **System check the origin password is correct, the origin password does not match the new password, the password matches re-password, i f not go to \*A1. Update the new password. Display the welcome page.** | **password** |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
|  |  | **Display invalidation message** |  |
| **1** | **Enter password and repassword** | **Go to main flow step 2** |  |

Post Conditions: User change password successfully

**Scenario 3: User forget Password**

Precondition: User does not log in the system

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(** **User)** | **System** | **Data Used** |
| **1** | **Request “forget password”** | **Display the “forget password” page** |  |
| **2** | **Enter the email** | **Validate email, if match, send an email with a link for changing the password. If not match, go to \*A1** | **email** |
| **3** | **Input password, re-password** | **check the password matches the re-password, update to database. If not match, go to \*A2** | **password** |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
|  |  | **Display no such account, input email address again** |  |
| **1** | **Modify input email address and resubmit** | **Go to main flow step 2** |  |

**Alternative flow: A2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
|  |  | **Display invalidation message** |  |
| **1** | **Enter password and repassword** | **Go to main flow step 2** |  |

Successful Post Conditions: User changes password successfully

**Scenario 4: User Modify Information**

Precondition: User must log in the system

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
| **1** | **Request to modify information** | **Display the user’s information** | **phoneNumber, firstName, lastName, facebook, twitter** |
| **2** | **Enter the information** | **Update database, display welcome page** | **email** |

Post Conditions: user modify information successfully

**Scenario 5: User login**

Precondition: System must be on

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
| **1** | **Request login** | **Display the login page** |  |
| **2** | **Enter the user email and password** | **Validate email and password, if not match, go to \*A1. If match, go to account welcome page** | **email, password** |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(User)** | **System** | **Data Used** |
|  |  | **Highlight invalid input,** | **email, password** |
| **1** | **Enter email and password** | **Go to main flow step 2** |  |

Post Conditions: User login successfully

* + 1. **Use Case Specification: Sales manager manage priceType and company information**

**Identifier: SalesManagerManagePriceType**

**Brief Description:**  This use case enables sales manager to manage the price type.

**Section 1: Business Rule(s):**

* priceMin must not greater than priceMax
* priceMin and PriceMax must be double

**Section 2: All Scenarios:**

**Scenario 1: Delete PriceType**

Precondition: Sales manager must log in, system displays the page with prompt of manage price and manage advertisement

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(Sales manager)** | **System** | **Data Used** |
| **1** | **Request to manage price type** | **Display the manage price type page with the list of price type and a prompt of add new** | **priceTypeId, priceType, priceMin, priceMax** |
| **2** | **Choose one type to delete** | **Confirm to delete** |  |
| **3** | **Confirm to delete** | **Update database, display the manage price type page.** |  |

Post Conditions: sales manager delete price type successfully

**Scenario 2: ModifyPriceType**

Precondition: Sales manager must log in, system displays the page with prompt of manage price and manage advertisement

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(Sales manager)** | **System** | **Data Used** |
| **1** | **Request to manage price type** | **Display the manage price type page with the list of price type and a prompt of add new** | **priceTypeId, priceType, priceMin, priceMax** |
| **2** | **Choose one type to modify** | **Display the price type** |  |
| **3** | **Enter new priceType, priceMin, priceMax** | **Verity the priceMin is less than the priceMax. If not, go to \*A1. verity they are both double, if not goto \*A2. update the database, display manager priceType page** | **priceTypeId, priceType, priceMin, priceMax** |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(sales manager)** | **System** | **Data Used** |
|  |  | **Display priceMin should be not greater than priceMax** |  |
| **1** | **Modify priceMin, priceMax** | **Go to main flow step 2** |  |

**Alternative flow: A2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(sales manager)** | **System** | **Data Used** |
|  |  | **Display priceMin and priceMax should be double** |  |
| **1** | **Modify priceMin, priceMax** | **Go to main flow step 2** |  |

Post Conditions: sales manager modify price type successfully

**Scenario 3: AddPriceType**

Precondition: Sales manager must log in, system displays the page with prompt of manage price and manage advertisement

