9.2 – Justification of Chosen Solution

I have chosen to make a game using a GUI, because it suited the clients wishes of the game being simple to use and could be used independently because the pupil can use the system, and there will be prompts for the user to types. The game will also be able to be more colourful and interesting using a GUI than just using the other solutions and will be easier to maintain in the long term, I can also make developments easily like adding more subjects in the near future and changing the layout for different age groups. The client also wanted all pupils to be able to access the system easily and understand how to use the system and I felt the GUI was the best way to do this, because I can use buttons and graphics to display what the client want them to do, which suits my client because it does not require keyboard shortcuts only the use of a mouse and the keyboard to type answers.

Design

Design

Overall System Design

1.1. Short description of the main parts of the system

My system will consist of five main parts, showing briefly what the system is meant to do in terms of input, output and processing:

- Log in window
  - Displays log in window and asks user to input user number and password.
  - Checks user number and password in pupil database to see if user is an existing user.
  - If User name = 'Admin' then checks password against the expected password, in the database.
  - Admin can access all databases and results and add pupils to the database; pupils can only access the game.
  - If accepted then it will grant access to the game.

- Allow users to answer questions using a graphical user interface:
  - Menu to select which mini game to play.
  - Select a difficulty from 3 options
  - User should be shown questions via a Graphical User Interface and will be asked to answer questions on the chosen topic.
  - They will have a box to enter the answers in
  - Questions will continue to be shown until time limit is up or they have answered all the questions.

- Answers should be gathered and checked against a database containing all answers to all the questions:
  - Answer is compared with the answer of the same question in the database.
  - If question is correct an image of a tick will be displayed with a sound effect of clapping, if incorrect then an image of a cross will be displayed.

  - If answer is correct then the score will be amended by 1, if answer is incorrect the score will stay the same.
  - Check time limit and store how long it took pupils to answer that particular question.

