SOFTWARE DEVELOPMENT OF THE ONLINE EXAMINATION SYSTEM

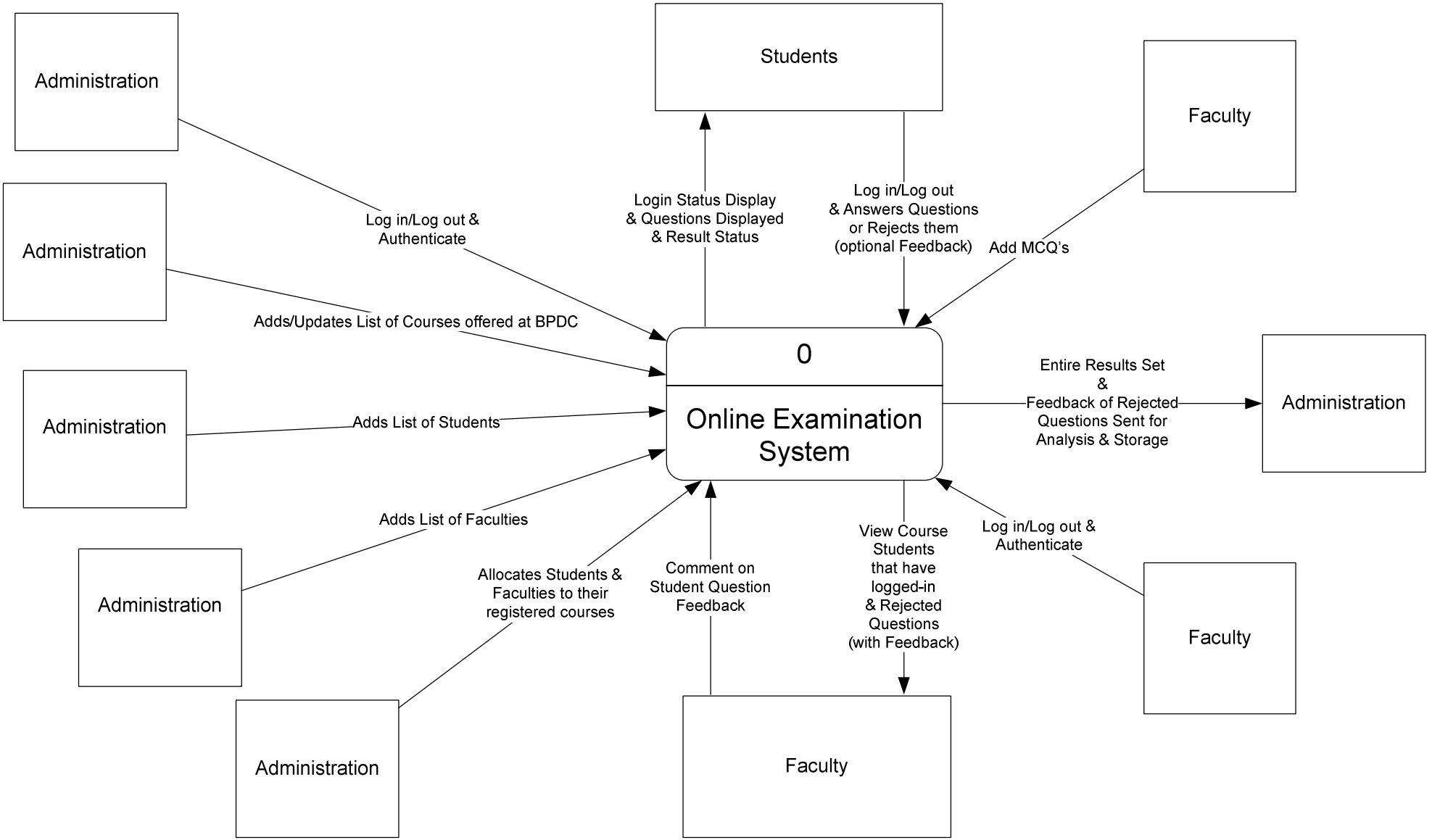
Individual Assignment

Name: Martin Vincent
ID Number: 2001U7PS055
Major: CS Engineering

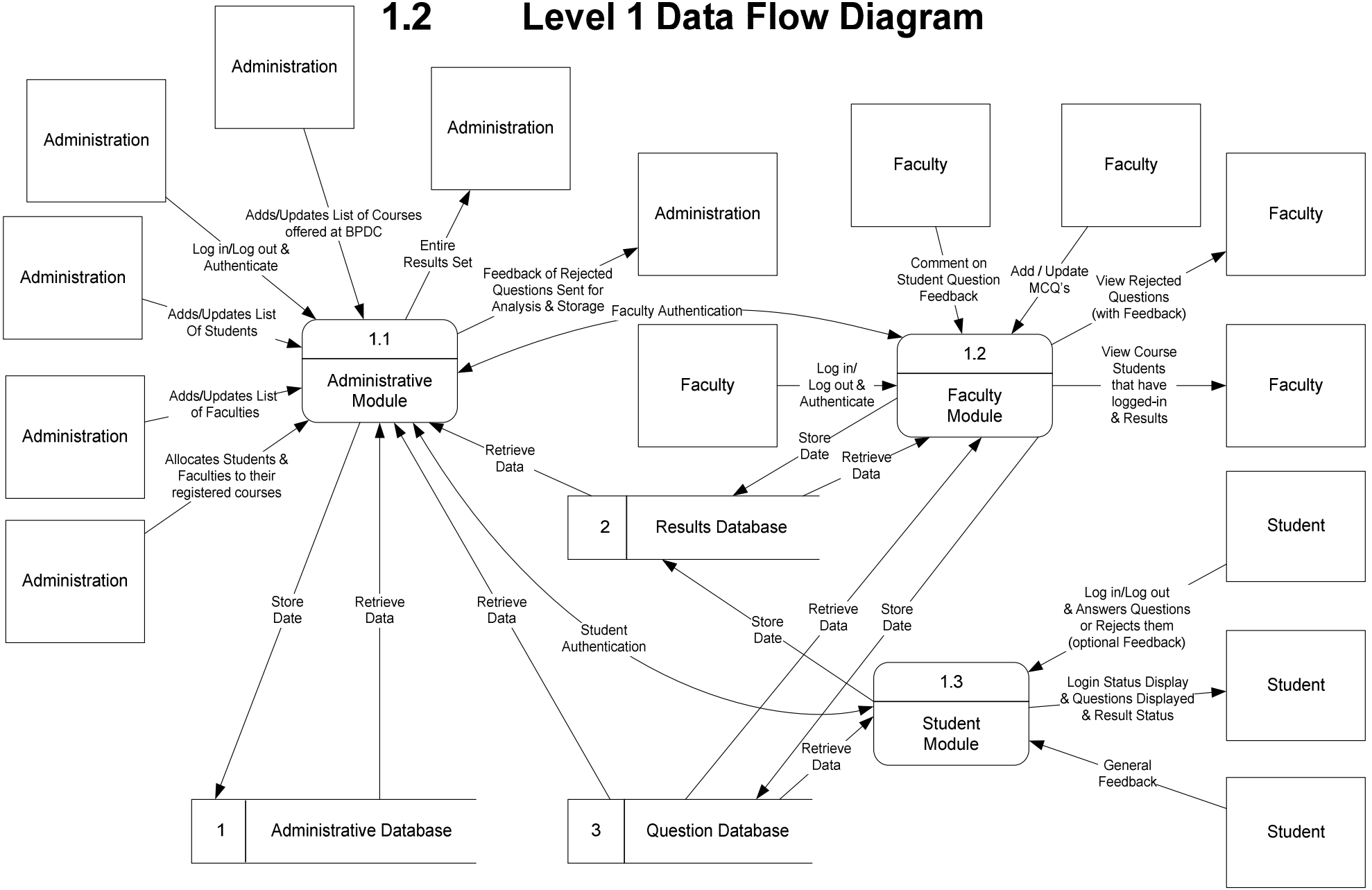
Index

1 Data Flow Diagrams