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(Sales manager)** | **System** | **Data Used** |
| **1** | **Request to manage price type** | **Display the manage price type page with the list of price type and a prompt of add new** | **priceTypeId, priceType, priceMin, priceMax** |
| **2** | **Choose add new priceType** | **Display the add price type page** |  |
| **3** | **Enter priceType, priceMin, priceMax** | **Verity the priceMin is less than the priceMax.if not, go to \*A1. verity they are both double, if not goto \*A2. save to the database, display manager pricetype page** | **priceTypeId, priceType, priceMin, priceMax** |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(sales manager)** | **System** | **Data Used** |
|  |  | **Display priceMin should be not greater than priceMax,** |  |
| **1** | **Modify priceMin, priceMax** | **Go to main flow step 2** |  |

**Alternative flow: A2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(sales manager)** | **System** | **Data Used** |
|  |  | **Display priceMin and priceMax should be double** |  |
| **1** | **Modify priceMin, priceMax** | **Go to main flow step 2** |  |

Post Conditions: sales manager add price type successfully

**Scenario 4: DeleteCompany**

Precondition: Sales manager must log in, system displays the page with prompt of manage company information

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(Sales manager)** | **System** | **Data Used** |
| **1** | **Request to manage company information** | **Display the manage company page** | **companyId, companyName, companyAddress, name, email, phone** |
| **2** | **Choose one company to delete** | **Confirm to delete** |  |
| **3** | **Confirm to delete** | **Update database, display the manage company page.** |  |

Post Conditions: sales manager delete company successfully

**Scenario 5: ModifyCompany**

Precondition: Sales manager must log in, system displays the page with prompt of manage company

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(Sales manager)** | **System** | **Data Used** |
| **1** | **Request to manage company** | **Display the manage company page with the list of company and a prompt of add new** | **companyId, companyName, companyAddress, name, email, phone** |
| **2** | **Choose one company to modify** | **Display the modify company page** |  |
| **3** | **Enter new companyName, companyAddress, name, email, phone** | **Validate form, update database. If any data is invalid, go to \*A1** |  |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(sales manager)** | **System** | **Data Used** |
|  |  | **Highlight the invalidated data** |  |
| **1** | **Enter data** | **Go to main flow step 2** |  |

Post Conditions: Sales manager modify company successfully

**Scenario 6: AddCompany**

Precondition: Sales manager must log in, system displays the page with prompt of manage company

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(Sales manager)** | **System** | **Data Used** |
| **1** | **Request to add company** | **Display add company page** |  |
| **2** | **Enter company information** | **Validate form, update database. If any data is invalid, go to \*A1** | **companyName, companyAddress, name, email, phone** |

**Alternative flow: A1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(sales manager)** | **System** | **Data Used** |
|  |  | **Highlight invalid data** |  |
| **1** | **Enter data** | **Go to main flow step 2** |  |

Post Conditions: Sales manager add company successfully

* + 1. **Use Case Specification: Administrator assign role**

**Identifier: AdministratorAssignRole**

**Brief Description:**  This use case enables administrator to assign the roles.

**Section 1: Business Rule(s):**

**Section 2: All Scenarios:**

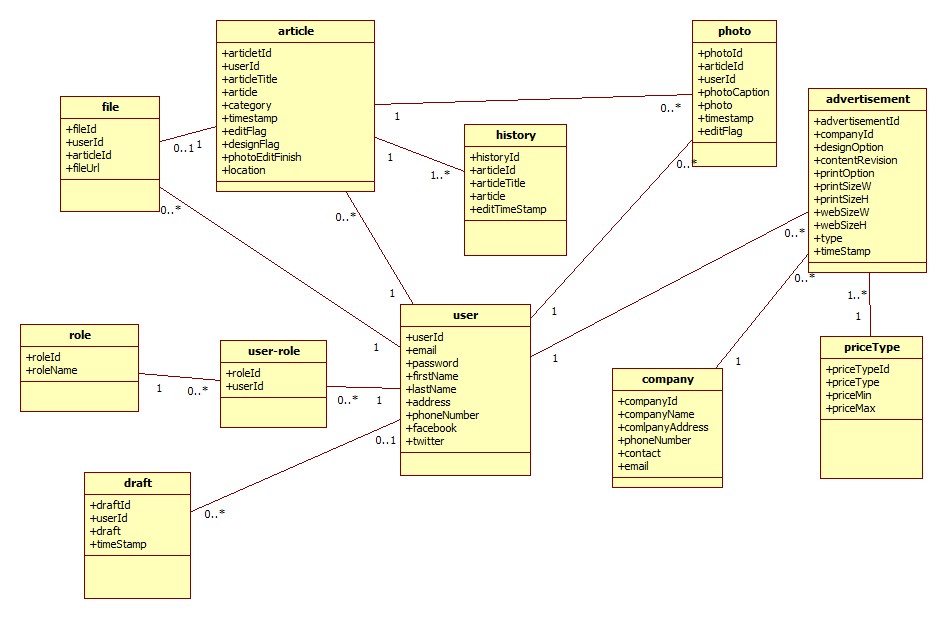
**Scenario 1: Assign Role**

Precondition: Administrator must log in, system displays the page with prompt of assigning roles