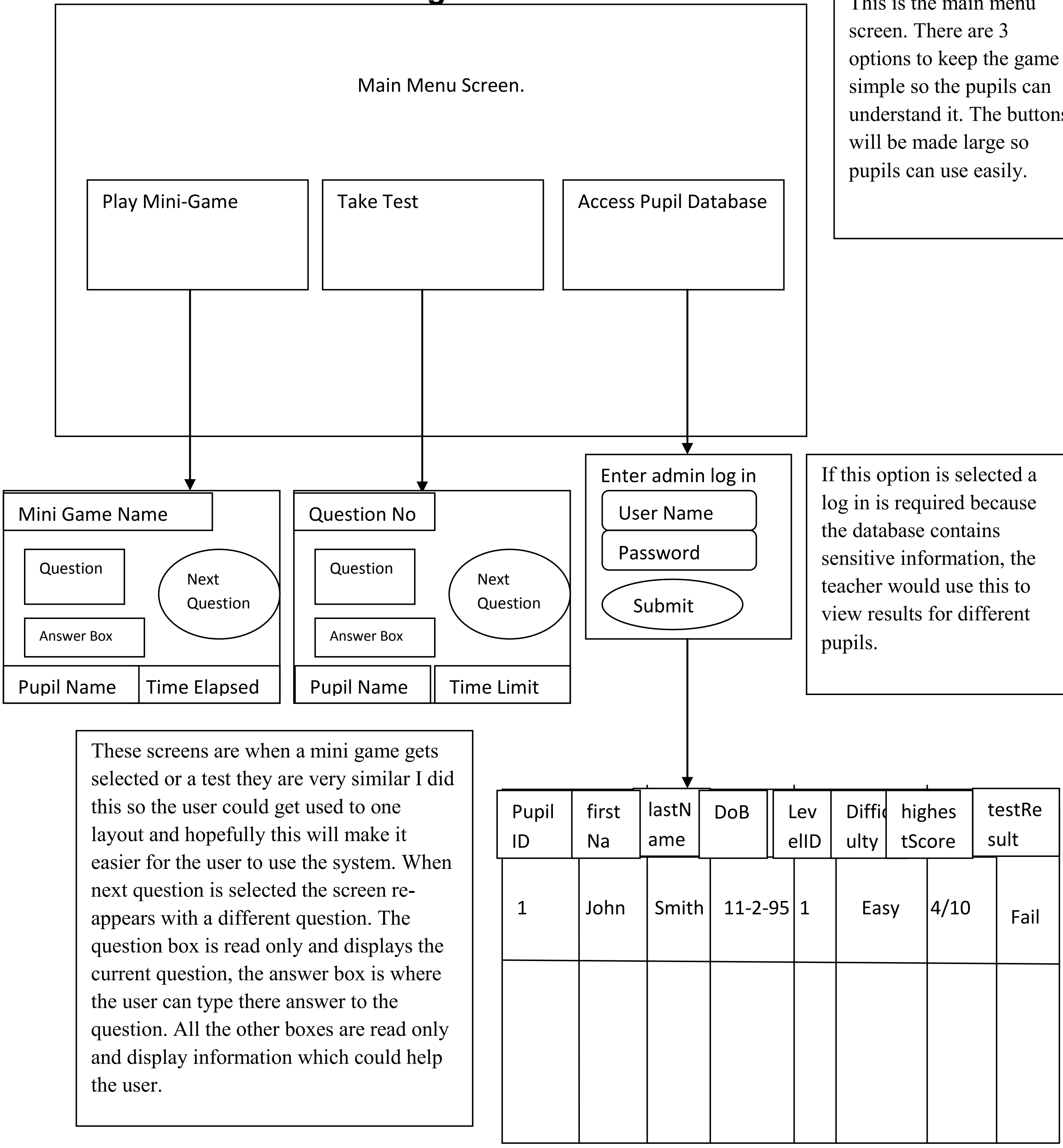
- Compares score to average scores and previous scores, data is stored in a database holding all answers and results from pupils and also add results to the pupil records:
  - Add result, difficulty and time completed in to the pupil database
  - Open pupil records for the selected set of questions.
  - Compare results against the previous attempt and record whether they improved or not and by what score.
  - Highlight name if pupil score is lower than average score or teacher's expectation.
  - Provide a guide to which pupils need help and which need harder questions.
  - Shows a graph to show improvement in class average score over recent weeks.

- Outputs the result of the test and reports a copy of results to teacher:
  - The score will be outputted to the pupil.
  - Send report to teacher consisting of any pupils scoring lower than average and what there score was, using pupil number.
  - Print a record of the pupil results with all the details on, for paper reference.