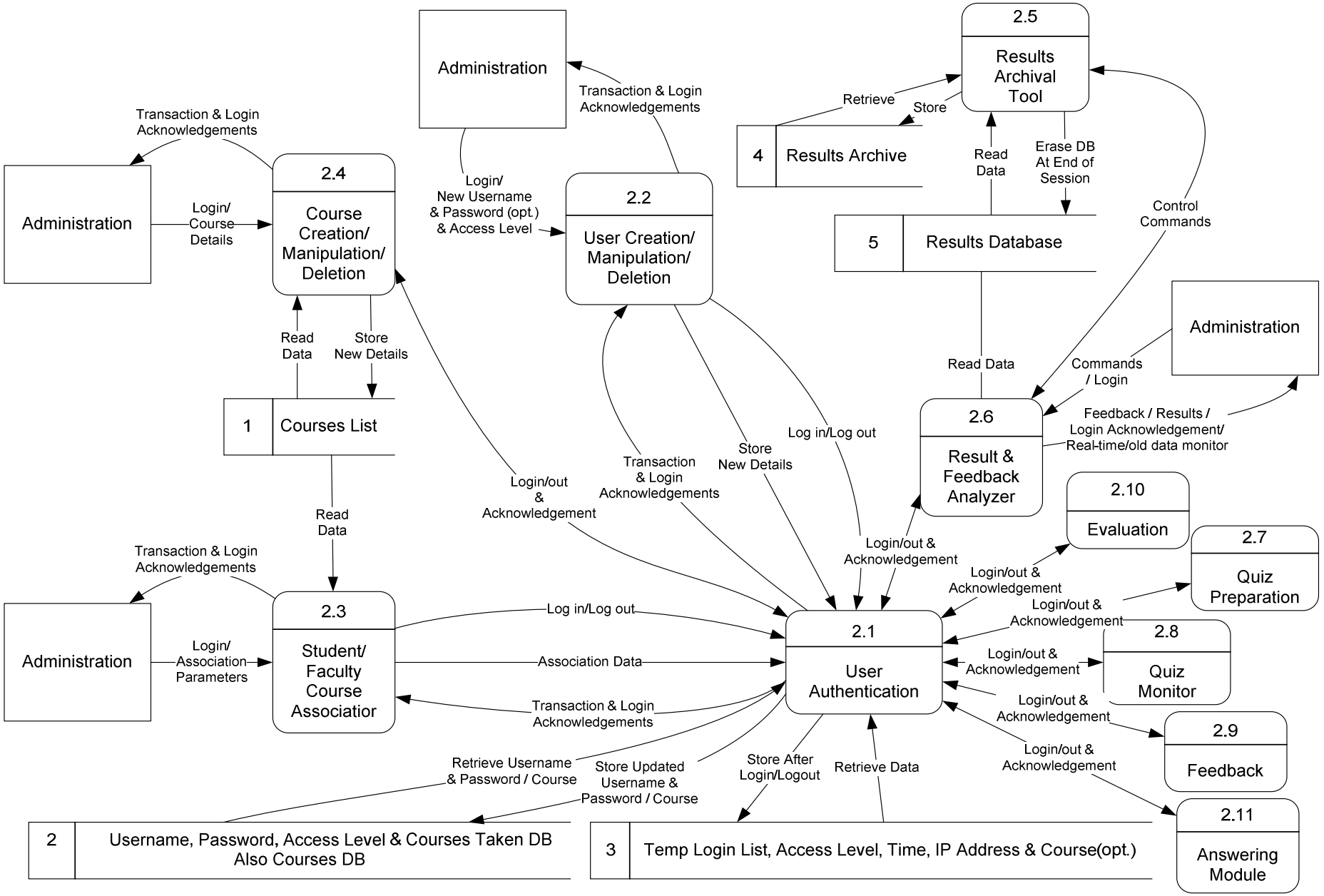
1.1 Context Level Data Flow Diagram / Level 0 Data Flow Diagram



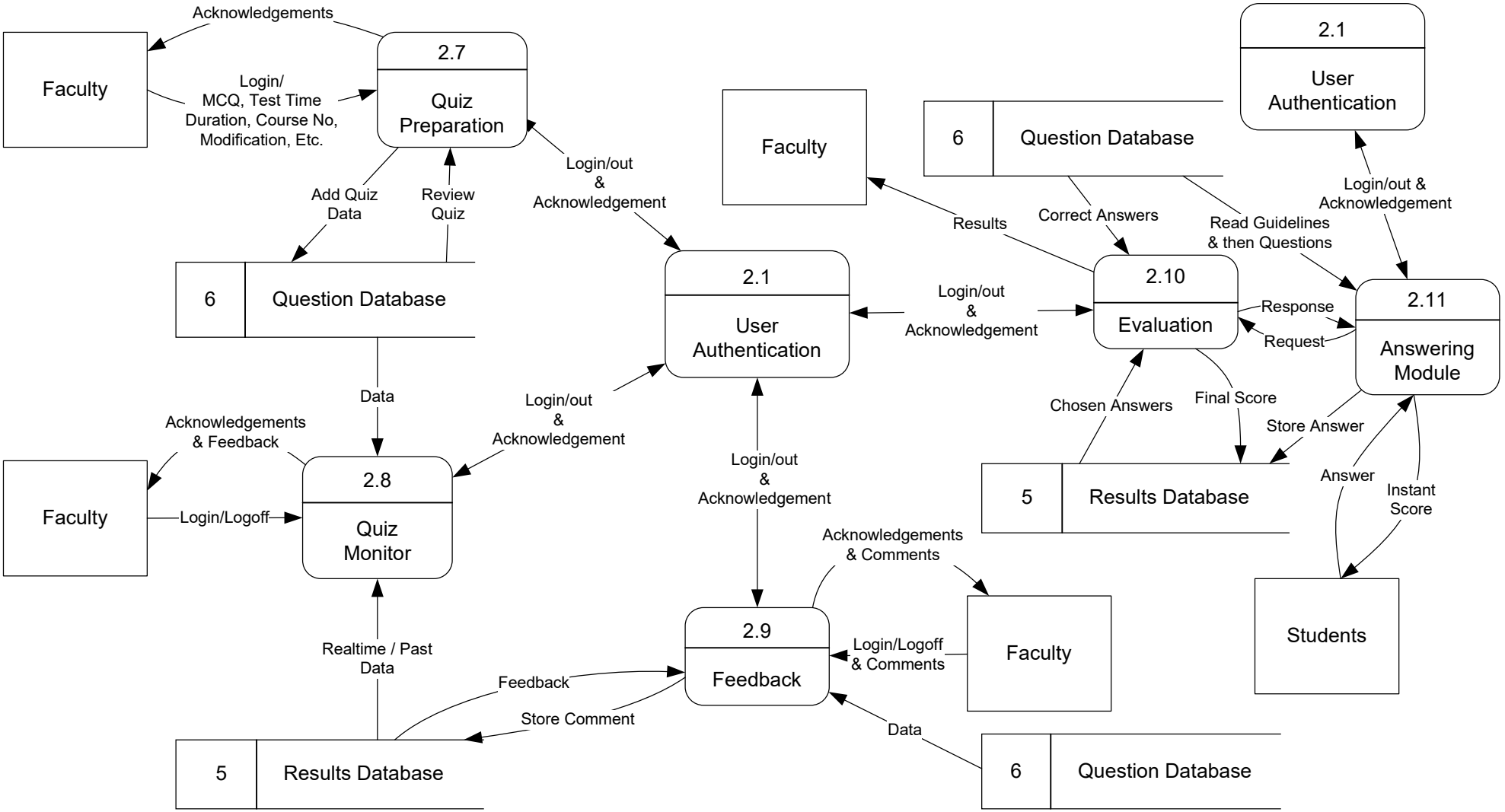
1.2 Level 1 Data Flow Diagram



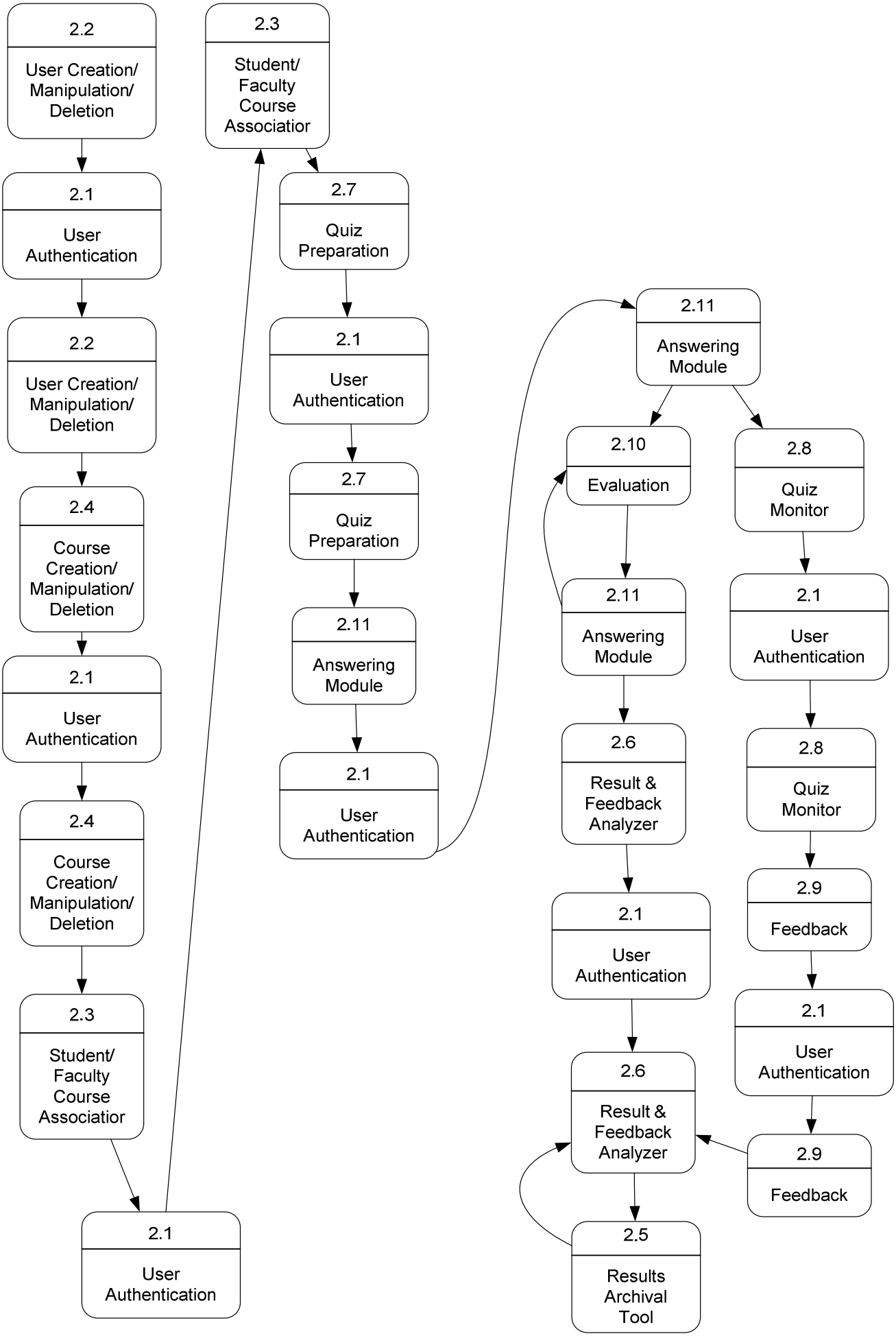
1.3.1 Level 2 Data Flow Diagram for Administration



