Criterion B: Design

All Classes:
User Side:

1. borrow page (Inventory.java)
2. borrow extension page (extendReturnPage.java)
3. main page (firstPage.java)
4. question page (questionPage.java)
5. return page (returnBorrowPage.java)
6. instruction page (welcomePage.java)

Admin Side:

1. check borrow record page (adminBorrowPage.java)
2. login page (adminLoginPage.java)
3. check users page (adminUserMaster.java)
4. create new admin page (createNewAdmin.java)
5. add new equipment page (teacherAddInventory1.java)
6. main page (teacherFirstPage.java)
7. check / change / delete equipment page (teacherInventory.java)

Tables in database:

1. adminmaster: stores admin information
2. equipmentmaster: stores equipment information
3. borrowrecord: stores borrow record
4. questionmaster: stores question and feedback information
5. usermaster: stores all user information

User side

Classes	Function	Communicate Database
1. Borrow Page	• Check equipment • Borrow Equipment	• Equipmentmaster • Usermaster • borrowRecord
2. Borrow Extension Page	• Make Extension	• Equipmentmaster • borrowRecord
3. Question Page	• Ask Question	• questionmaster
4. Return Page	• Return Equipment	• borrowRecord • equipmentmaster • usermaster

		- questionmaster

Admin side

1. check borrow record page	- check borrow record	- borrowRecord - Usermaster
2. login page	- verify admin login	- adminmaster
3. check users page	- change user’s borrow authentication	- usermaster
4. create new admin page	- create new admin	- adminmaster
5. add new equipment page	- add new equipment	- equipmentmaster
6. check / change / delete equipment page	- check equipment - change equipment status - delete equipment	- equipmetnmaster
7. main page	- resolve list	- questionmaster

Database
equipmentmaster

Column	Type	Default	
equipmentID	Int	Null	Primary Key, Auto_Increment
category	Varchar(100)	Null	
name	Varchar(500)	Null	
sourceBuyuing	Varchar(500)	Null	
timeBought	Date	Null	
equipmentStatus	Tinyint	3	
currentStatus	Tinyint(1)	0	
lastReview	Date	‘time bought’	Default_Generated
itRoom	Varchar(10)	Null	
specificStoredLocation	Varchar(500)	Null	
maxAllowDuration	Tinyint	14	
extensionTime	Tinyint	7	

timesOfExtension	Tinyint	2	

usermaster

Column	Type	Default	
userId	Int	Null	Primary Key
isStudent	Tinyint(1)	Null	
Name	Varchar(50)	Null	
Class	Varchar(2)	Null	
classNumber	Tinyint	Null	
Password	Int	Null	
timesBorrowRecord	Tinyint	0	
Gender	Tinyint(1)	Null	
lateRecord	tinyint	0	
numberOfBorrowedItems	Tinyint	0	
borrowAuthorization	Tinyint(1)	1	

borrowrecord

Column	Type	Default	
recordId	Int	Null	Primary Key, auto_increment
equipmentId	Int	Null	
timesOfExtensionUsed	Timestamp	Null	
Feedback	Tinyint	0	
transactionTime	Varchar(200)	Current_timestamp	Default_generated
dateStart	Timestamp	Curdate()	Default_generated
dateEnd	Date	Curdate() + interval 14 days	Default_generated

lateOrNot	tinyint(1)	Null	
userId	Int	Null	
isStudent	Tinyint(1)	Null	
name	Varchar(50)	Null	
Class	Varchar(2)	Null	
classNumber	Tinyint	Null	
timesBorrowRecord	Tinyint	Null	
Gender	Tinyint(1)	Null	
lateRecord	Tinyint	Null	
numberOfBorrowedItems	Tinyint	0	
borrowAuthorization	Tinyint(1)	null	

questionmaster

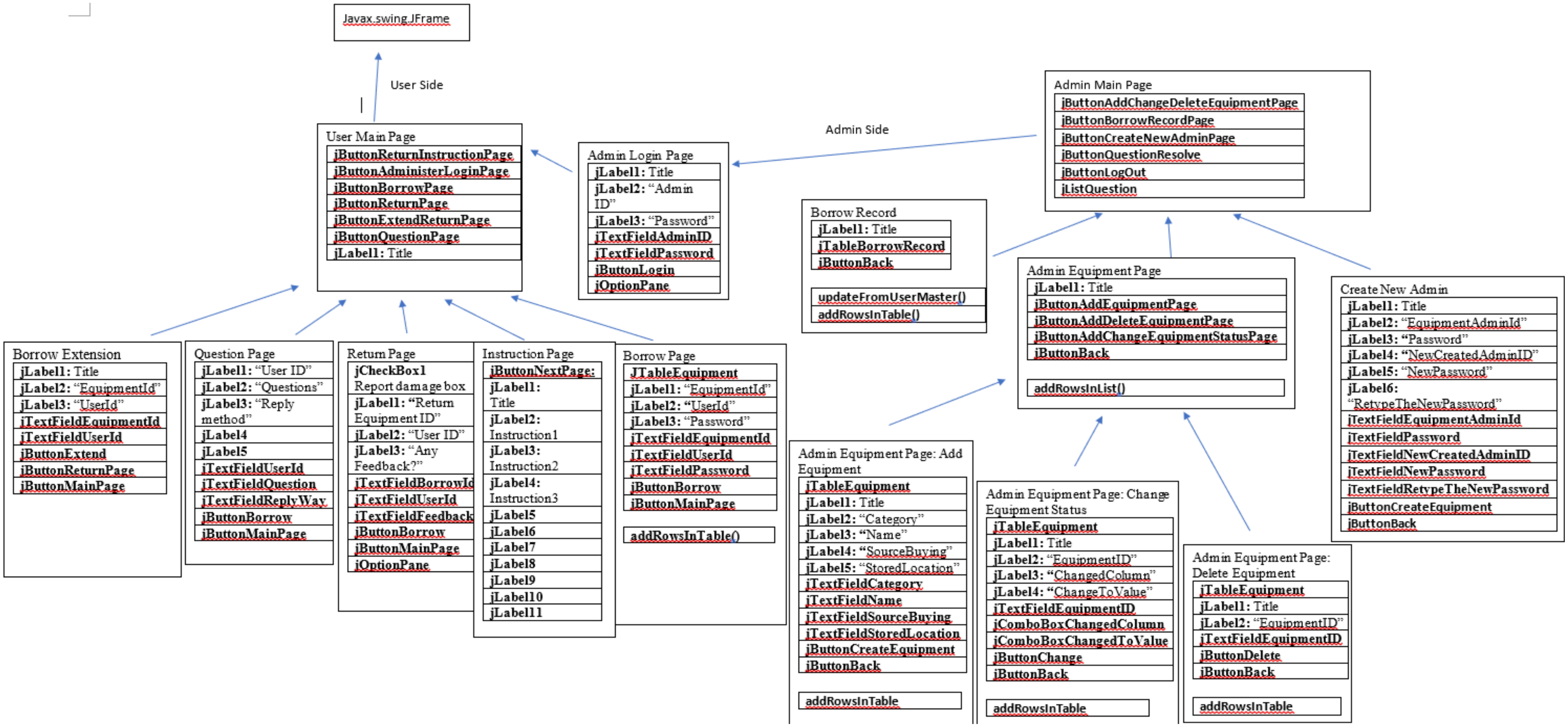
Column	Type	Default	
questionNumber	Int	Null	Auto_Increment
userId	Int	Null	
equipmentId	Int	Null	
questionLocation	Tinyint(1)	Null	
Feedback	varchar(500)	Null	
replyMethod	varchar(500)	Null	
isDamage	Tinyint(1)	Null	
isRead	Tinyint(1)	0	