1.2. System flowcharts showing an overview of complete system

See end of document

2. User Interface Designs

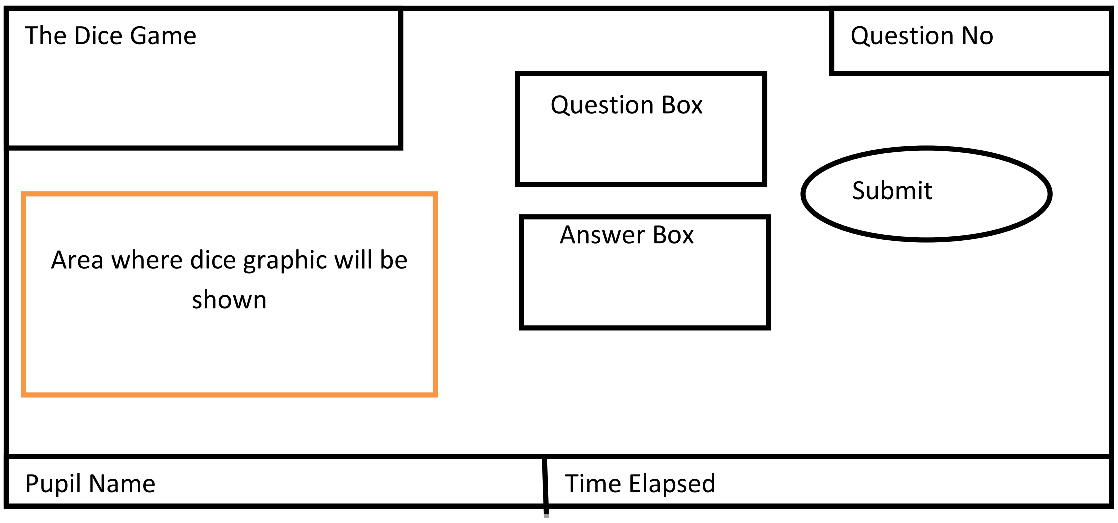


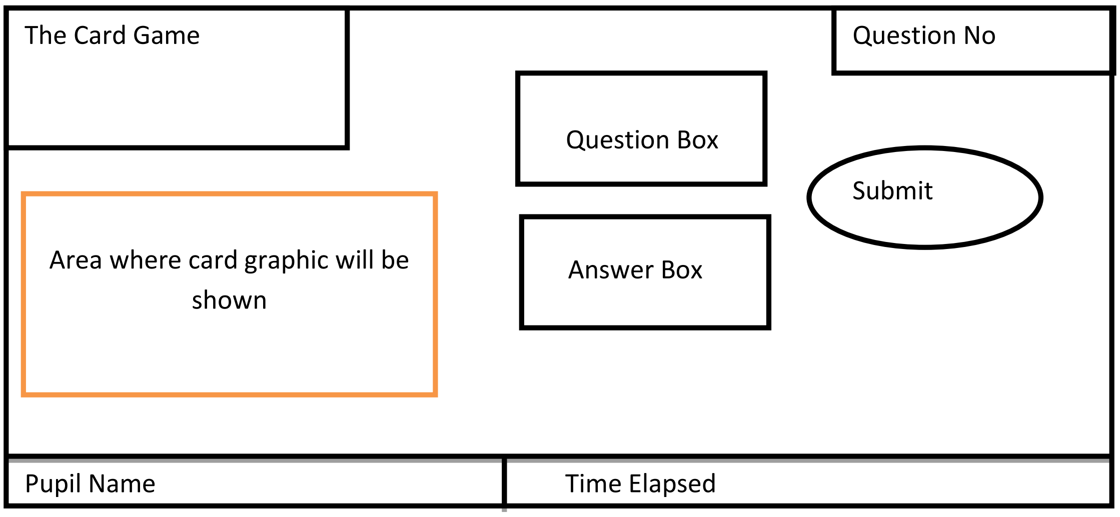
Pupil ID	first Na	lastName	DoB	LevelID	Difficulty	highestScore	testResult
1	John	Smith	11-2-95	1	Easy	4/10	Fail

This is the main menu screen. There are 3 options to keep the game simple so the pupils can understand it. The buttons will be made large so pupils can use easily.