1.3.2 Level 2 Data Flow Diagram for Faculty & Students



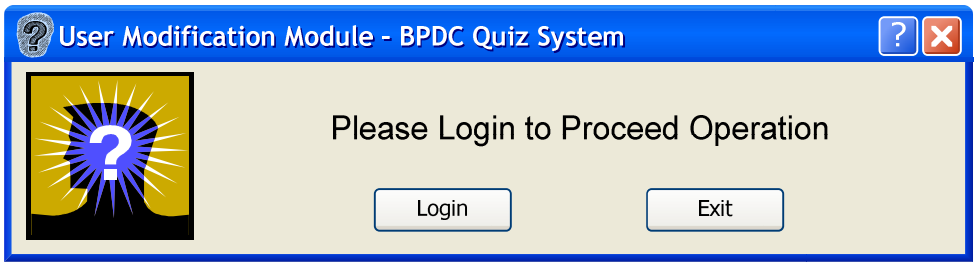
2 Architectural Design

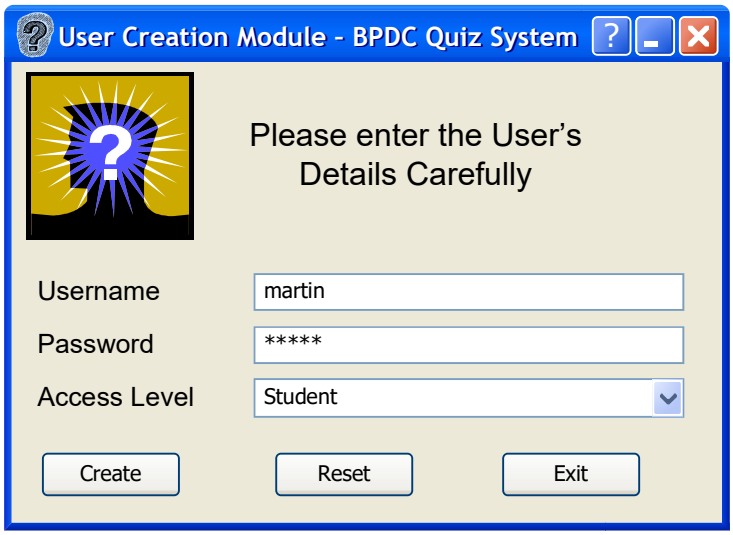

2.2
User Creation/ Manipulation/ Deletion
2.1
User Authentication
2.2
User Creation/ Manipulation/ Deletion
2.4
Course Creation/ Manipulation/ Deletion
2.1
User Authentication
2.4
Course Creation/ Manipulation/ Deletion
2.3
Student/ Faculty Course Associatior
2.1
User Authentication
2.3
Student/ Faculty Course Associatior
2.7
Quiz Preparation
2.1
User Authentication
2.7
Quiz Preparation
2.11
Answering Module
2.1
User Authentication
2.11
Answering Module
2.10
Evaluation
2.11
Answering Module
2.6
Result & Feedback Analyzer
2.1
User Authentication
2.6
Result & Feedback Analyzer
2.5
Results Archival Tool
2.8
Quiz Monitor
2.1
User Authentication
2.8
Quiz Monitor
2.9
Feedback
2.1
User Authentication
2.9
Feedback

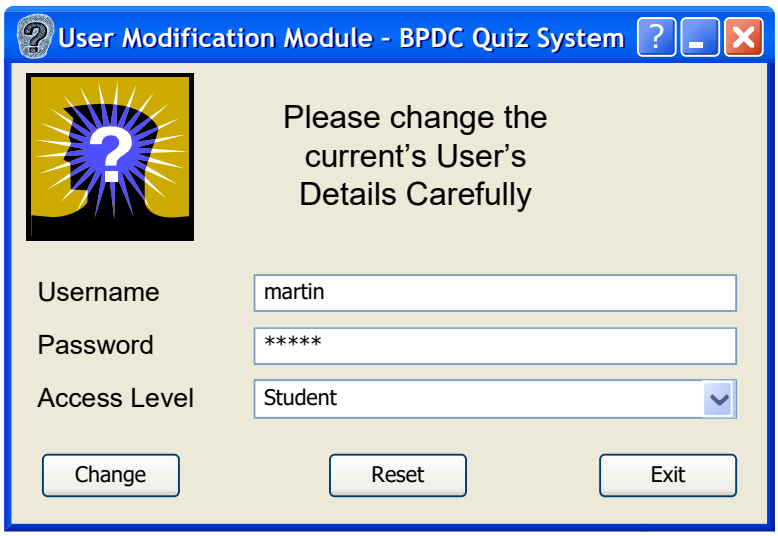
3 User Interface Design

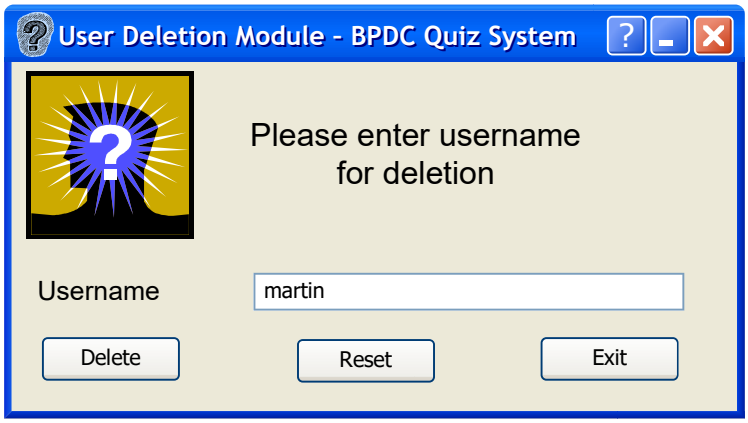
Note: I have rearranged the screen dumps in the architectural design order to facilitate understanding of the operation of this application.