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor(Sales manager)** | **System** | **Data Used** |
| **1** | **Request to assign roles** | **Display the user list.** | **userId, firstName, lastName** |
| **2** | **Choose one user to assign** | **Display the role which the use already have, display the role list, add new role to the user or delete the role from the user** | **roleId** |
| **3** | **Confirm to assign** | **Update database, display the user list.** |  |

Post Conditions: Administrator assign role successfully

* 1. **Domain Class Diagram**

****

1. **Database Design**

**6.1 user**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **userId** | **int** | **not null** | **pk** |
| **email** | **nvarchar** | **not null** | **unique** |
| **password** | **nvarchar** | **not null** |  |
| **firstName** | **nvarchar** | **not null** |  |
| **lastName** | **nvarchar** | **not null** |  |
| **address** | **nvarchar** |  |  |
| **phoneNumber** | **nvarchar** |  |  |
| **facebook** | **nvarchar** |  |  |
| **twitter** | **nvarchar** |  |  |

**6.2 role**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **roleId** | **int** | **not null** | **pk** |
| **roleName** | **nvarchar** | **not null** |  |

**6.3 user-role**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **roleId** | **int** | **not null** | **pk,fk** |
| **userId** | **int** | **not null** | **pk,fk** |

**6.4 article**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **articleId** | **int** | **not null** | **pk** |
| **userId** | **int** | **not null** | **fk** |
| **articleTitle** | **nvarchar** | **not null** |  |
| **article** | **nvarchar** | **not null** |  |
| **category** | **nvarchar** |  |  |
| **timeStamp** | **datetime** | **not null** |  |
| **editFlag** | **boolean** | **0** |  |
| **designFlag** | **boolean** | **0** |  |
| **photoEditfinish** | **boolean** | **0** |  |
| **location** | **nvarchar** |  |  |

**6.5 file**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **fileId** | **int** | **not null** | **pk** |
| **userID** | **int** | **not null** | **fk** |
| **articleId** | **int** | **not null** | **fk** |
| **fileUrl** | **nvarchar** | **not null** |  |

**6.6 history**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **historyId** | **int** | **not null** | **pk** |
| **articleId** | **int** | **not null** | **fk** |
| **articleTitle** | **nvarchar** | **not null** |  |
| **article** | **nvarchar** | **not null** |  |

**6.7 photo**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **photoId** | **int** | **not null** | **pk** |
| **articleId** | **int** | **not null** | **fk** |
| **userId** | **int** | **not null** | **fk** |
| **photoCaption** | **nvarchar** | **not null** |  |
| **photo** | **blob** |  |  |
| **timeStamp** | **datetime** |  |  |
| **editFlag** | **boolean** | **0** |  |

**6.8 advertisement**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **advertisementId** | **int** | **not null** | **pk** |
| **companyId** | **int** | **not null** | **fk** |
| **designOption** | **nvarchar** |  |  |
| **contentRevision** | **nvarchar** |  |  |
| **printOption** | **nvarchar** |  |  |
| **printSizeW** | **nvarchar** |  |  |
| **printSizeH** | **nvarchar** |  |  |
| **webSizeW** | **nvarchar** |  |  |
| **webSizeH** | **nvarchar** |  |  |
| **type** | **nvarchar** |  |  |
| **timeStamp** | **datetime** |  |  |

**6.9 priceType**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **priceTypeId** | **int** | **not null** | **pk** |
| **priceType** | **nvarchar** | **not null** |  |
| **priceMin** | **decimal** |  |  |
| **priceMax** | **decimal** |  |  |