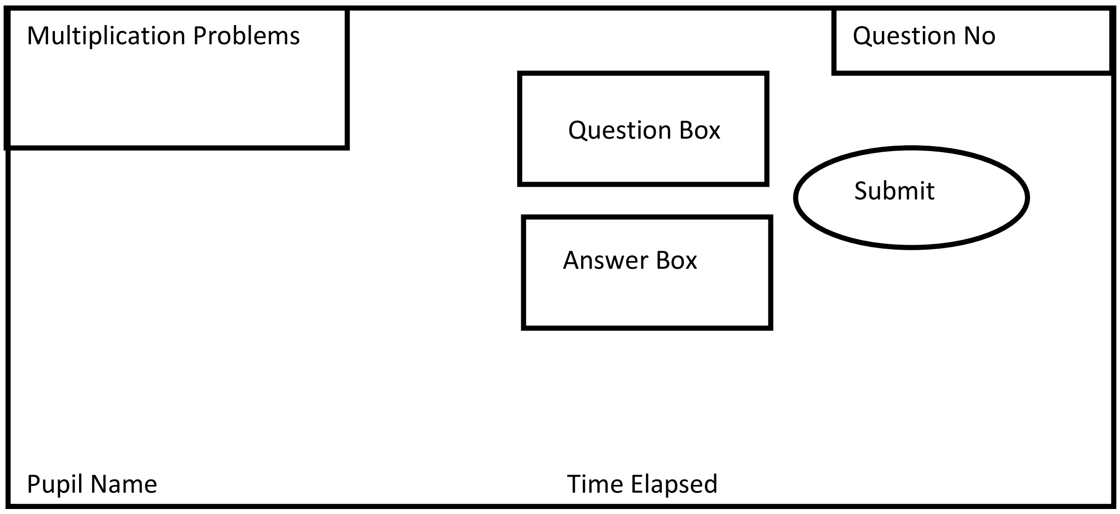
If this option is selected a log in is required because the database contains sensitive information, the teacher would use this to view results for different pupils.

These screens are when a mini game gets selected or a test they are very similar I did this so the user could get used to one layout and hopefully this will make it easier for the user to use the system. When next question is selected the screen re-appears with a different question. The question box is read only and displays the current question, the answer box is where the user can type there answer to the question. All the other boxes are read only and display information which could help the user.



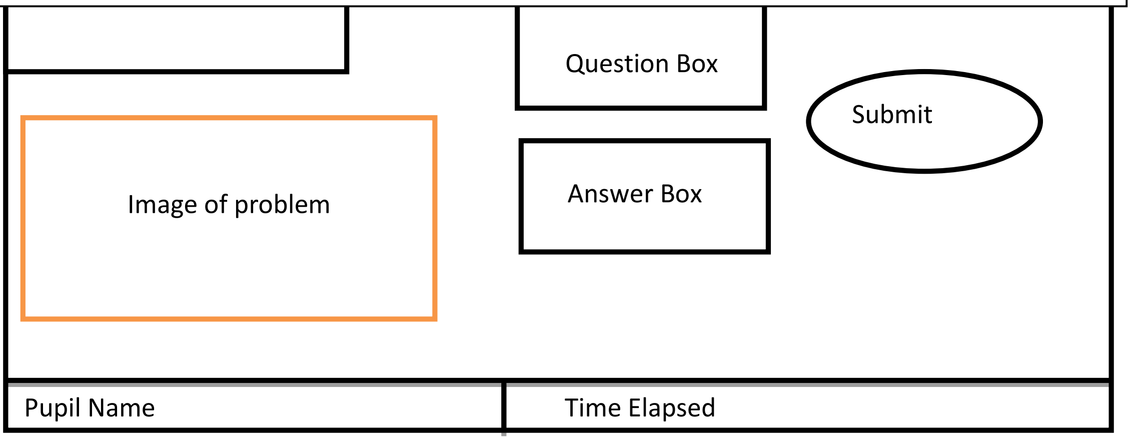


Above are the two main mini game modes the dice and card game these both contain a graphic to engage the pupils and these designs model the rest of the game mode designs as I want the system to be used as a teaching tool and an individual tool.