3.1 M2.2 → User Creation/Manipulation/Deletion Module

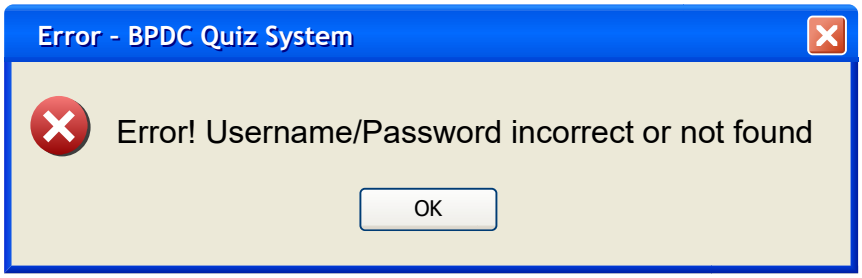




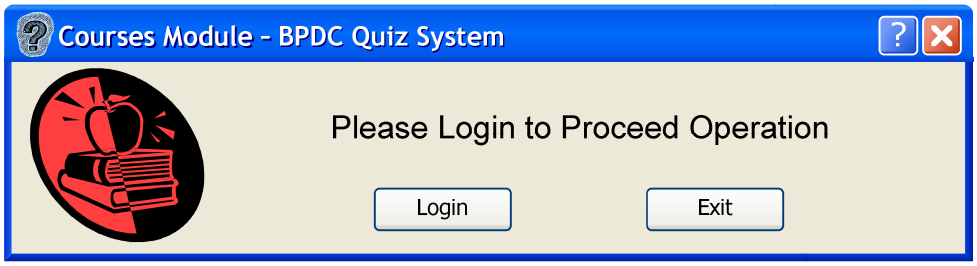


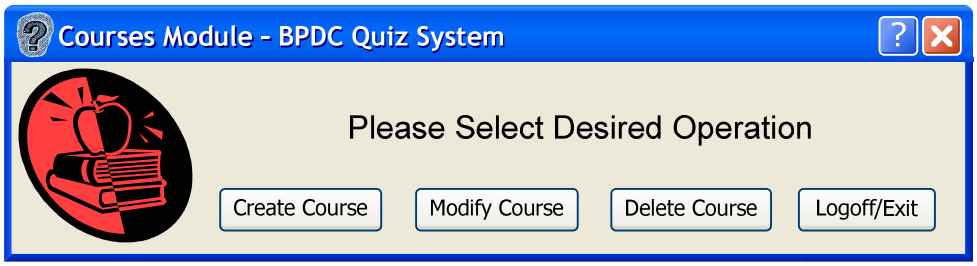


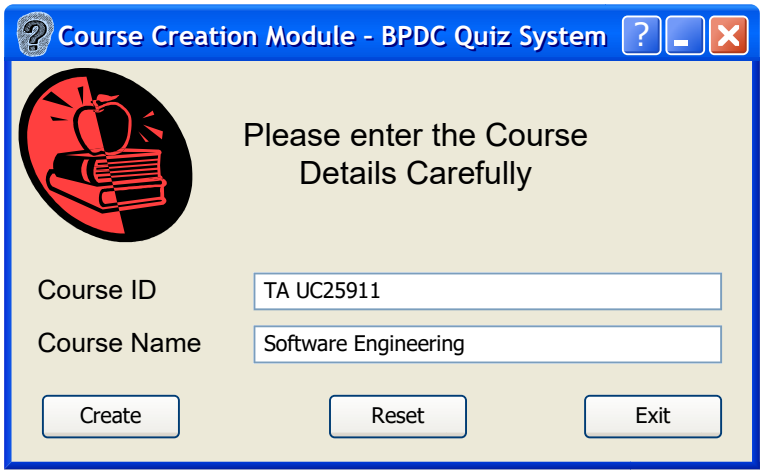
3.2 M2.1 → User Authentication Module

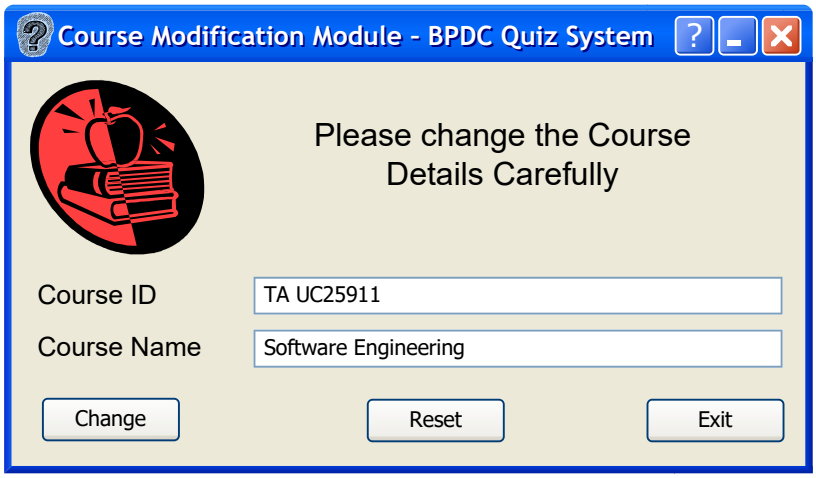


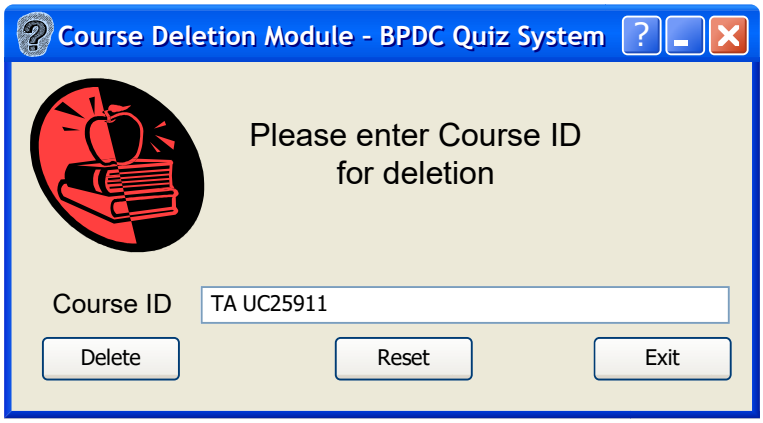
3.3 M2.4 → Course Creation/Manipulation/Deletion Module



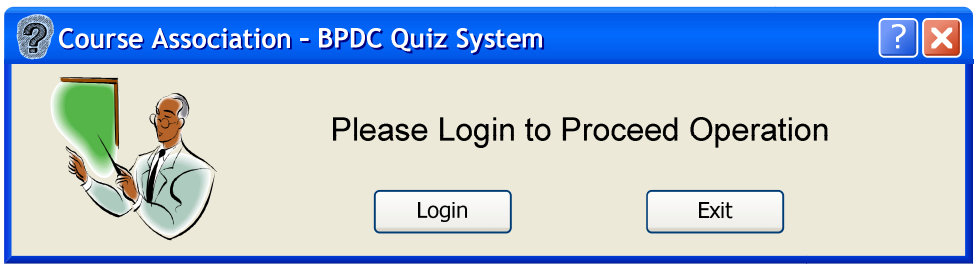


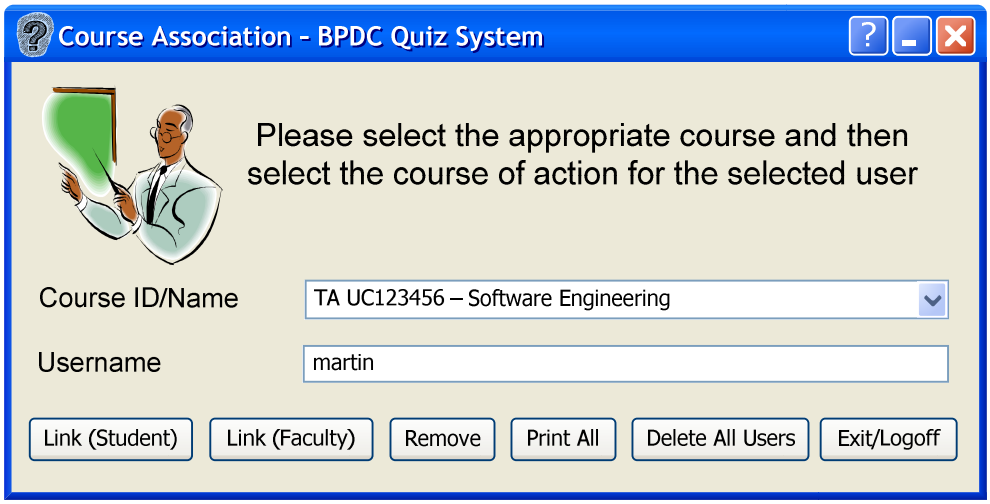




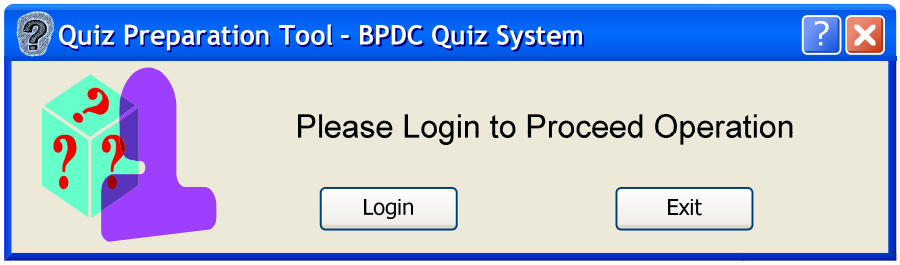


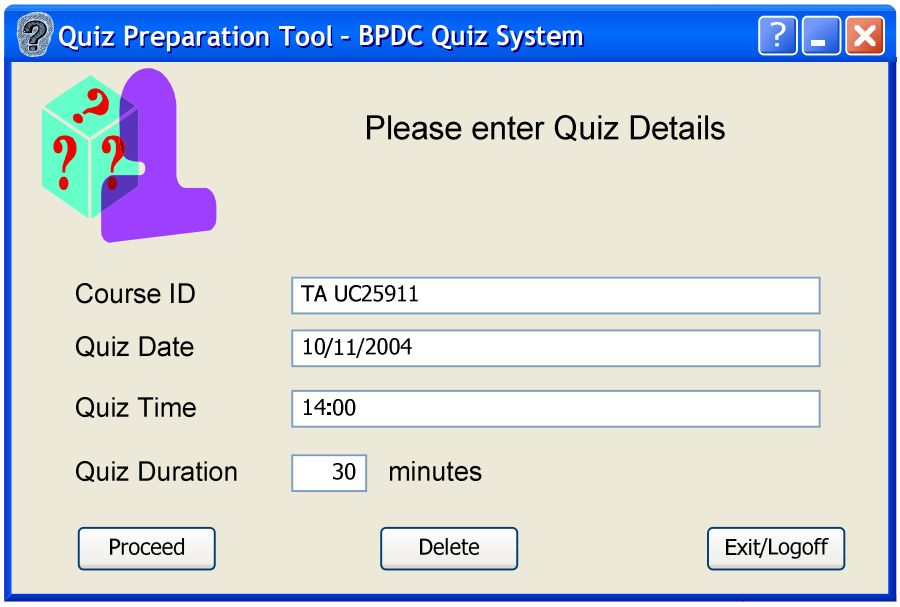
3.4 M2.3 → Student Faculty Course Association Module