Adminmaster

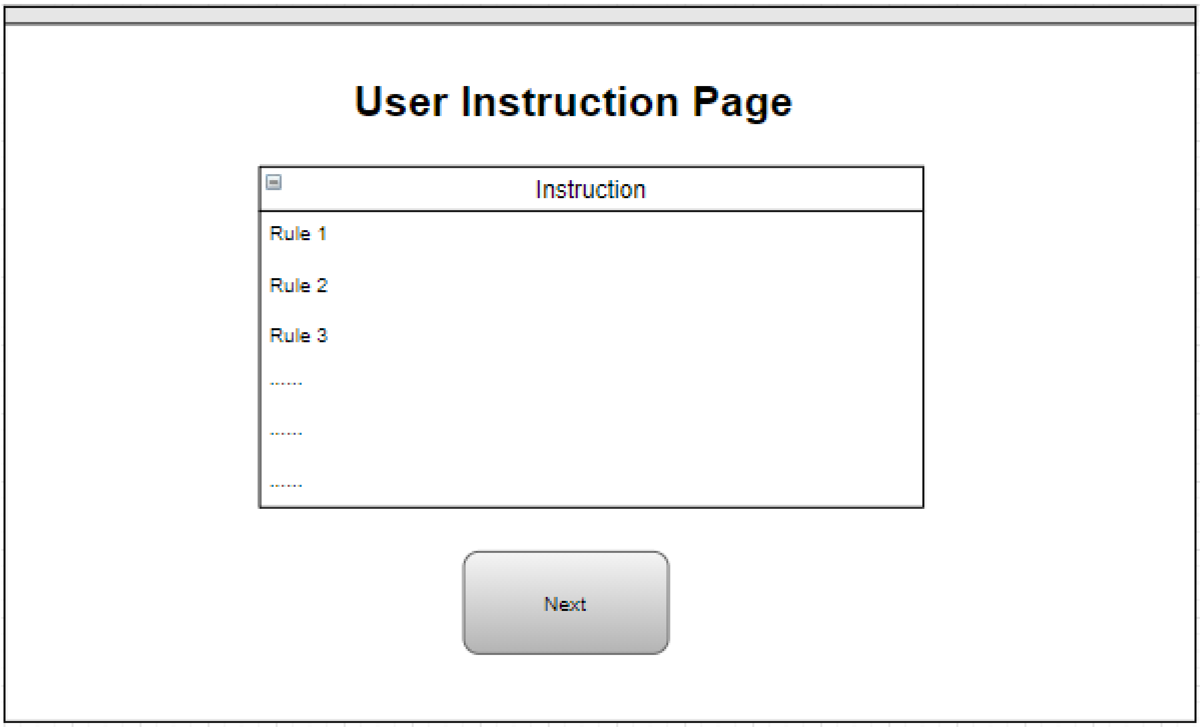
Column	Type	Default	
adminId	Int	Null	

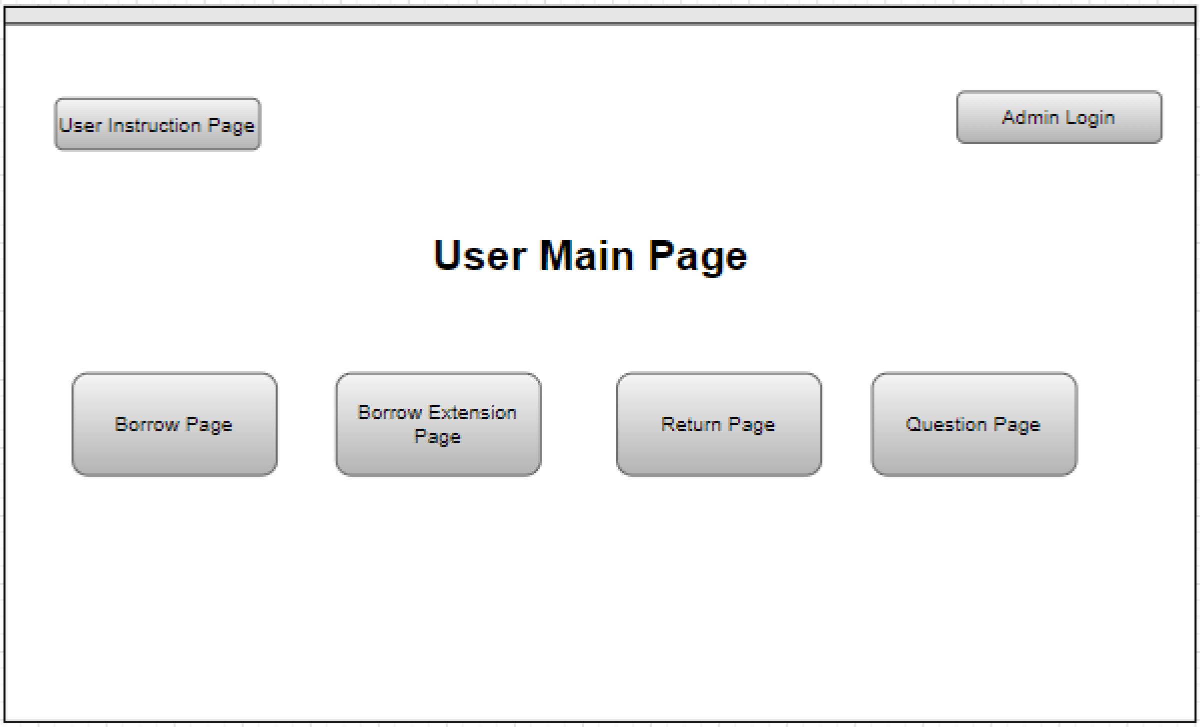
Password	Int	Null	

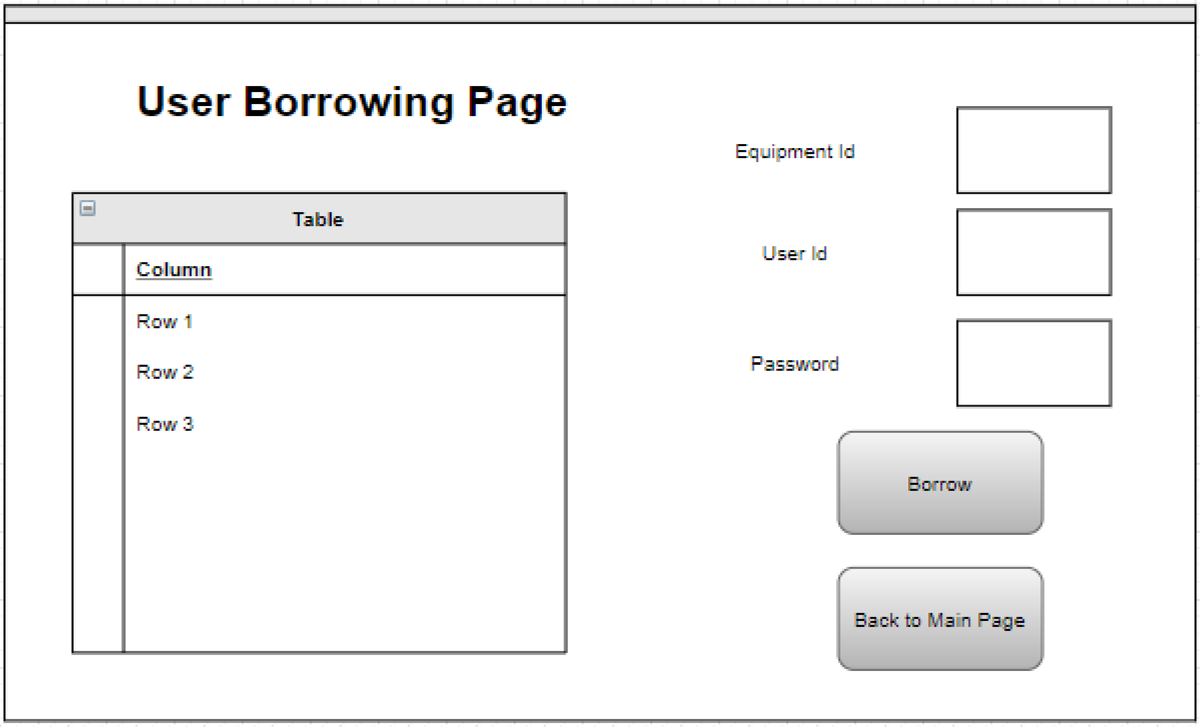
UML Diagram:

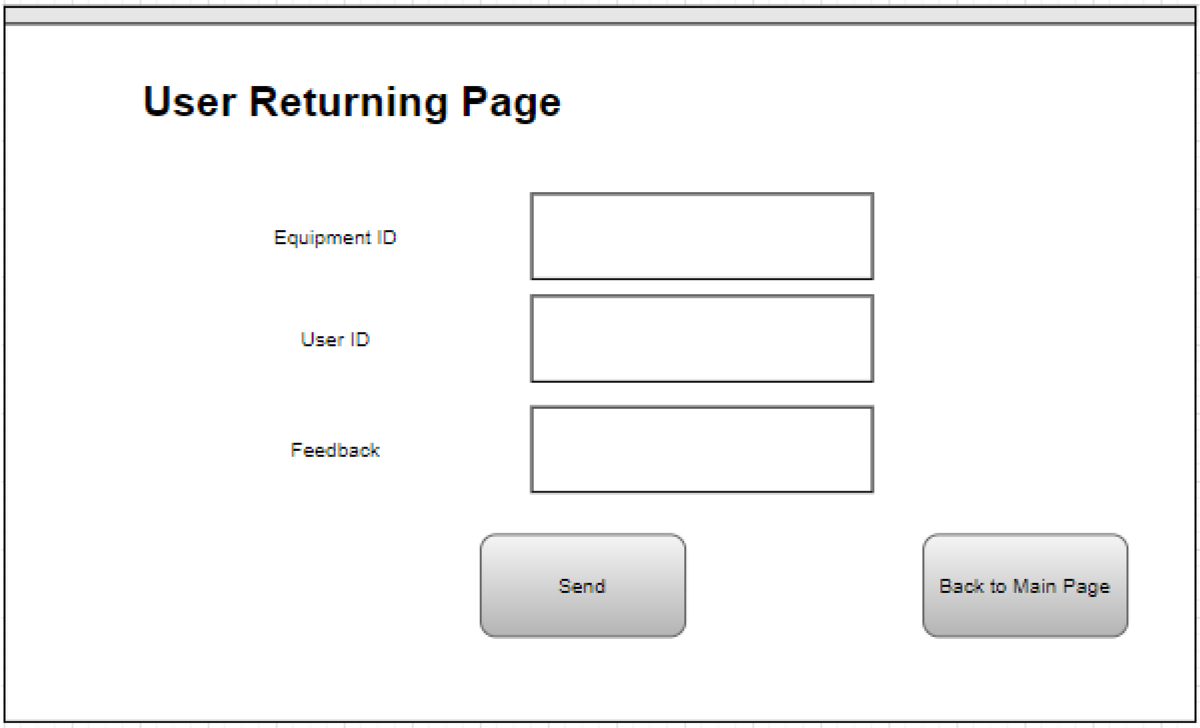

Javax.swing.JFrame
User Side
Admin Side
User Main Page
jButtonReturnInstructionPage
jButtonAdministerLoginPage
jButtonBorrowPage
jButtonReturnPage
jButtonExtendReturnPage
jButtonQuestionPage
jLabel1: Title
Admin Login Page
jLabel1: Title
jLabel2: "Admin ID"
jLabel3: "Password"
jTextFieldAdminID
jTextFieldPassword
jButtonLogin
jOptionPane
Admin Main Page
jButtonAddChangeDeleteEquipmentPage
jButtonBorrowRecordPage
jButtonCreateNewAdminPage
jButtonQuestionResolve
jButtonLogOut
jListQuestion
Borrow Record
jLabel1: Title
jTableBorrowRecord
jButtonBack
updateFromUserMaster()
addRowsInTable()
Admin Equipment Page
jLabel1: Title
jButtonAddEquipmentPage
jButtonAddDeleteEquipmentPage
jButtonAddChangeEquipmentStatusPage
jButtonBack
addRowsInList()
Create New Admin
jLabel1: Title
jLabel2: "EquipmentAdminId"
jLabel3: "Password"
jLabel4: "NewCreatedAdminID"
jLabel5: "NewPassword"
jLabel6: "RetypeTheNewPassword"
jTextFieldEquipmentAdminId
jTextFieldPassword
jTextFieldNewCreatedAdminID
jTextFieldNewPassword
jTextFieldRetypeTheNewPassword
jButtonCreateEquipment
jButtonBack
Borrow Extension
jLabel1: Title
jLabel2: "EquipmentId"
jLabel3: "UserId"
jTextFieldEquipmentId
jTextFieldUserId
jButtonExtend
jButtonReturnPage
jButtonMainPage
Question Page
jLabel1: "User ID"
jLabel2: "Questions"
jLabel3: "Reply method"
jLabel4
jLabel5
jTextFieldUserId
jTextFieldQuestion
jTextFieldReplyWay
jButtonBorrow
jButtonMainPage
Return Page
jCheckBox1 Report damage box
jLabel1: "Return Equipment ID"
jLabel2: "User ID"
jLabel3: "Any Feedback?"
jTextFieldBorrowId
jTextFieldUserId
jTextFieldFeedback
jButtonBorrow
jButtonMainPage
jOptionPane
Instruction Page
jButtonNextPage:
jLabel1: Title
jLabel2: Instruction1
jLabel3: Instruction2
jLabel4: Instruction3
jLabel5
jLabel6
jLabel7
jLabel8
jLabel9
jLabel10
jLabel11
Borrow Page
JTableEquipment
jLabel1: "EquipmentId"
jLabel2: "UserId"
jLabel3: "Password"
jTextFieldEquipmentId
jTextFieldUserId
jTextFieldPassword
jButtonBorrow
jButtonMainPage
addRowsInTable()
Admin Equipment Page: Add Equipment
jTableEquipment
jLabel1: Title
jLabel2: "Category"
jLabel3: "Name"
jLabel4: "SourceBuying"
jLabel5: "StoredLocation"
jTextFieldCategory
jTextFieldName
jTextFieldSourceBuying
jTextFieldStoredLocation
jButtonCreateEquipment
jButtonBack
addRowsInTable
Admin Equipment Page: Change Equipment Status
jTableEquipment
jLabel1: Title
jLabel2: "EquipmentID"
jLabel3: "ChangedColumn"
jLabel4: "ChangeToValue"
jTextFieldEquipmentID
jComboBoxChangedColumn
jComboBoxChangedToValue
jButtonChange
jButtonBack
addRowsInTable
Admin Equipment Page: Delete Equipment
jTableEquipment
jLabel1: Title
jLabel2: "EquipmentID"
jTextFieldEquipmentID
jButtonDelete
jButtonBack
addRowsInTable