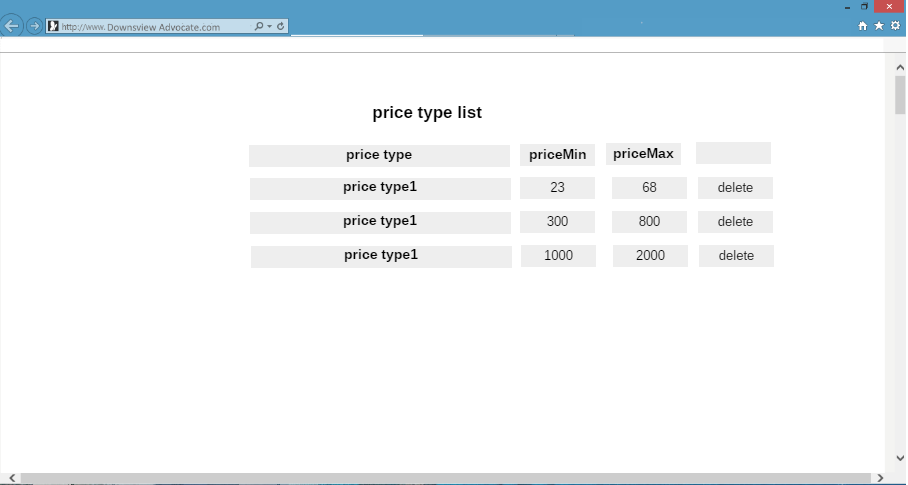
**6.10 company**

|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **companyId** | **int** | **not null** | **pk** |
| **companyName** | **nvarchar** | **not null** |  |
| **companyAddress** | **nvarchar** |  |  |
| **phoneNumber** | **char(10)** |  |  |
| **contact** | **nvarchar** |  |  |
| **email** | **nvarchar** |  |  |

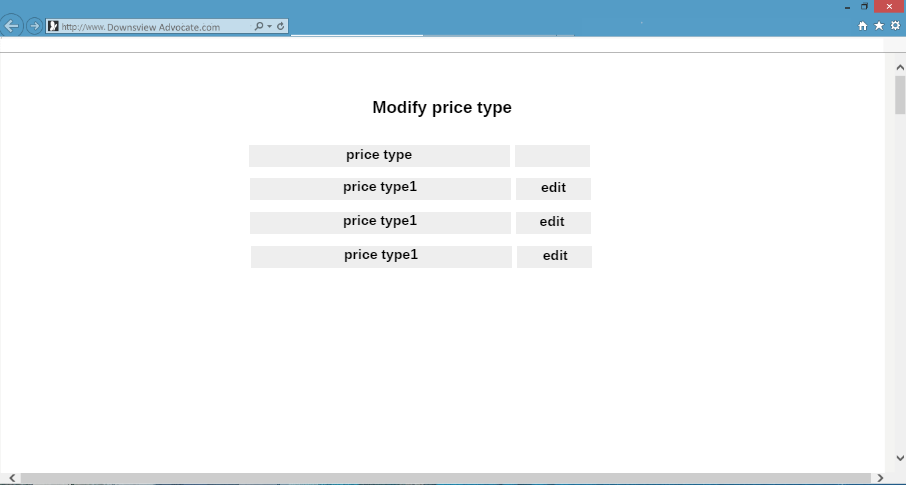
**6.11 draft**

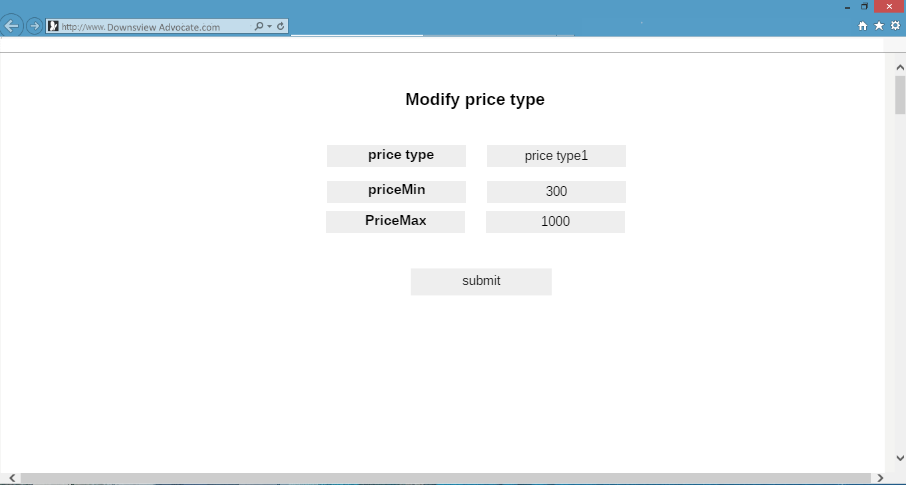
|  |  |  |  |
| --- | --- | --- | --- |
| **column name** | **type** | **default** | **constraint** |
| **draftId** | **int** | **not null** | **pk** |
| **userId** | **int** | **not null** | **fk** |
| **draft** | **blob** |  |  |
| **timeStamp** | **datetime** |  |  |