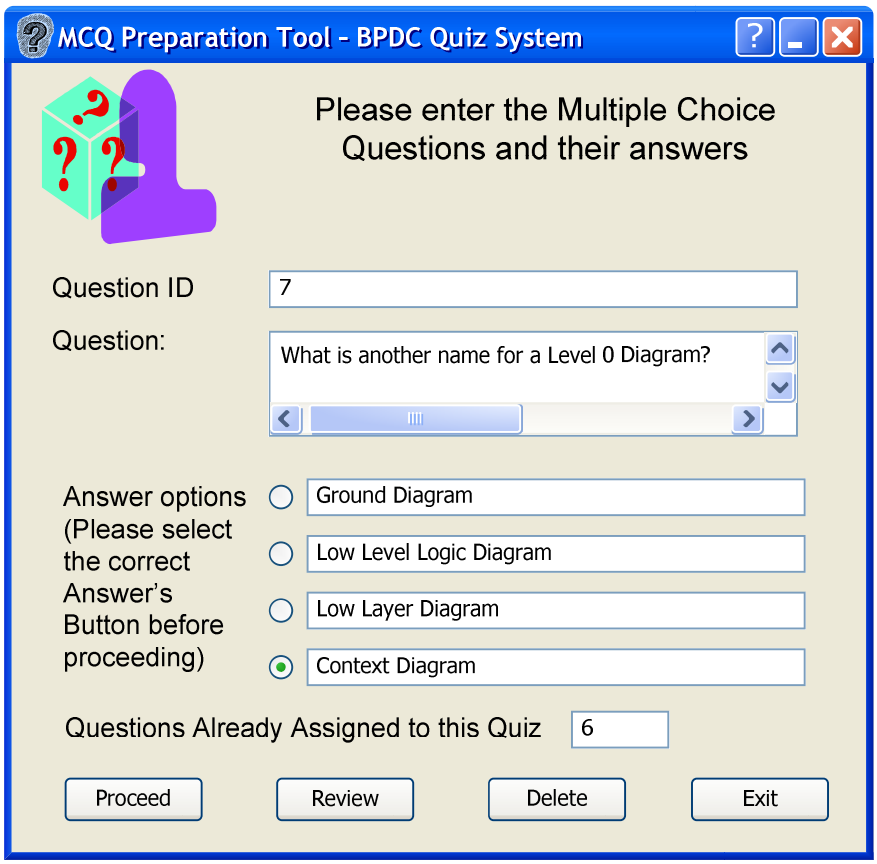




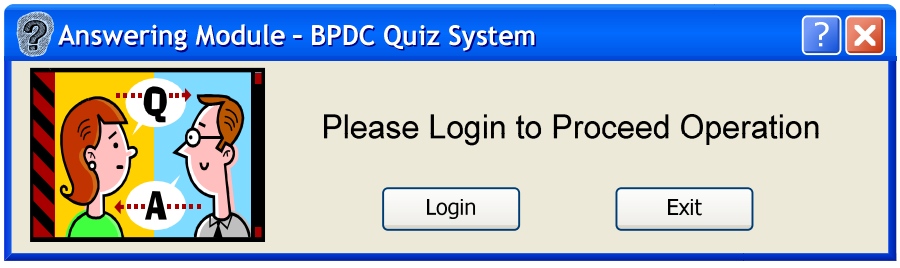
3.5 M2.7 → Quiz Preparation Module

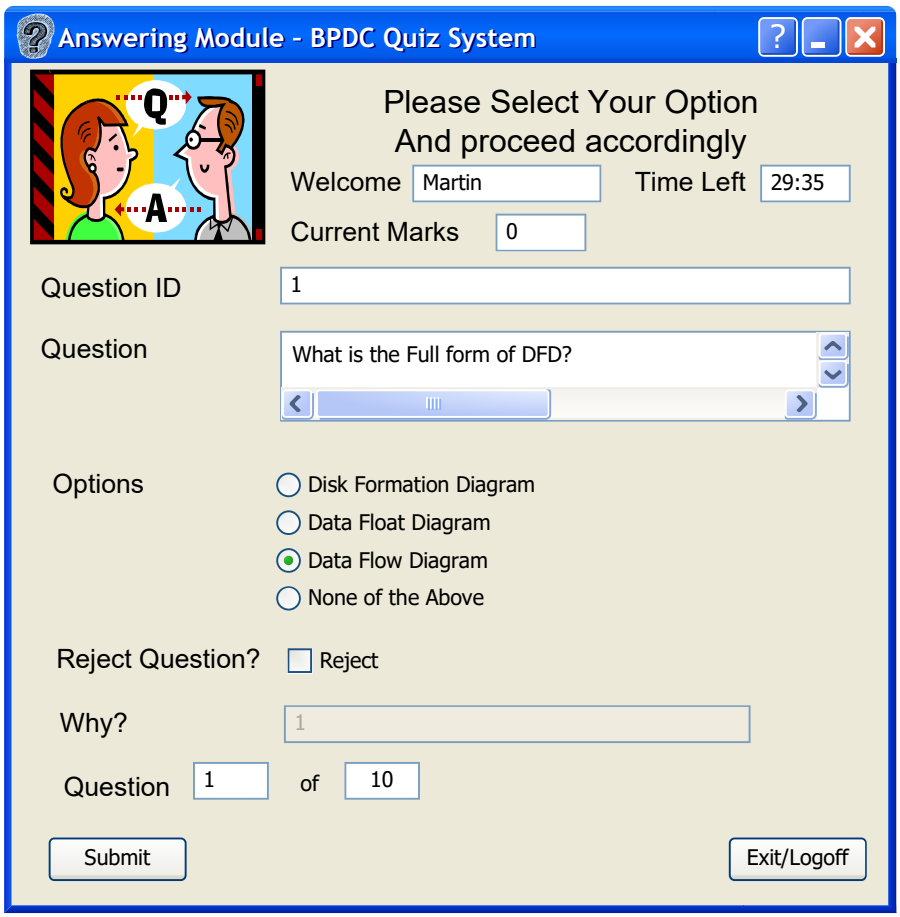




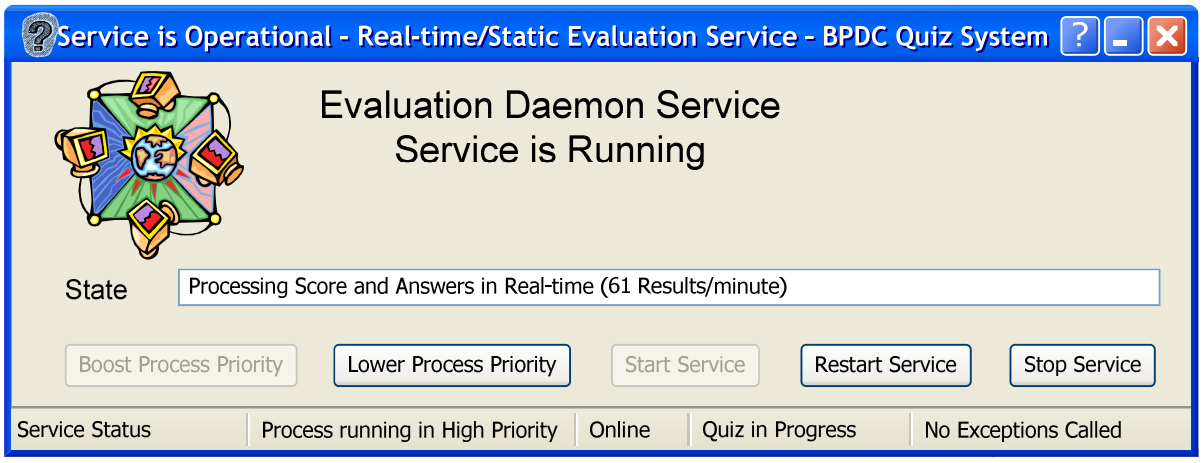


3.6 M2.11 → Answering Module

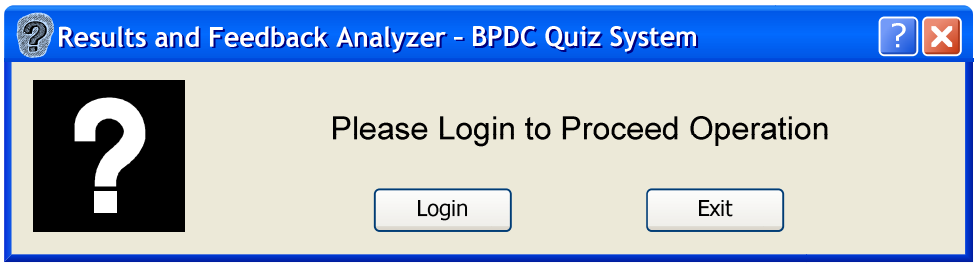


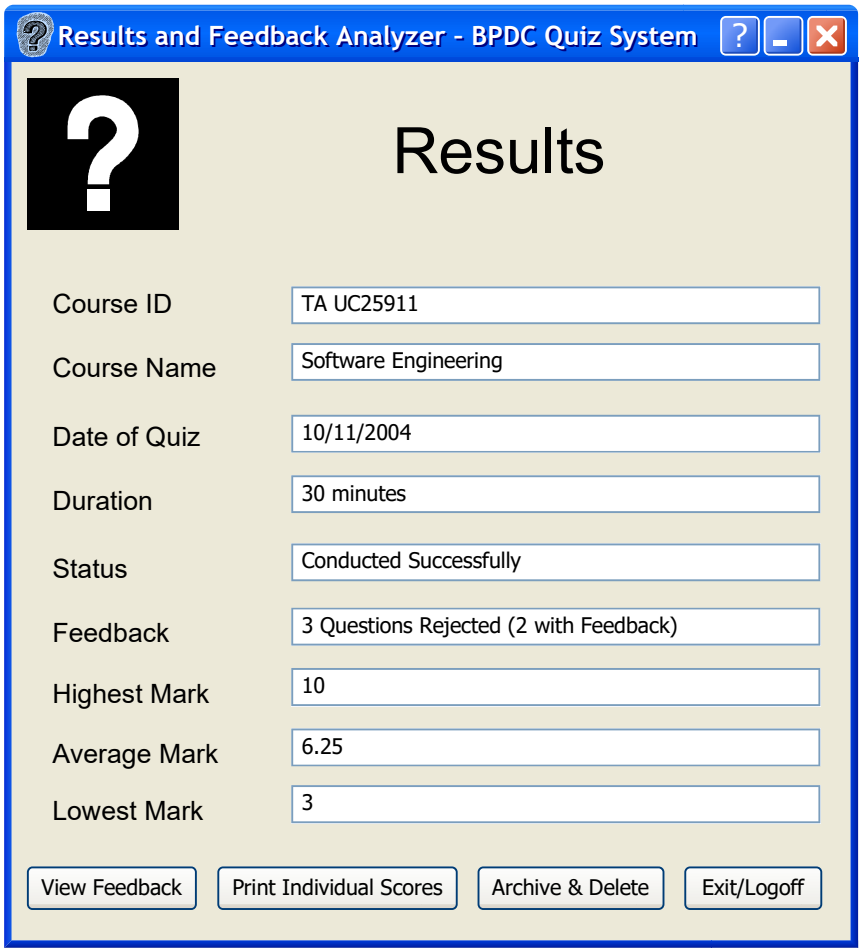


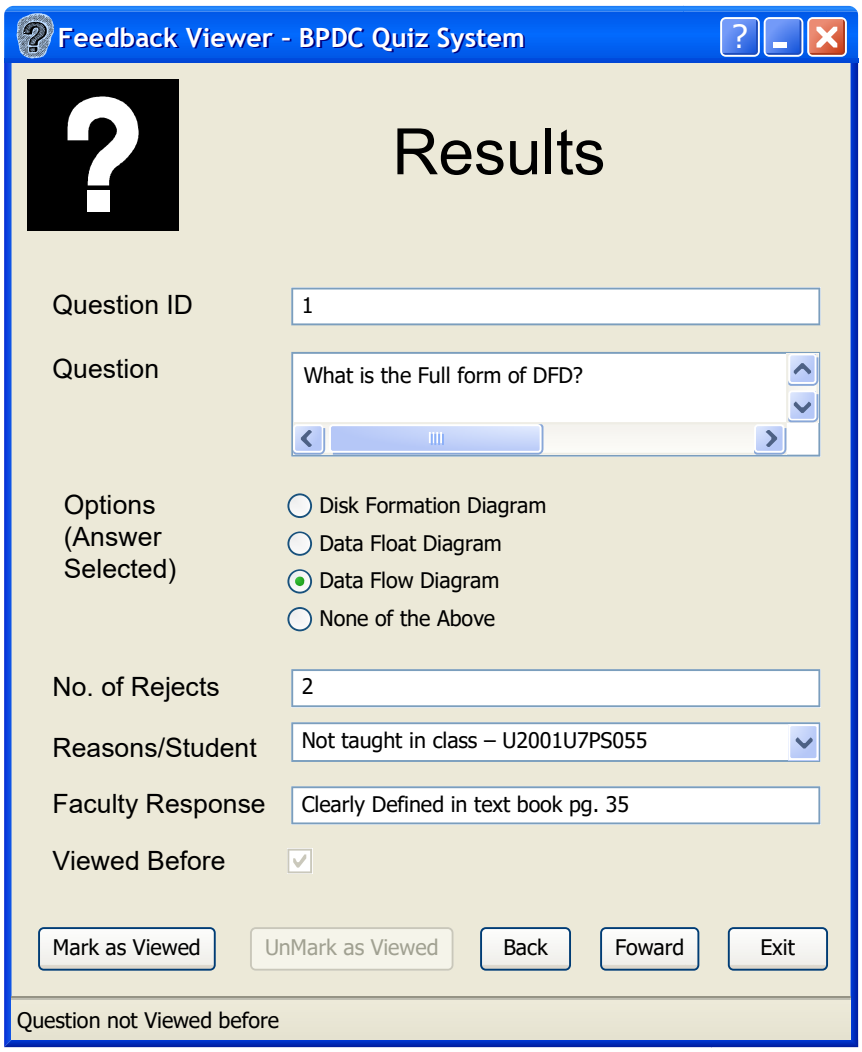
3.7 M2.10 → Evaluation Module



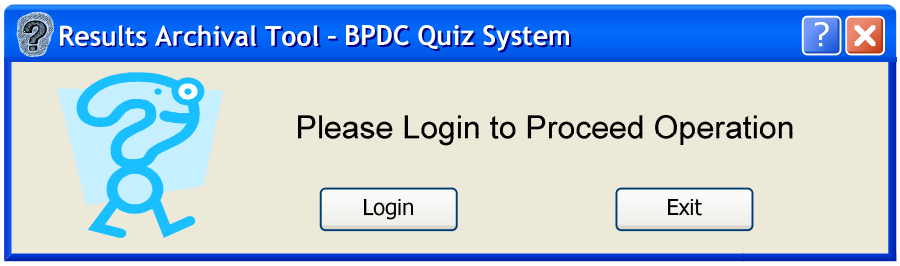
3.8 M2.6 → Result & Feedback Analyzer Module