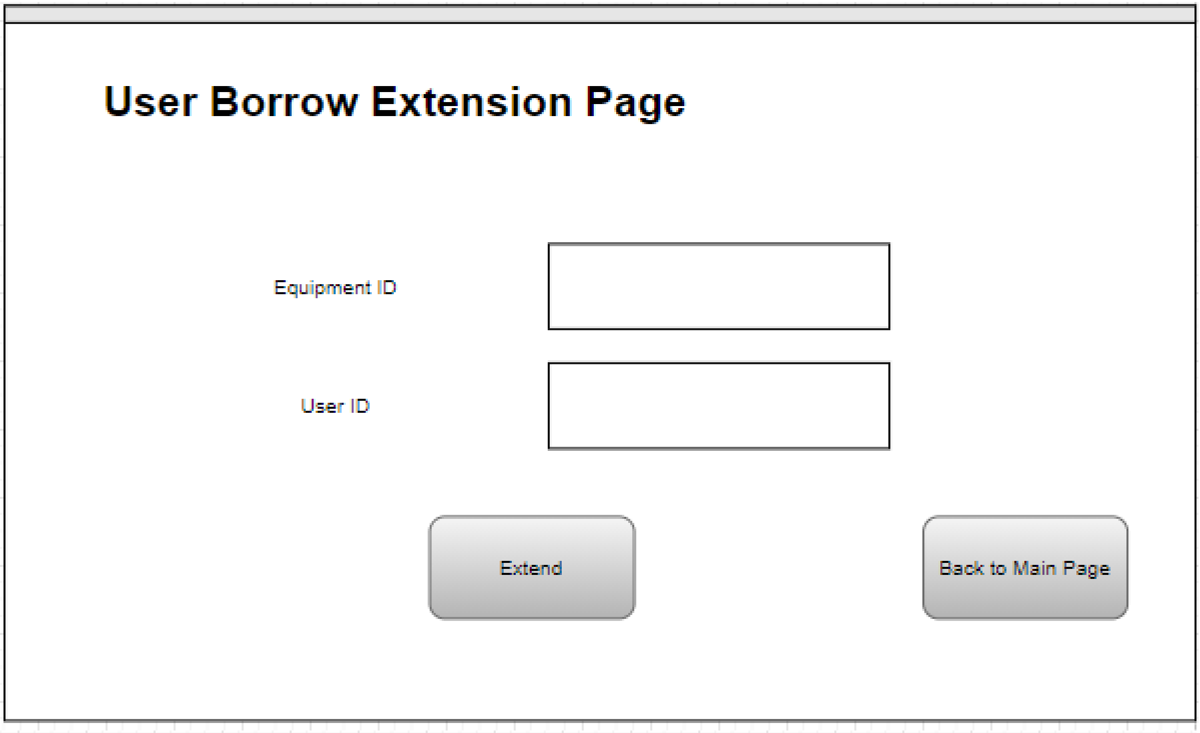
Design of panels (Graphical User Interface)
User Side



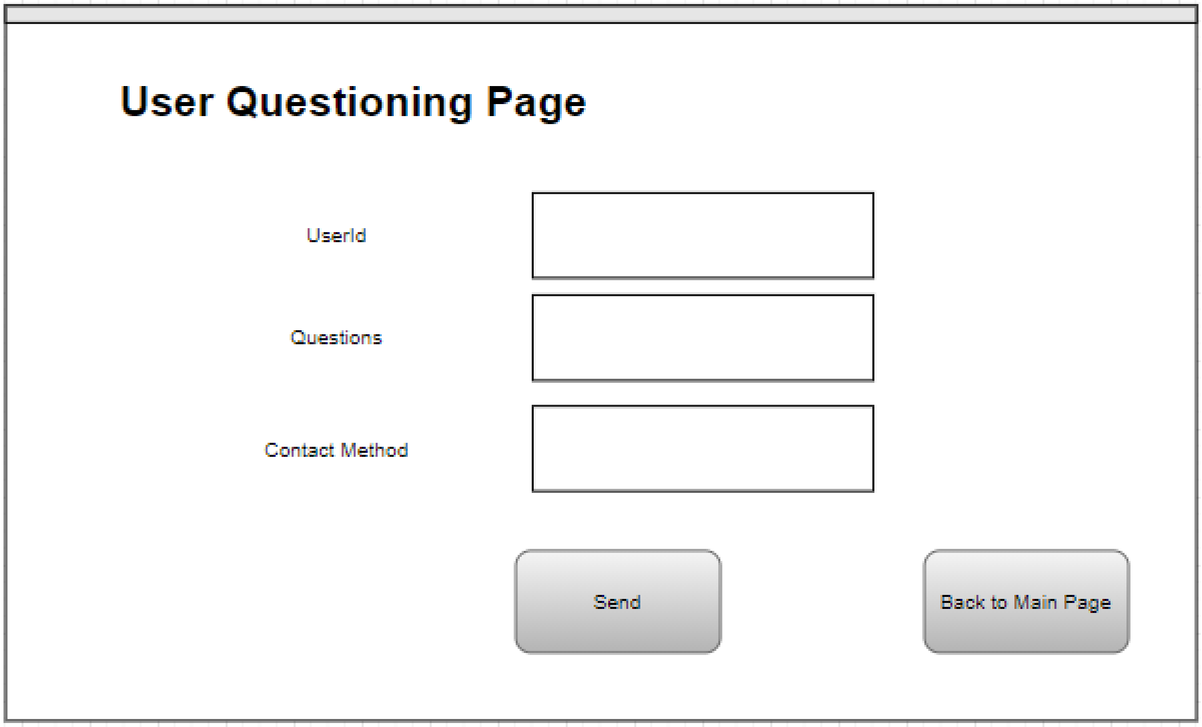



User Borrowing Page
Table
Column
Row 1
Row 2
Row 3
Equipment Id
User Id
Password
Borrow
Back to Main Page