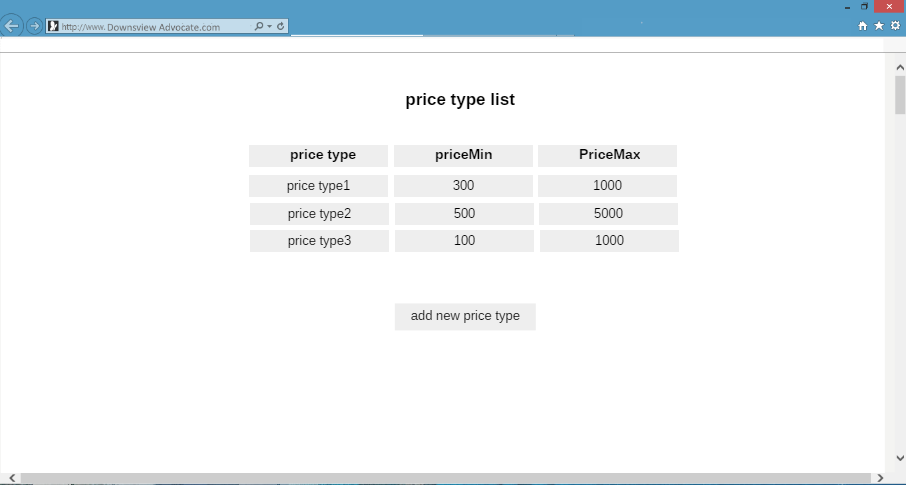
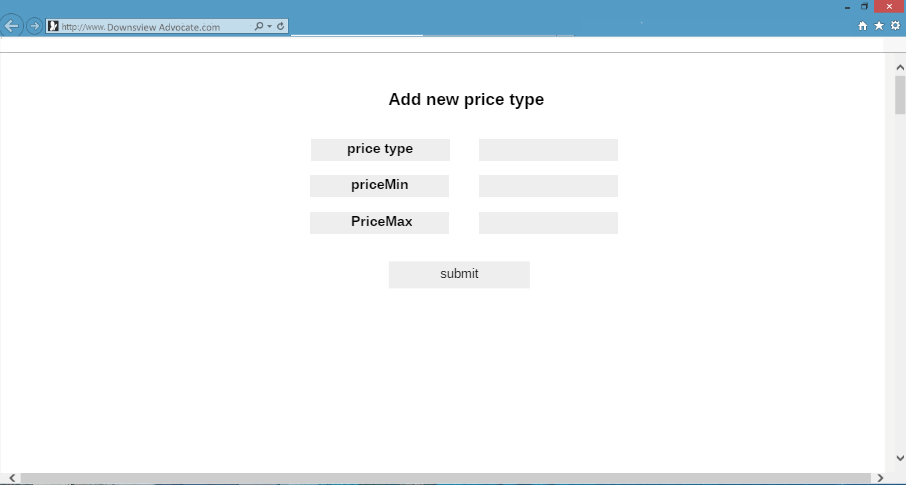
1. **Interface Mock-ups  
     