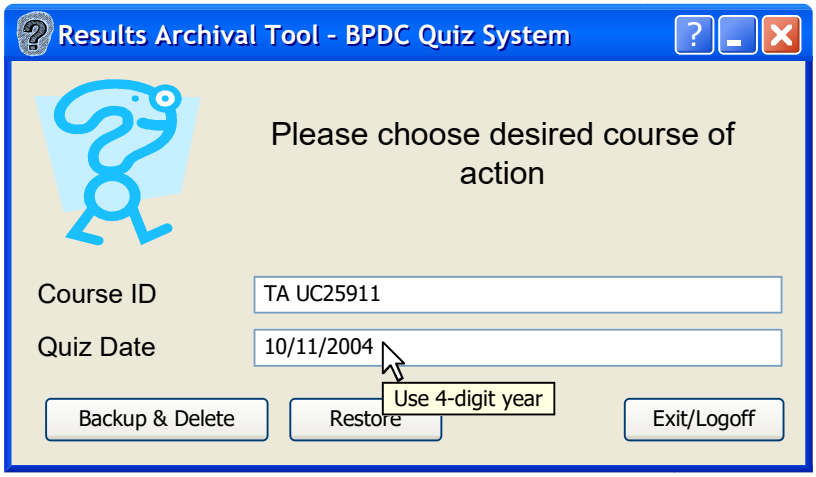




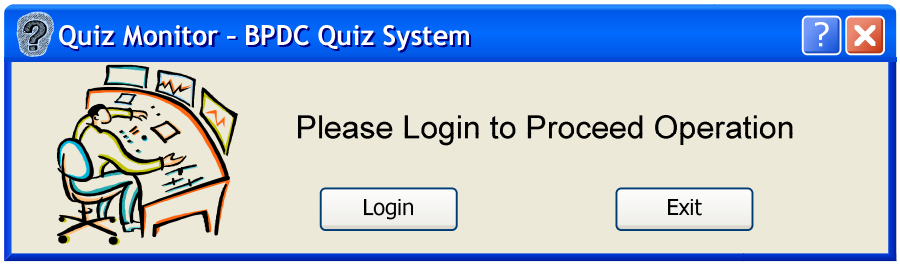

Feedback Viewer - BPDC Quiz System
Results
Question ID
1
Question
What is the Full form of DFD?
Options (Answer Selected)
Disk Formation Diagram
Data Float Diagram
Data Flow Diagram
None of the Above
No. of Rejects
2
Reasons/Student
Not taught in class – U2001U7PS055
Faculty Response
Clearly Defined in text book pg. 35
Viewed Before
Mark as Viewed
UnMark as Viewed
Back
Foward
Exit
Question not Viewed before