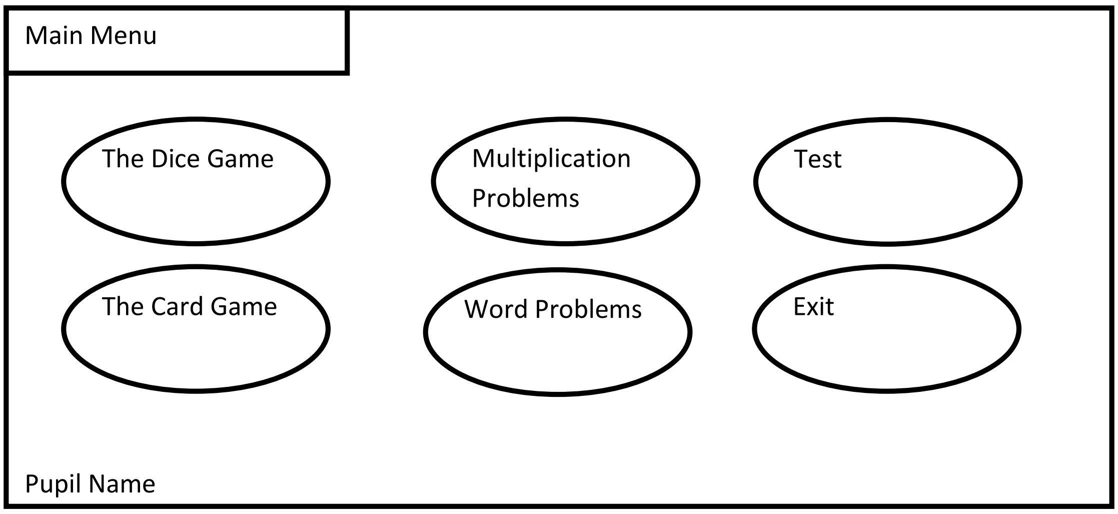


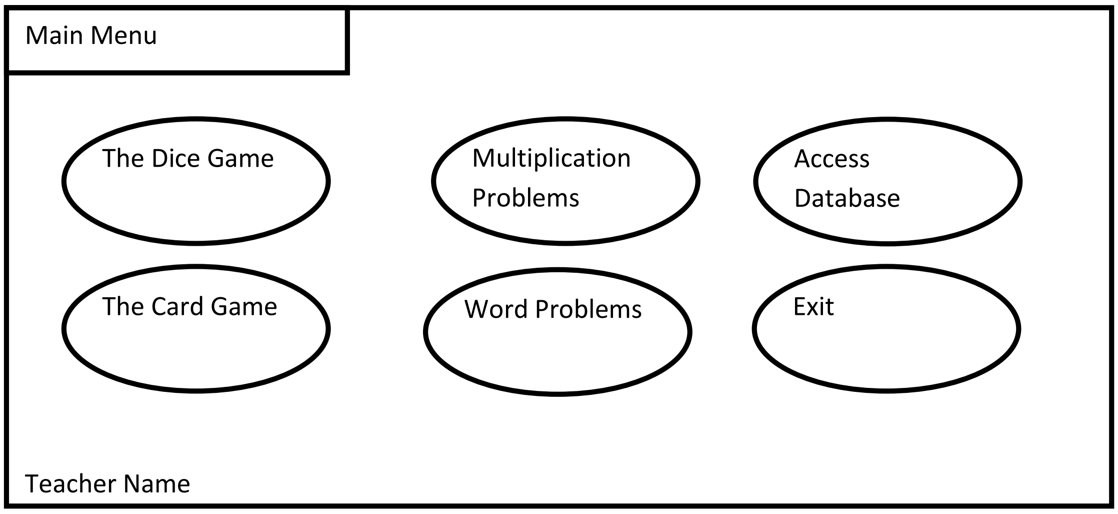
This is the UI screen for the word problem mini game just like the other game modes I have tried to keep the design similar and simple so the pupil can get used to the feel of the system and use it independently.

Above is the UI screen for the multiplication problem game mode I have kept with the similar design so that pupils can get used to the system and have a feel for the system.