User Returning Page
Equipment ID
User ID
Feedback
Send
Back to Main Page

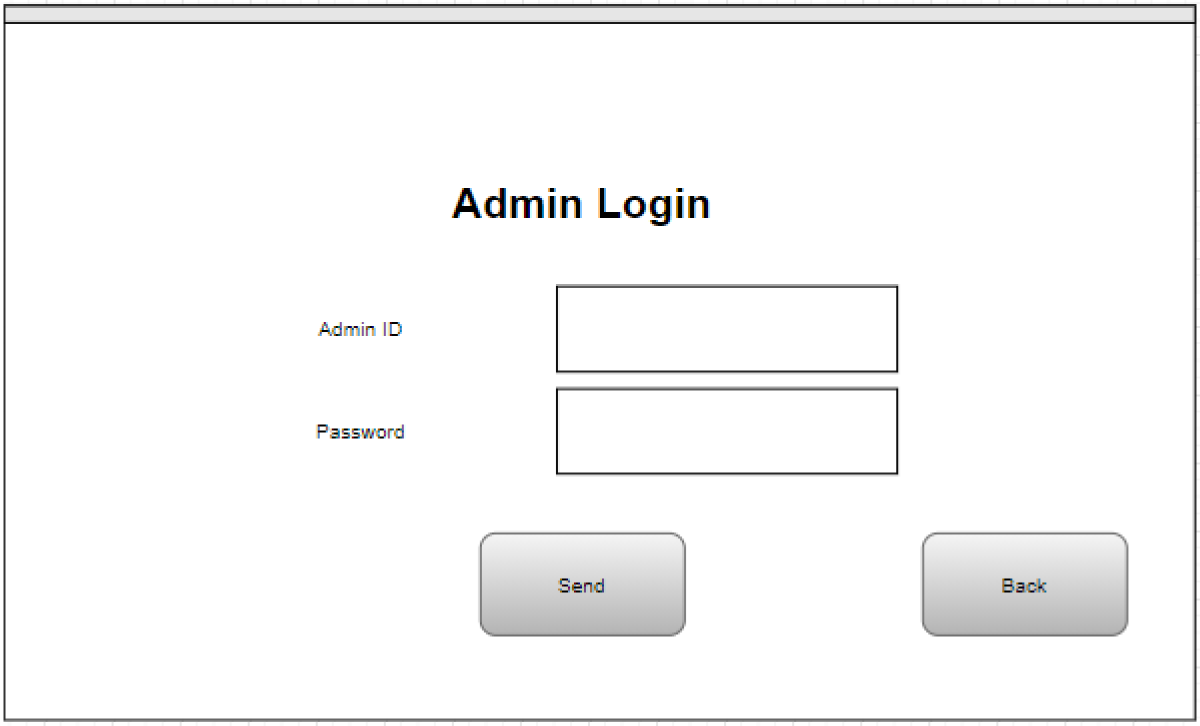


User Borrow Extension Page

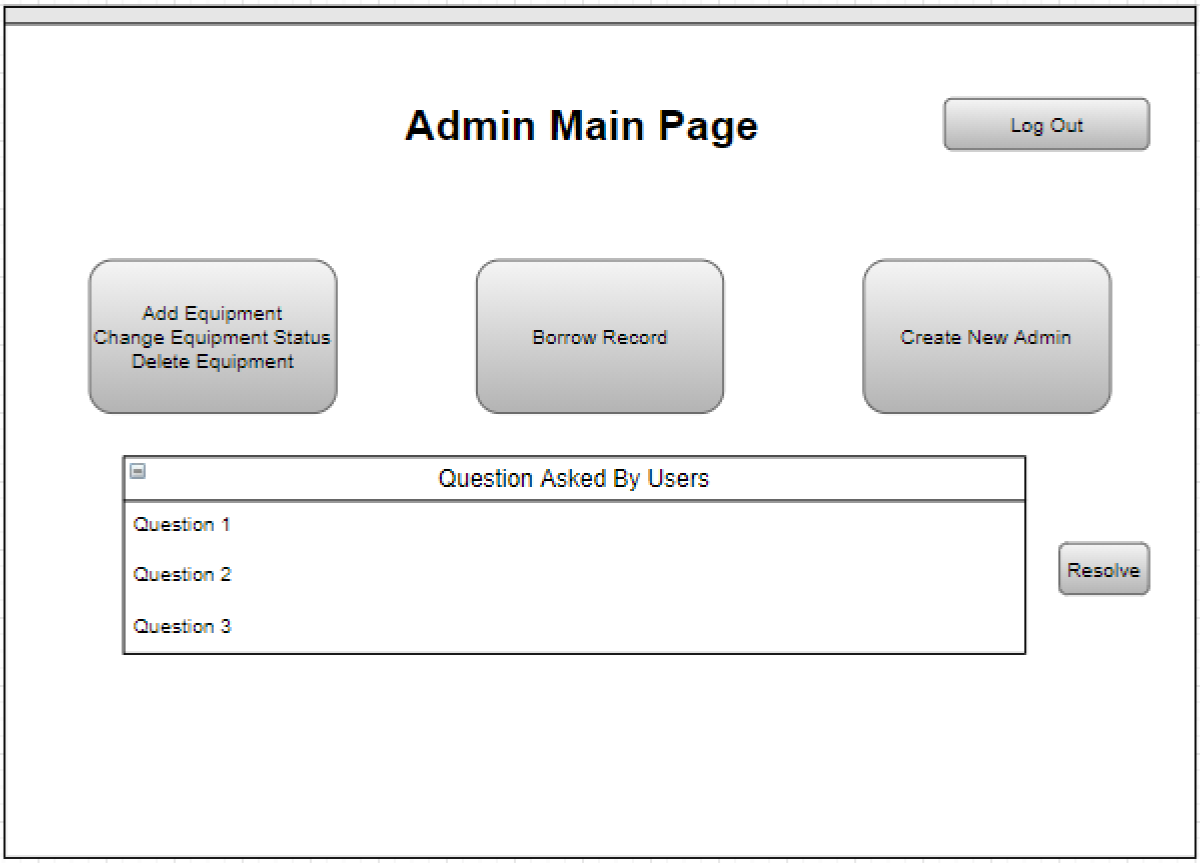


User Questioning Page

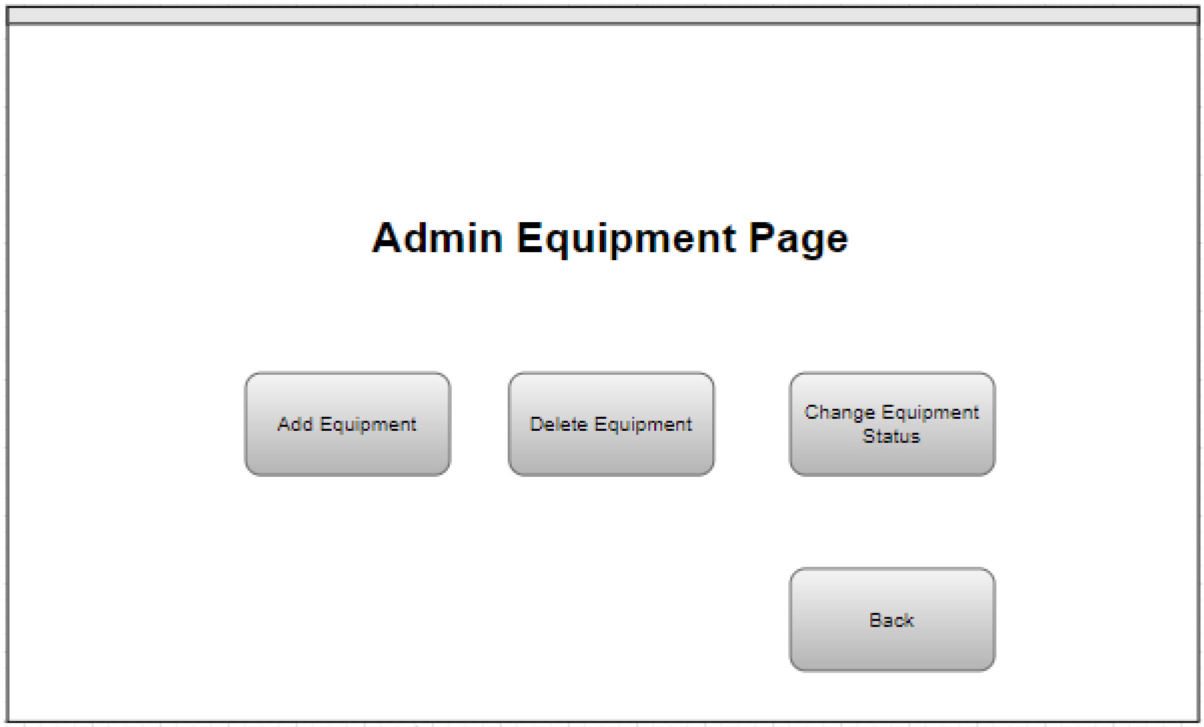
Admin Side:

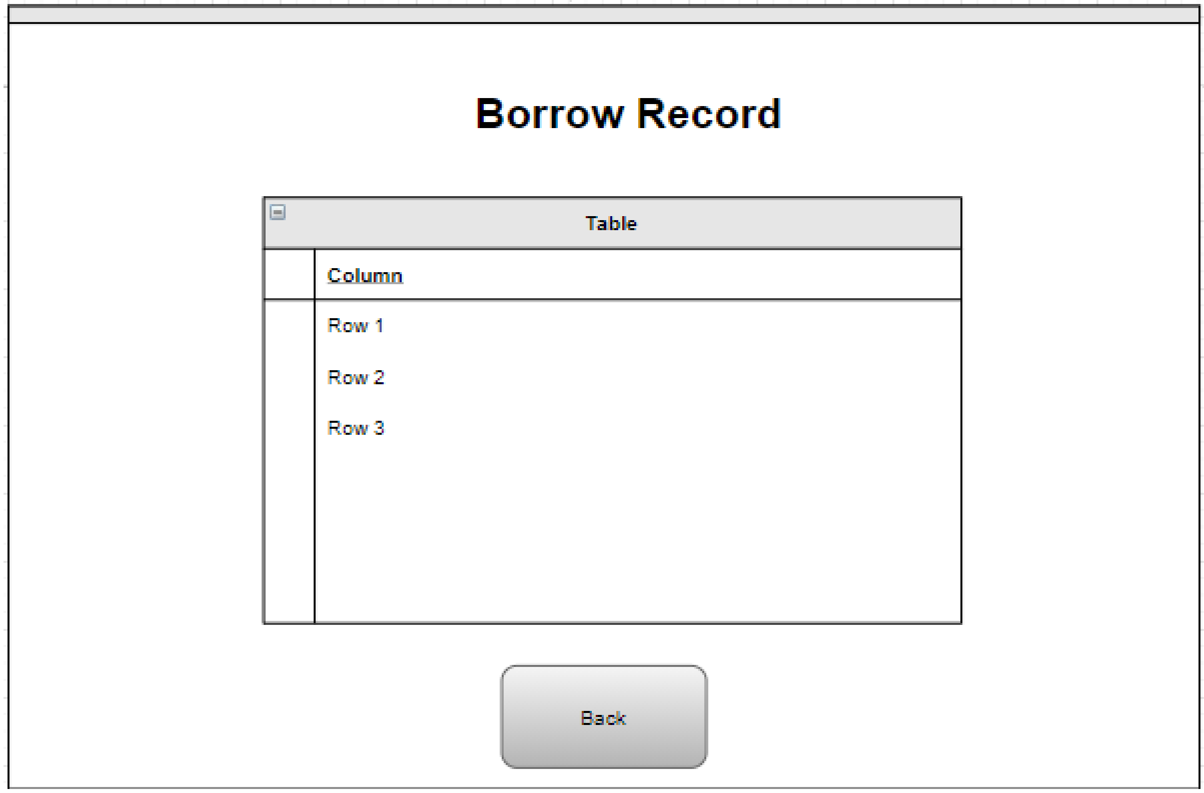


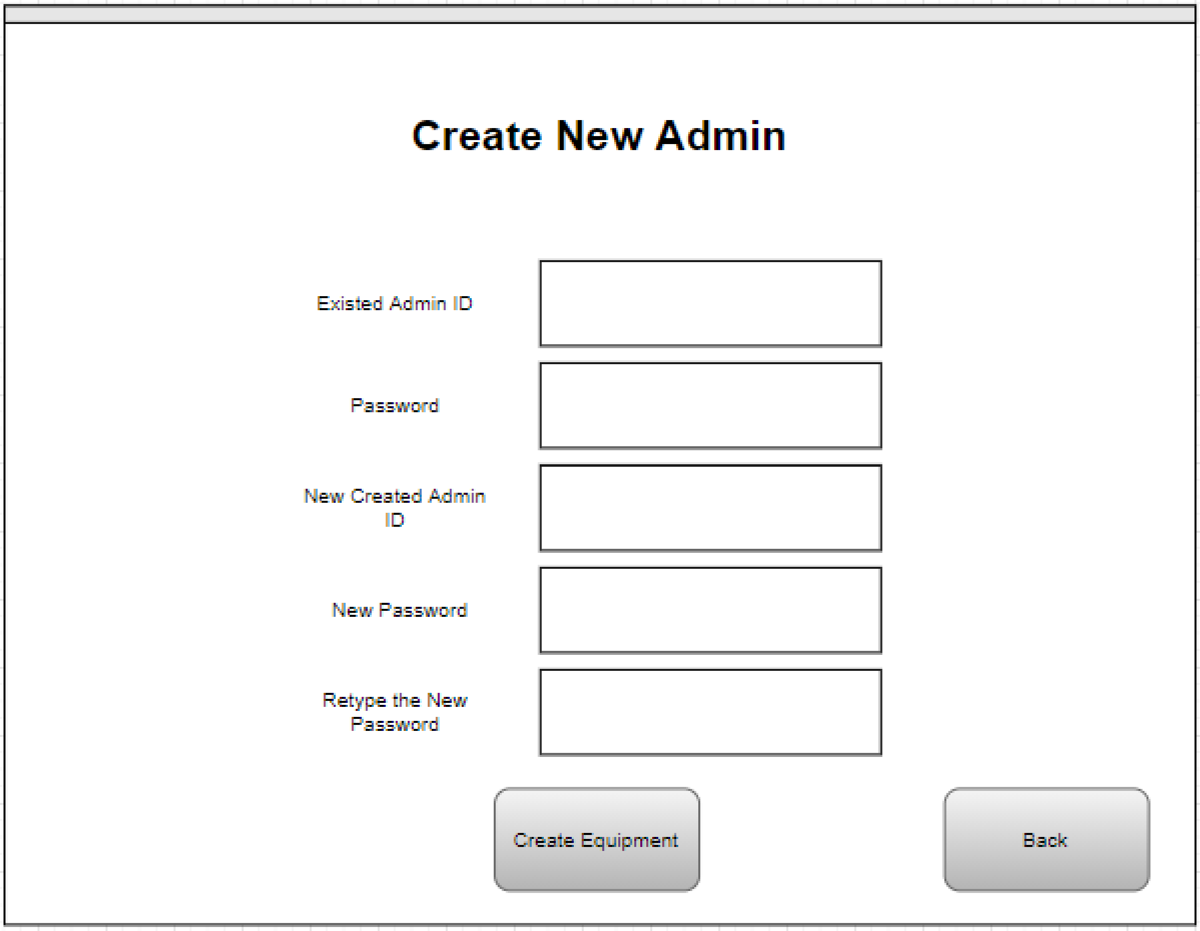
Admin Login



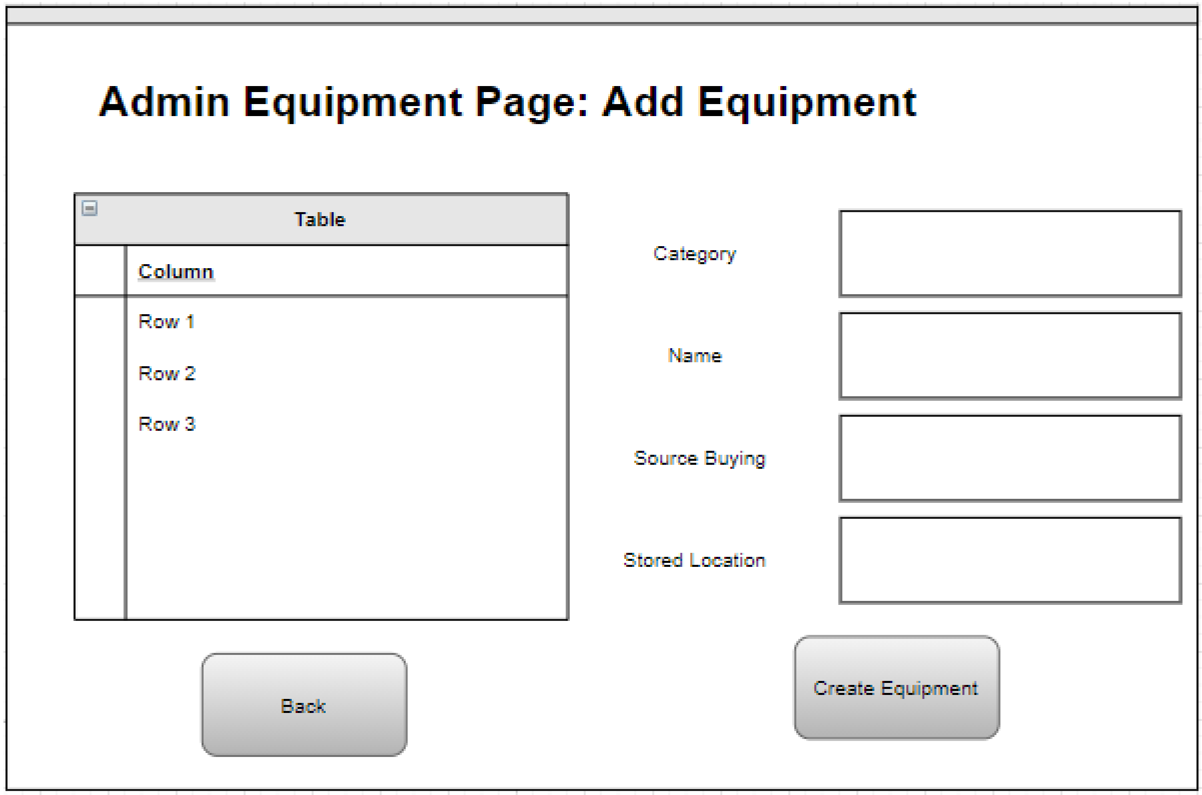
Admin Main Page


Admin Equipment Page
Add Equipment
Delete Equipment
Change Equipment Status
Back


Borrow Record
Table
Column
Row 1
Row 2
Row 3
Back

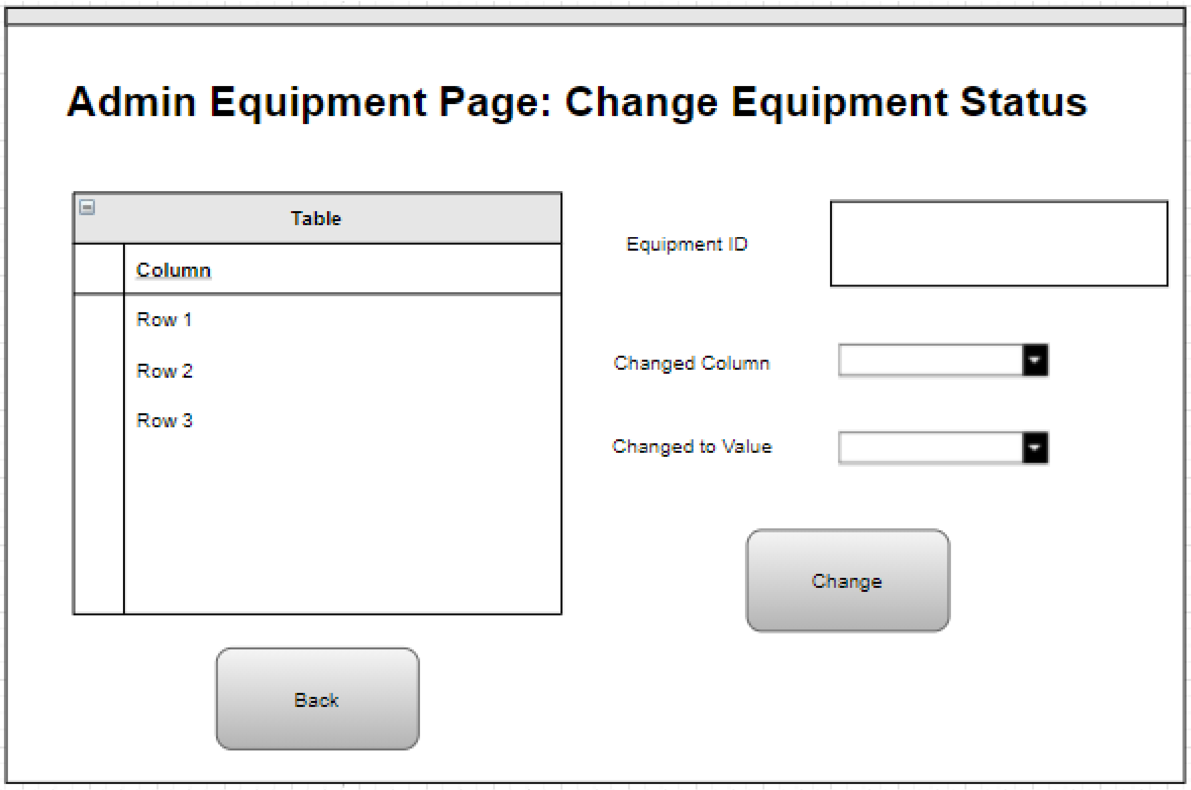


Create New Admin

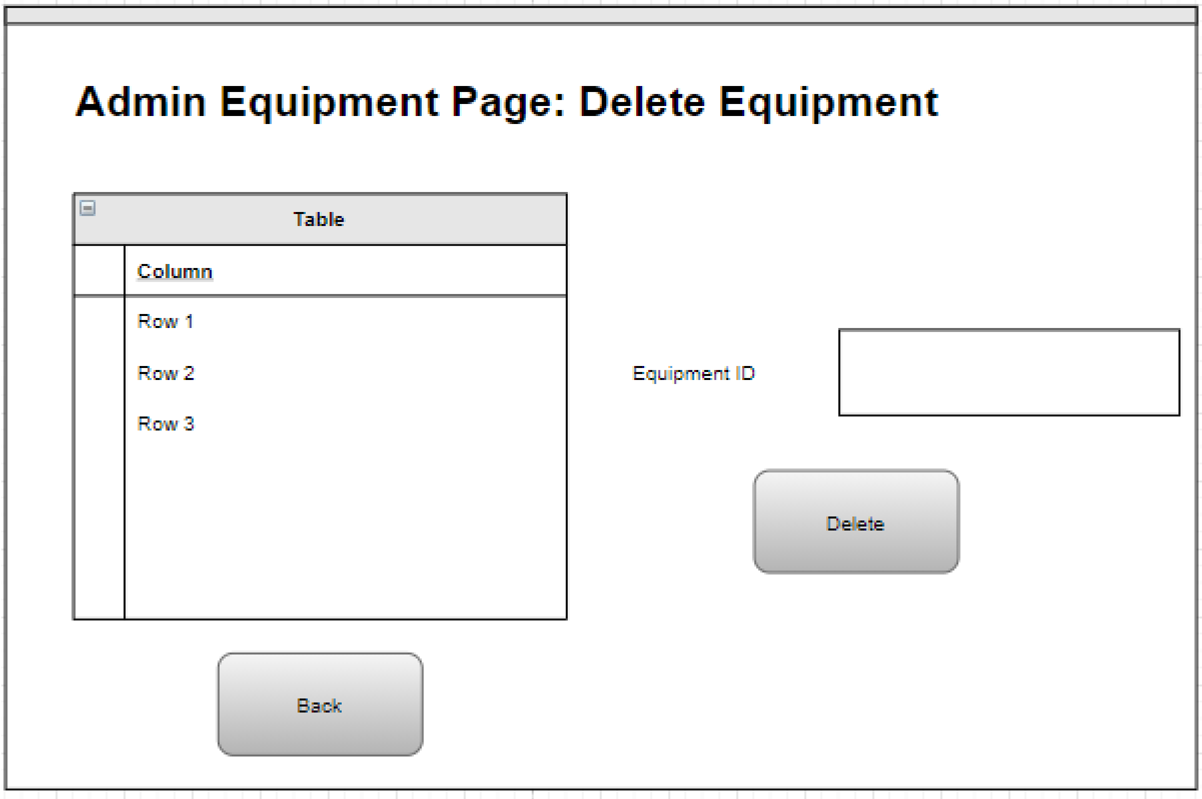


Admin Equipment Page: Add Equipment

Table
Column
Row 1
Row 2
Row 3



Admin Equipment Page: Change Equipment Status



Admin Equipment Page: Delete Equipment

Test Plan

Action to be Tested	Test method
Have a database to store all the records permanently.	Check whether there is a database. Whether updating the program in the appropriate

	position can change the value in the database. Check whether after closing the program and re-open it, the changed value still exists. The criteria would be met if the above statements are valid.
Allow users and admin access to different pages in a clear and easy way	Run the program, try whether it can go through different pages in a clear and easy way. The criteria would be met if there are very less buttons and may have a main page which connects most of the pages in both user and admin side.
Allow admin to check the items in the equipment page anytime when it was checked	Run the program, check the equipment page. Then try to make some changes in the user side such as borrow an item, then check the equipment page whether the equipment’s state is set as borrowed. Also check after an equipment being returned whether the admin equipment page was correct reflecting the scenario. The criteria would be met if the equipment page is reflecting the scenario.
Have sorting functions provided in the equipment page	Run the program. Check the equipment page. There should be a table, check whether pressing the column header would be able to sort the equipment in alphabetical or numerical order. The criteria would be met if it can do so.
Enable users to borrow equipment Each equipment can only be borrowed by one person Each person can only borrow a maximum of two items	Run the program. Enter the borrow page. Check if the user satisfy the following criteria could borrow an item: 1. type in a valid user name and password 2. type in the equipmentId of an equipment that is “open to borrow” but “not being borrowed yet” 3. the user has the borrow authorization right 4. the user didn’t borrow more then two items Also check if the user didn’t satisfy any of these criteria whether he could still borrow.

	I can confirm whether the user have typed in the valid user name and password, have the borrow authorization right and whether the user have borrow more then two items by checking the usermaster from the database. I can confirm whether the equipment is having the state of “open to borrow” but “not yet being borrowed” by checking the equipment master. The criteria would be met if the user can borrow with satisfying all 4 criteria and can’t if any of them doesn’t met.