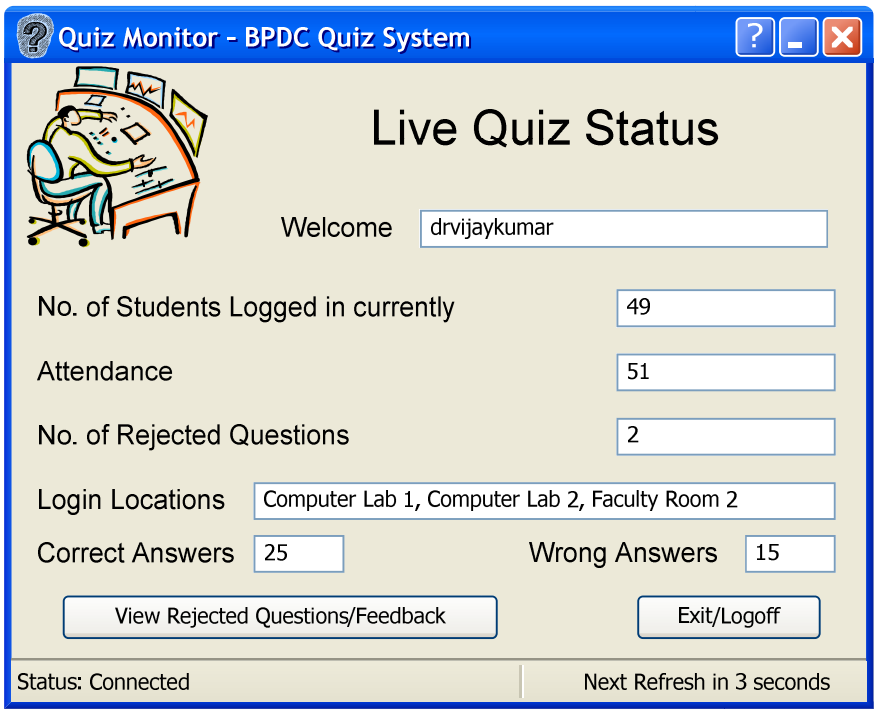
3.9 M2.5 → Results Archival Module

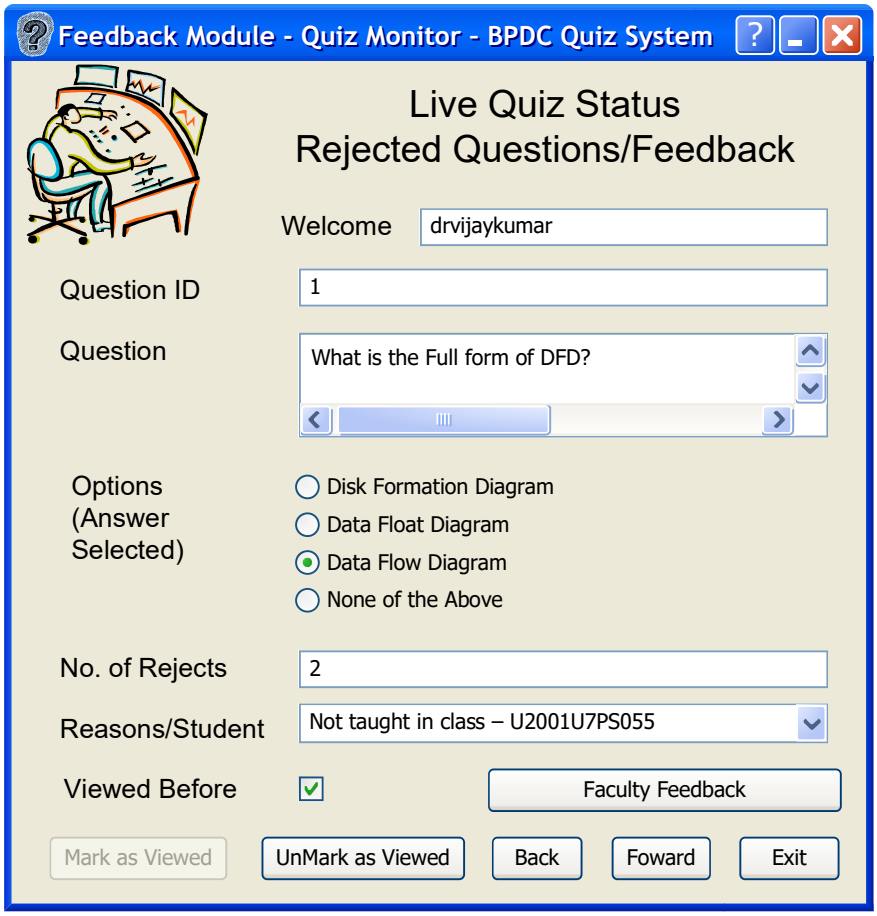




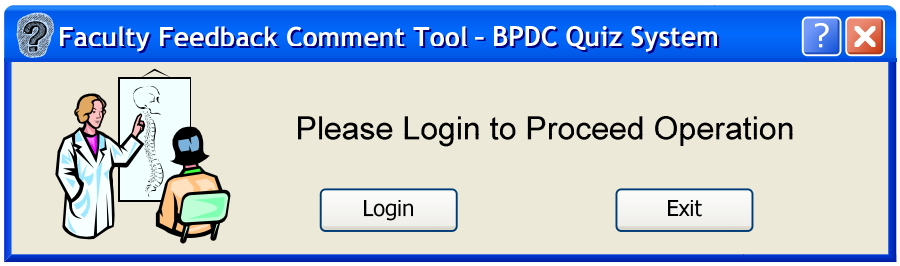
3.10 M2.8 → Quiz Monitor Module

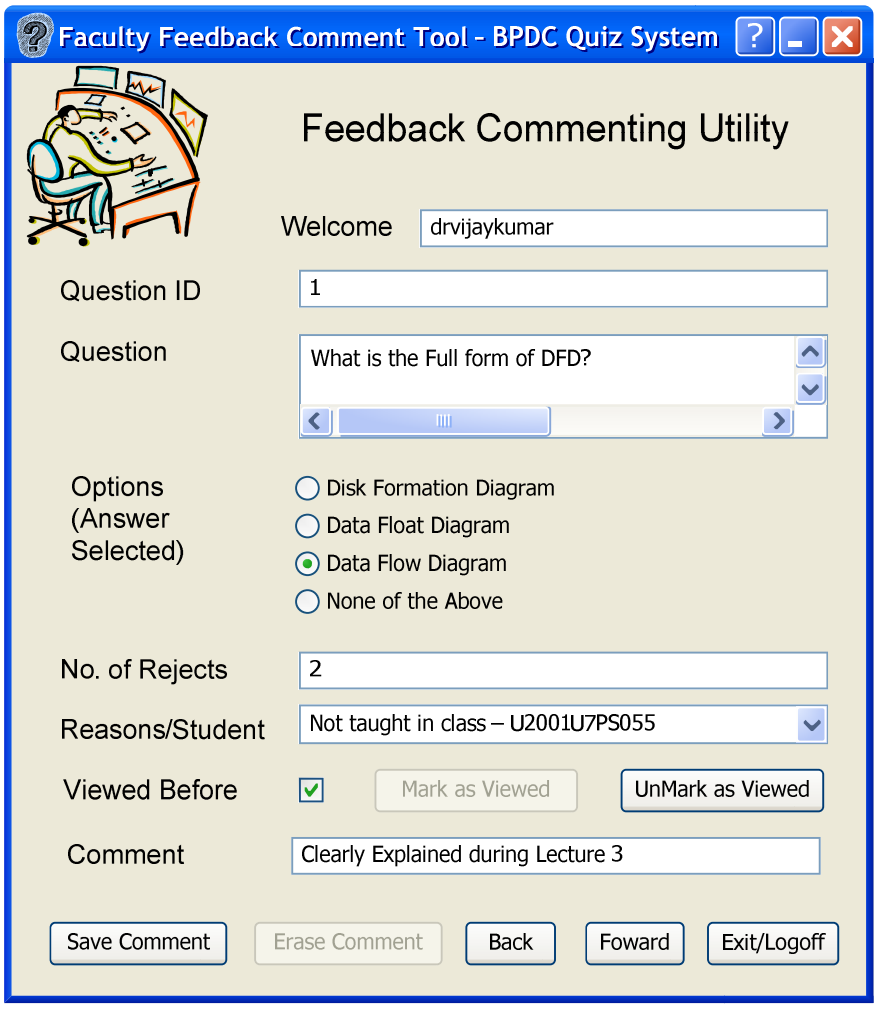



Quiz Monitor - BPDC Quiz System
Live Quiz Status
Welcome
drvijaykumar
No. of Students Logged in currently
49
Attendance
51
No. of Rejected Questions
2
Login Locations
Computer Lab 1, Computer Lab 2, Faculty Room 2
Correct Answers
25
Wrong Answers
15
View Rejected Questions/Feedback
Exit/Logoff
Status: Connected
Next Refresh in 3 seconds


Feedback Module - Quiz Monitor - BPDC Quiz System
Live Quiz Status
Rejected Questions/Feedback
Welcome
drvijaykumar
Question ID
1
Question
What is the Full form of DFD?
Options (Answer Selected)
Disk Formation Diagram
Data Float Diagram
Data Flow Diagram
None of the Above
No. of Rejects
2
Reasons/Student
Not taught in class – U2001U7PS055
Viewed Before
Faculty Feedback
Mark as Viewed
UnMark as Viewed
Back
Foward
Exit

3.11 M2.9 → Feedback Module





4 Component Level Design

Note: *I have rearranged the component data design segments in order to facilitate understanding of the operation of this application.*

4.1 Module 2.2 – User Creation / Manipulation / Deletion

Inputs: Username, Password, Access Level

Outputs: Username, Password, Access Level

Description: Helps provide front-end for storing/updating/deleting username, password and/or access level

Logic:

Login and verify user else terminate session
If Create is chosen then
    Accept user data
    Check if user is already present
        If yes abort with an error message
        If not proceed
    Send data to module 2.1 User Authentication in encrypted form
    Wait for Acknowledgement to verify operation's success
If Update is chosen then
    Request existing user information
    Accept updated data
    Check if user is present
        If not abort with an error message
        If so proceed
    Accordingly update system with new information with M2.1 in encrypted form
    Wait for Acknowledgement to verify operation's success
If Deletion is chosen then
    Check if user is exists in the system
        If not abort with an error message
        If so proceed
    Request Module 2.1 to delete user
    Wait for Acknowledgement to verify operation's success
Logout at the end of session

Associations with:

1. Module 2.1 User Authentication

4.2 Module 2.1 – User Authentication

Inputs: User Information
- Username
- Password
- Access Level
- Remote Procedure

Course Information
- Course ID
- Course Name

Courses Association Information
- Course Selected
- Student or Faculty that is associated with the course

Authentication Information
- Which module requested user authentication
- Username
- Password
- Access Level (from Database)

Outputs: User Information
- Username
- Password
- Access Level

Course Information
- Course ID
- Course Name

Courses Association Information
- Course Selected
- Users *(Faculty & Students)* related to the course selected

Authentication Acknowledgement or Error

Logging information of usage information and activity to "**3**" → Log File Store

Description: Acts as an intelligent User Authentication Service

Stores & Recalls usernames and passwords with access level information

Stores & Recalls course information

Stores & Recalls user/course association information

Maintains a log data-store of all activity, who did it and where

Logic: Receive RPC from other modules

If request to authenticate
- Decrypt the username and password
- Check for matching values in the Database and reply if not
- Check to see if access level matches or exceeds the module
- Send result back

If request to add user

Check for existing user in the system and verify admin login
Check for existing user in the system and verify admin login
If yes then reply to sender about this, else
Add new user details
Send acknowledgement back
If modification is needed
Check for user in the system and verify admin login
If not then reply to sender about this, else
Change the data
Send acknowledgement back
If deletion is needed
Check for user in the system and verify admin login
If not then reply to sender about this, else
Delete the record
Send acknowledgement back
If check for user's authentic log in to a service
Check for user in the system and verify in system log file store
If not then reply to sender about this, else
End Process

Associations with:

- **2.2** User Creation / Manipulation / Deletion Module
- **2.3** Student/Faculty/Course Associator Module
- **2.4** Course Creation / Manipulation / Deletion Module
- **2.6** Result and Feedback Analyzer Module
- **2.7** Quiz Preparation Module
- **2.8** Quiz Monitor Module
- **2.9** Feedback Module
- **2.10** Evaluation Module
- **2.11** Answering Module

4.3 Module 2.4 – Course Creation / Manipulation / Deletion

Inputs: Course ID, Course Name

Outputs: Course ID, Course Name

Description: Helps provide front-end for storing/updating/deleting Course ID, Course Name

Logic:

Login and verify user else terminate session
If Create is chosen then
- Accept course data
- Check if course is already present
  - If yes abort with an error message
  - If not proceed
- Send data to module 2.1 User Authentication in encrypted form
- Wait for Acknowledgement to verify operation's success

If Update is chosen then
- Request existing course information
- Accept updated data
- Check if course is present
  - If not abort with an error message
  - If so proceed
- Accordingly update system with new information with M2.1 in encrypted form
- Wait for Acknowledgement to verify operation's success

If Deletion is chosen then
- Check if course is exists in the system
  - If not abort with an error message
  - If so proceed
- Request Module 2.1 to delete course
- Wait for Acknowledgement to verify operation's success

Logout at the end of session

Associations with: Module 2.1 User Authentication

4.4 Module 2.3 – Student Faculty Course Association

Inputs: Course ID, Username

Outputs: A List with courses to choose, A list of Users associated to a course

Description: Helps provide front-end for association of users to courses and viewing the list of users already associated with that course

Logic:

Login and verify user else terminate session
If Link (Student) is chosen then
    Check if user is already associated to the course
        If yes abort with an error message
        If not proceed
    Send data to module 2.1 User Authentication to save data as access level = 'student'
    Wait for Acknowledgement to verify operation's success
If Link (Faculty) is chosen then
    Check if user is already associated to the course
        If yes abort with an error message
        If not proceed
    Send data to module 2.1 User Authentication to save data as access level = 'faculty'
    Wait for Acknowledgement to verify operation's success
If Delete All Users is chosen then
    Send Command to delete all association information related to that course
    Wait for Acknowledgement to verify operation's success
If Removal is chosen then
    Check if user exists in the system
        If not abort with an error message
        If so proceed
    Request Module 2.1 to delete user
    Wait for Acknowledgement to verify operation's success
Logout at the end of session

Associations with: Module 2.1 User Authentication

4.5 Module 2.7 – Quiz Preparation

Inputs: Course ID, Quiz Date, Quiz Time, Quiz Duration, Question ID, Question, Answers with correct solution

Outputs: Question ID, Question, Answers with correct solution and total number of Questions in the Quiz

Description: Helps provide front-end for creation and reviewing (changing existing questions). The purpose of this module is to 'develop' an entire quiz by setting all parameters as well as safely modifying an existing quiz.

