**Assignment #1: Project Requirement Documents**

Due date: May 15

In this homework, your job is to make part of requirement document for the project you are assigned. As shown in class, normal requirement documents require lots of things including project scope, functions, constraints, non-functional aspects. This project homework does not cover everything in normal requirement documents. Rather than that, we only focus some important aspects in requirement documents.

Please check the description and submit project assignment to e-class by due date.

1. User Stories

Build user stories from the project description. You have 9 functional requirements from the project description, and you must build user stories as many as the project descrives.

User story has the following simple template.

As a <type of user>, I want <some goal> so that <some reason>.

Example)

1. As an uploader, I want uploading my picture, so that other users can see my picture.
2. As an uploader, I want deleting my old posts, so that others cannot see my old posts.

…

1. Use cases

Build use cases for your project.

Choose NINE user stories from the user stories you created. Then build NINE use cases where each use case corresponds to the any of use case you created. Do not have to create all the use cases from user stories you created.

Here is the use case format.

|  |  |
| --- | --- |
| **Name of Use case** |  |
| **Date Created** |  |
| **Description** |  |
| **Actor** |  |
| **Precondition** |  |
| **Post condition** |  |
| **Flow** |  |
| **Exception** |  |

For reference, we provide some examples.

Example)

|  |  |
| --- | --- |
| **Name of Use case** | Get money from ATM |
| **Date Created** | May 24, 2023 |
| **Description** | User extracts some amount of money from ATM machine. |
| **Actor** | User, ATM |
| **Precondition** | 1. ATM is on. 2. The user’s check card is valid. 3. The user’s account balance is more than the amount of money the user want to extract. |
| **Post condition** | 1. User receives money. 2. User’s account balance gets smaller by the extracted amount. 3. ATM does not have check card inside the ATM machine. |
| **Flow** | 1. User insert a check card to ATM machine. 2. ATM machine validates the check card and prompt PIN number. 3. User enters PIN number for the check card. 4. User enters the amount of money. 5. ATM ejects the check card. 6. ATM ejects the money. 7. ATM ends the user session. |
| **Exception** | 1. In step 2, check card validation is failed.    1. Show message    2. Eject the card.    3. End the user session. 2. In step 3, PIN number is incorrect.    1. Show message to enter new PIN number.    2. Go to step 3 with new PIN number message. 3. In step 4. The amount does not match with user’s account balance.    1. Show message to with the current account balance.    2. Go to step 4 with new amount input message. 4. In step 6, The ATM does not have enough money.    1. Show alarm message.    2. Send the ATM status to bank system.    3. End user session. |
| **Additional Requirements** | 1. All messages should have OK and Cancel option. OK option will proceed to the next step and cancel option will end user session. |

|  |  |
| --- | --- |
| **Name of Use case** | Upload pictures |
| **Date Created** | May 24, 2023 |
| **Description** | User uploads a travel pictures to my account. |
| **Actor** | User, Server |
| **Precondition** | 1. Server is running. 2. User has an account and is logged in his/her account. |
| **Post condition** | 1. Pictures are uploaded and shown in a post. 2. Picture description and hash tags are shown in the post. 3. Pictures are stored in a database. |
| **Flow** | 1. User clicks ‘add post’ button in his/her my page. 2. The server displays ‘add post’ web page to the user’s web browser.. 3. User clicks ‘add picture’ button in the ‘add post’ web page. 4. User can see the dialog box to choose any picture in his/her PC. 5. User clicks ‘OK’ button after choosing preferred pictures. 6. Users writes the description about the pictures in the ‘add post’ web page. 7. User adds hash tags. 8. User clicks upload button to finish uploading process. 9. Server receives descriptions and pictures from user’s browser. 10. Server stores descriptions and pictures to its database. 11. Server sends back newly created posting page to the user’s browser. 12. User sees the newly created posting page. |
| Exception | 1. In step 2, failure in displaying ‘add post’ web page.    1. Show error message with returning back to my page section. 2. In step 9,10,11, failure in upload posting function.    1. Show error message keeping ‘add post’ web page such that user can choose clicking same button again or returning back to my page section.. |
| Additional Requirements | 1. User can choose multiple files in a dialog box. |

1. Test Cases

Create your test cases based on the above use cases. That is, you must build NINE test cases for your implementation. This test cases will be used for testing your implementation after you implement the project.

Here is the format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | |  | **Test Case Description** | |  | | | | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | | | | | | | |
|  |  |  |  |  | |  | |  |  | |  |  |  | | | |
| **Step #** | **Prerequisites:** | |  | | **Step #** | | **Test Data Requirement** | | | | | | | |
| 1 |  | |  | | 1 | |  | | | | | | | |
| 2 |  | |  | | 2 | |  | | | | | | | |
| 3 |  | |  | | 3 | |  | | | | | | | |
| 4 |  | |  | | 4 | |  | | | | | | | |
| 5 |  | |  | | 5 | |  | | | | | | | |
| … |  | |  | | … | |  | | | | | | | |
|  |  |  |  |  | |  | |  |  | |  |  |  | | | |
| **Test Conditions** | | |  |  | |  | |  |  | |  |  |  | | | |
|  |  |  |  |  | |  | |  |  | |  |  |  | | | |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | | | **Pass / Fail / Not executed / Suspended** | | | |
|  | |
| 1 |  | |  | |  | | | | |  | | | |  | |
| 2 |  | |  | |  | | | | |  | | | |  | |
| 3 |  | |  | |  | | | | |  | | | |  | |
| 4 |  | |  | |  | | | | |  | | | |  | |
| 5 |  | |  | |  | | | | |  | | | |  | |
| … |  | |  | |  | | | | |  | | | |  | |

Here is the example. As you have not implemented anything, do NOT fill slots for ‘Actual Results’ and ‘pass/fail/not executed/suspended.’

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | 001 | **Test Case Description** | Get Money from ATM |
| **Created By** | LEE Heeyoung | **Reviewed By** | JUNG Kihoon |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Step #** | **Prerequisites:** | | |  | | **Step #** | **Test Data** | | | | |
| 1 | ATM is working. Customer has a valid check card. | | |  | | 1 | A valid check card | | | | |
| 2 | PIN entering message is showing | | |  | | 2 | PIN = ‘1111’ | | | | |
| 3 | PIN entering message with PIN error message is showing | | |  | | 3 | PIN = ‘0909’ | | | | |
| 4 | Action selection message is showing | | |  | | 4 | Selection = ‘withdraw’ | | | | |
| 5 | … | | |  | | 5 | … | | | | |
| … |  | | |  | | … |  | | | | |
|  |  |  |  |  | |  |  |  |  |  |  |
| **Test Scenario** | | | | | Verify on entering valid userid and password, the customer can login | | | | | | |
|  |  |  |  |  | |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Insert check card | | Password Message comes in | | |  | | | *Not executed* | | |  |
| 2 | Enter invalid password | | Warn for invalid password and ask reentering. | | |  | | | *Not executed* | | |  |
| 3 | Enter valid password | | Show message for selecting action | | |  | | | *Not executed* | | |  |
| 4 | Select ‘withdraw’ | | Show the ‘enter the amount to withdraw’ | | |  | | | *Not executed* | | |  |
| 5 | … | | … | | |  | | |  | | |  |
| … |  | |  | | |  | | |  | | |  |