Enable users to return equipment	Run the program. Enter the return page. Check if a user borrowed an item, whether or not he can return the item by typing in a correct equipmentId and his userId which he uses when he borrows the equipment. Check if the user didn’t type in the correct equipmentId that is borrowed by that user, he couldn’t return the item. I can confirm whether he successfully return the item by checking the equipmentmaster from the database or directly looking at the equipment table on the user side or the admin side which is projecting the database. The criteria is met if all of them is satisfied.
Enable users to extend the date of return	Run the program. Check if a user can extend the borrowing time if he satisfied the following

	criteria: 1. by typing in a correct equipmentId and his userId which he uses when he borrow the equipment 2. if he didn’t return late (this can be checked by seeing if the return date passes the “dateEnd” in the borrowrecord table in database) 3. He extends less than the times of extension the equipment is set (default to be 2) I can confirm whether he return late by checking if today’s date passes the “dateEnd” in the borrowrecord in database. I can also confirm whether he extends less than the times of extension by checking the “numberOfExtensionUsed” in the borrowrecord. Check if any of the criteria does not satisfy, can the user still extend the date The criteria would be met if the user can borrow with satisfying all 3 criteria and can’t if any of them doesn’t met.
Have a feedback function for users	Run the program. If the user can provide a feedback in the return page and the teacher receives it. the criteria would be met.
Have a question function for users	Run the program. If the user can provide a feedback in the question page and the teacher receives it. the criteria would be met.
Allow admin to easily notice the question ask by users	Run the program. If the admin can check the questions very easily and in a very obvious location (such as the admin main page), the structure of the questions listed in the list is clear too, then these criteria would be met.

Allow admin to know the borrowing records anytime when it was checked	Run the program, check the borrowrecord page. Then try to make some changes in the user side such as borrow an item, then check the borrowrecord page whether a new record is created and whether different details in it are correct. Also check after an equipment being returned whether the borrowrecord page was correct reflecting the scenario. The criteria would be met if the borrowrecord page is reflecting the scenario.