Logic:

Login and verify user else terminate session
If Proceed is chosen then
    Check if a quiz is already set at that time
        If yes go to the MCQ Preparation Tool section below
        If not proceed
    Send data to Quiz Database
    Wait for Acknowledgement to verify operation's success
    Open MCQ Preparation Tool
        If Proceed is chosen
            Enter the Question data into the Question DB
            If Question (ID or question data) already exists
                abort with message
            Increase Question count by 1
        If Review is chosen
            Call Question data to front-end application for
                editing
            Send back the edited question to the Question
                Database
            If question is not edited for technical reason give
                error message
        If Delete is chosen in the MCQ module
            Remove all details of the question from the
                question database
            Decrease the question counter by 1
If delete is chosen in the primary module
    Check if question exists and if not give error message
    Delete all questions and quiz details from the Question DB
Logout at the end of session

Associations with: Module 2.1 User Authentication

4.6 Module 2.11 – Answering

Inputs: Question ID, Question, Answers, Question number, total number of questions & Current marks

Outputs: Selected Answer, Rejected? And reason, username, Current Marks

Description: Helps provide front-end for answering questions in a suitable informative environment while providing option to reject with feedback

Logic: Login and verify user else terminate session
At submission
check if an answer was selected
else give error message
Send result or rejection with comment for storage in the results database
Provide Next question till sufficient questions are answered and repeat
Logout at the end of session

Associations with:

1. Module 2.1 – User Authentication
2. Module 2.10 – Evaluation Service

4.7 Module 2.10 – Evaluation Module

Inputs: Answers, Correct Solution, Selected Solution, Username, question number

Outputs: Marks, Results

Description: A system level service that helps service the answering modules running on each students computer by storing the results and calculating & sending the marks back to the answering module

Logic:
Login and verify service else terminate session
Check if quiz is in progress
If yes accept connections from clients
Verify
Identity
Location
Time of Answer
Else terminate connection
Accept answers
Store result in results database
Send instant result back to client
Repeat till quiz ends AND all connections dropped
Logout if service is stopped

Associations with:

1. Module 2.1 – User Authentication
2. Module 2.11 – Answering

4.8 Module 2.6 – Result & Feedback Analyzer

Inputs: Course ID, Course Name, Date of Quiz, Duration, Status, Feedback, Marks, Question ID, Question, Correct Answer, reasons for rejection, faculty response, viewed before

Outputs: Course ID, Highest Mark, Average Mark, Lowest Mark, Question ID, Question, Correct Answer, reasons for rejection, faculty response, viewed before, Print out of student scores

Description: A front end for administration to view feedback of students and the faculty's response, calculate quiz statistics, print out scores of students, and archive the test results and delete.

Logic:
Login and verify service else terminate session
Check if course id has a quiz in Results DB else give error message
Load the information on the main screen from the Databases
Calculate the Highest, Lowest and Average marks and display
If user selects View feedback
    Display the 1st rejected question with
        Usernames of Rejecters and reasons
        Faculty response
    Mark as viewed if the Unmark as viewed is not clicked
    Proceed as Next or Forward as requested or exit feedback mode
If user selects Print Individual Scores then send list of scores to printer
If user selects Archive & Delete then
    Archive the Results of the Quiz in the Results Archive DB
    Delete the quiz information from the Results DB
    OR Vice Versa if the user requests to call back an archive
Logout if user selects the option.

Associations with:

1. Module 2.1 – User Authentication
2. Module 2.5 – Results Archival Tool

4.9 Module 2.5 – Results Archival Tool

Inputs: Course ID, Course Name, Date of Quiz, Duration, Status, Feedback, Marks, Question ID, Question, Correct Answer, reasons for rejection, faculty response, viewed before information

Outputs: Course ID, Course Name, Date of Quiz, Duration, Status, Feedback, Marks, Question ID, Question, Correct Answer, reasons for rejection, faculty response, viewed before information

Description: A back end service that purpose is to archive and retrieve quiz information as per request of the M2.6 - Result and Feedback Analyzer

Logic: Receive the Command from M2.6 – Result and Feedback Analyzer
If archive is the command then
Read data from the Results database
Store all the information in the Results Archival Database
If retrieve is the command then
Read data of the requested quiz information from the Archive
Store the quiz information in the Results Database

Associations with: Module 2.6 – Result & Feedback Analyzer

4.10 Module 2.8 – Quiz Monitor

Inputs: Students logged in, attendance, rejections, login locations of users, number of wrong and correct answers, question id, question information with answer and reason of rejection, viewed? information

Outputs: Faculty's feedback

Description: A front end user monitoring service to view online real-time statistics with faculty feedback module implemented to respond to student feedback

Logic:
Login and verify user else terminate application
Scan the Results DB for any Quiz in Process
    If no Quiz is in progress then
        Display the last quiz related to that faculty login
    If a Quiz is in progress then
        Read data & calculate variables needed to be displayed
        Lookup location of login based on ip addres range
If View Rejected Questions/Feedback is selected then
    Display the 1st rejected question with
        Usernames of Rejecters and reasons
        Question ID
        Question
        Answer with solution
    Mark as viewed if the Unmark as viewed is not clicked
    If Faculty Feedback is selected then invoke the module
    Proceed as Next or Forward as requested or exit feedback mode
Refresh on a time period of 1 second
When user selects Exit/Logoff terminate session and monitoring

Associations with
1 Module 2.1 – User Authentication
2 Module 2.9 – Feedback (non-data association)

4.11 Module 2.9 – Feedback

Inputs: Question ID, Question, Answers with correct option, student feedback with student username, viewed before mark

Outputs: Faculty comment with question ID

Description: A front end for faculty to post their comments on the student feedback after viewing them in the module. Also reject count is calculated to indicate intensity of rejection.

Logic: Login and verify faulty else terminate application
Display the 1st rejected question with
- Usernames of Rejecters and reasons
- Question ID
- Question
- Answer with solution

Calculate the number of rejects of that question and display
Mark as viewed if the Unmark as viewed is not clicked
If comment is present display it
If Save Comment is clicked Save it in the Results Database
If Erase Comment is clicked then delete the comment
Traverse forwards and backwards with the Back & Forward buttons
When user selects Exit/Logoff terminate module

Associations with

1 Module 2.1 – User Authentication
2 Module 2.8 – Quiz Monitor (non-data association)

5 Conclusion

This project is a development based on the structure and operation of the existing manual quiz conduction system present in the university. The design is optimized to blend in these characteristics so as to easily implement the system.

This system will require some modules (service level) to be run on a reliable server, preferably a windows 2003 server. The rest will run on normal personal computers to be used as a front-end to the users.

Security is the number 1 priority. The overall design is aimed at decentralizing the functionality and operation into individual many processes and this may seem very unprofessional but this actually is a security characteristic. This means students only get the student modules, similarly faculties and administration accordingly. This will disallow a student from doing anything other than his privileges as a student allow, similarly with faculty and administration. No one can play someone else. A bigger security measure is also implemented in this application, this is a decentralized and universal user authentication tool which does all verification for modules and gives individuals access to only their specific module. This module also does handling of administration data for security reasons. All changes will be sent to the user authentication module and it and only it can communicate to the administrative level databases, i.e. No remote connections will be accepted. Also each module requires a log-in and privacy related modules have the minimize option to avoid on lookers from viewing the application and data. Also the login screens and other suitable screens have the minimize option disabled so it makes it easier to spot fraudulent activity in the labs during tests. Also to avoid intruders and foreign logins from going unnoticed the login procedure stores the IP address of the client PC. This can easily be used to reverse DNS lookup name of the PC and also by name as well as IP Address range wise it can be used to locate the pc. Thus an intruder answering in the other lab or library, etc will be spotted by the Quiz Monitor. This as well as all transactions are logged for analysis during moments of suspicion or complaints of malpractice. The databases are also separated to increase security as Question Database has read only access for students and answers are stores separately. The duration disallows later login to manipulate the answers. Note that the questions answered are a random selection of a few from the many present in the database. Thus passing the answer is severely disabled as only a few will overlap and there is no way to go back to review the question.

The automation allows eased/foolproof (transparent) communication between students, faculty and administration. Thus nothing goes unnoticed in all domains. Also result generation and vital statistics are calculated and displayed on screen. Such as attendance, current logins, number of rejections, student lookup to see who complained and track erroneous feedback on an administrative level.

Generally this system is developed to ease & automate the task of quiz conduction in modern sophisticated manner, making no compromise on security. On the contrary improving the fairness and authenticity of the quiz evaluation component as a method of evaluating students based on their knowledge of the subject.