     
   Sales Manager Delete Price Type**

****

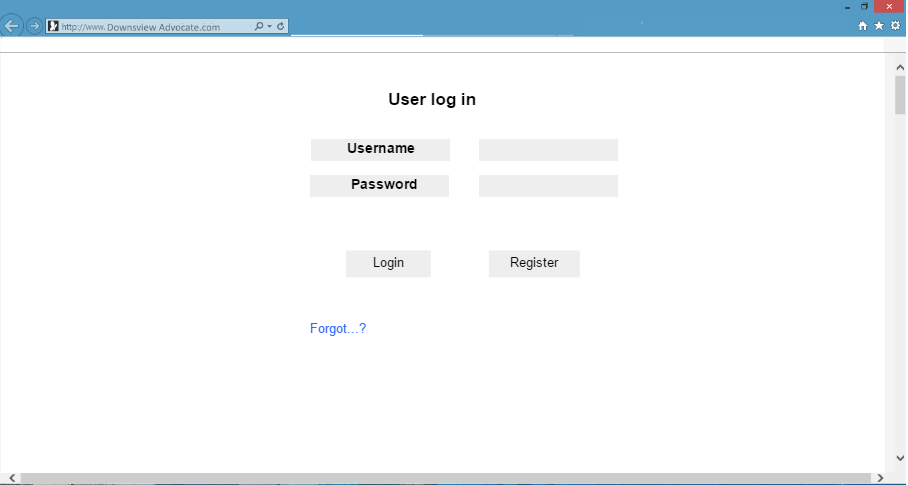
**Sales Manager Modify Price Type**

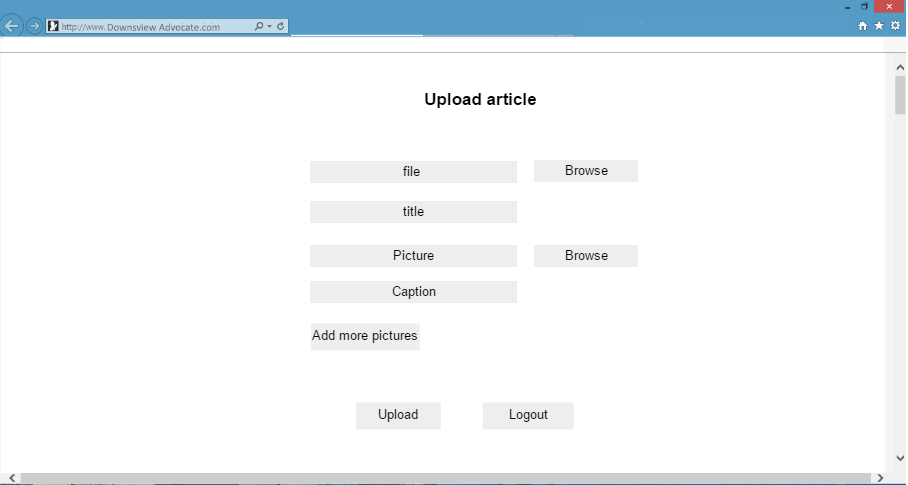
****

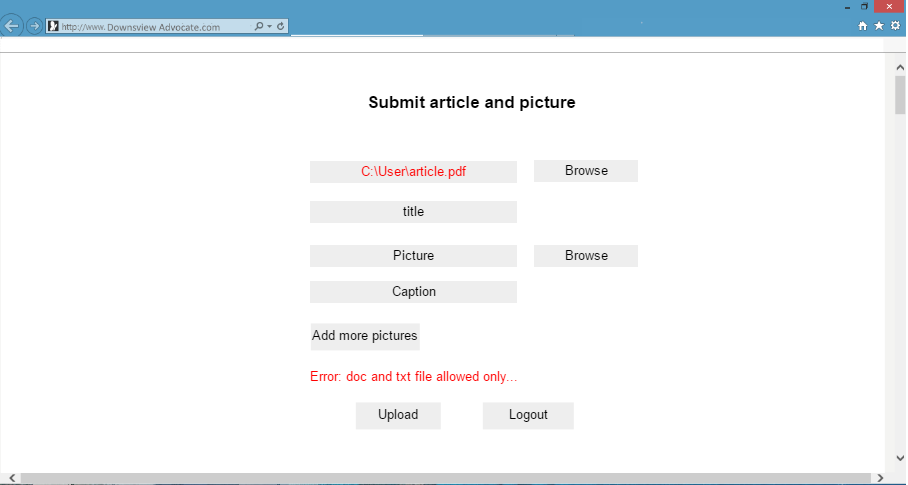
**  
  
Sales Manager Add Price Type**

**  
**

**Writer uploads contents**

****

****

****

1. **Acceptance Criteria**
2. **Implementation schedule for PRJ 666 -** [**MPP File**](https://zenit.senecac.on.ca/~prj566_172a04/files/DownsviewAdvocate.mpp)

****