Allow admin to add / delete items in the equipment table	Add item: Run the program, enter the add new equipment page. After filling in all the information and add an equipment, enter the equipment page and see whether there is a new item added. The criteria would be met if there is a new item added. Delete item: Run the program, enter the equipment page. After entering the equipmentId and press the button, check if the equipment with the correspond equipmentId is deleted. The criteria would be met if that equipment is deleted.
Allow admin to edit status of items weather it is opened or not opened for borrowing	Run the program, enter the equipment page. Check weather entering a valid equipmentId, type in “currentStatus” and “open to borrow” or “not opened to borrow”. After typing these in, check whether the status of that column has changed. The criteria would be met if the status have changed.
Provide error boxes automatically if there are any data entry errors.	Check all functions in all pages, type in something that are not supposing correct, there should be a Pop Up box which tells you what is wrong with your entry. If all trials of entering wrong data entry and have a pop-up

	box examining the correct reason of why not functioning, the criteria would be met.
Provide a clear and easy instructions to guid users in how to use the program.	Run the program. Enter the welcome page. There should be clear instructions with all written in point-form. It should also mention all of the points that need the user to beware of. The criteria should be met if these are done
Allow admin to create new admin	Run the program, enter the add new admin page. Check weather after filling in all the information with 1. Entering a valid adminId and password that is currently existing 2. Entering a new adminId that is not being used and are all integers 3. The password and the retyped password are the same and are all integers The new admin should be created. To check this, visit the adminmaster in the database, if a new admin is created with the adinId and password specified, this criteria is met.
Not allowing users to borrow new items if return late 2 times	Run the program, enter the borrow page. If a user have returned late two times, it should not able to borrow, his borrow authorization should be set as “not allow”. Check if the user can borrow any more items, if he can’t, the criteria is met.
Allow admin to re-open users borrow authorization	Run the program. Enter the user page. If the users borrow authorization is set as “not allow”, after typing in the userId and press the button, that user’s borrow authorization should changed to allow. If It does, the criteria Is met.