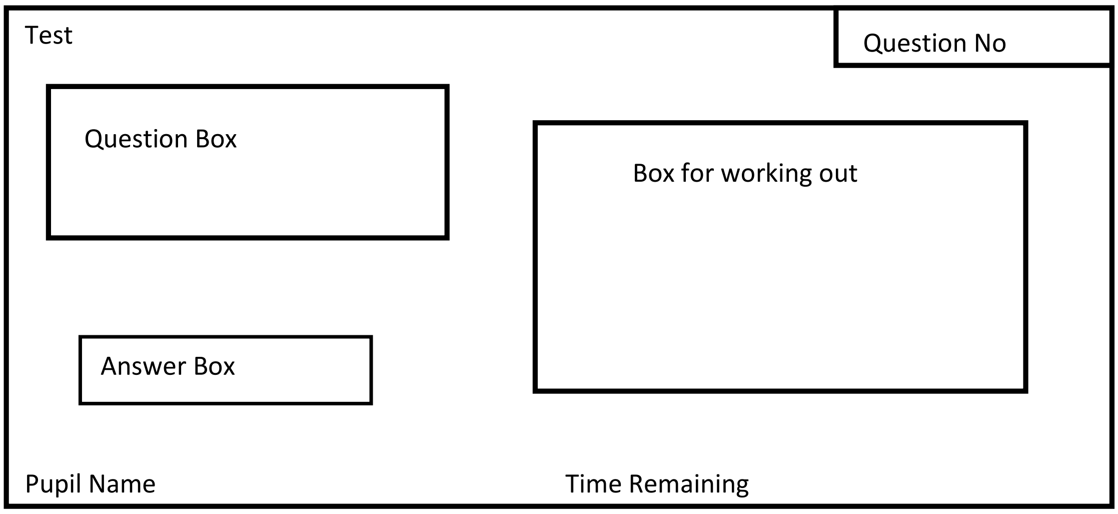


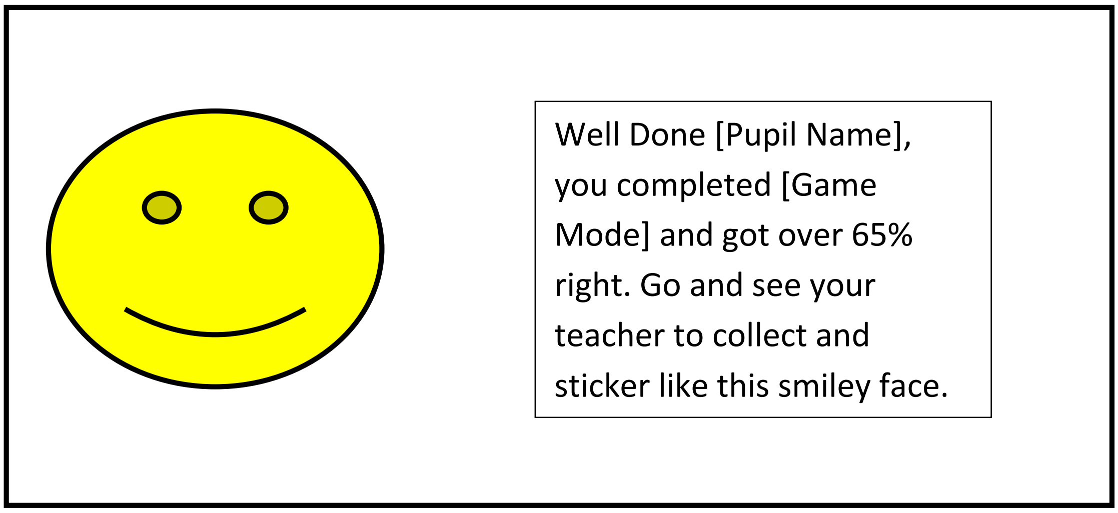
Below are the two menu screens the first is from the point of view of a pupil, the second one is from the point of view of the teacher, I have made the design simple so that the pupils can use the system independently.





Here is the test screen which does not have graphics, it is a plain screen and is meant to feel like a test so that pupils have a feel of what a test is like.




Well Done [Pupil Name], you completed [Game Mode] and got over 65% right. Go and see your teacher to collect and sticker like this smiley face.

This is the screen which will be shown at the end of a test or game mode to congratulate the pupil on a good result this has been requested by client.

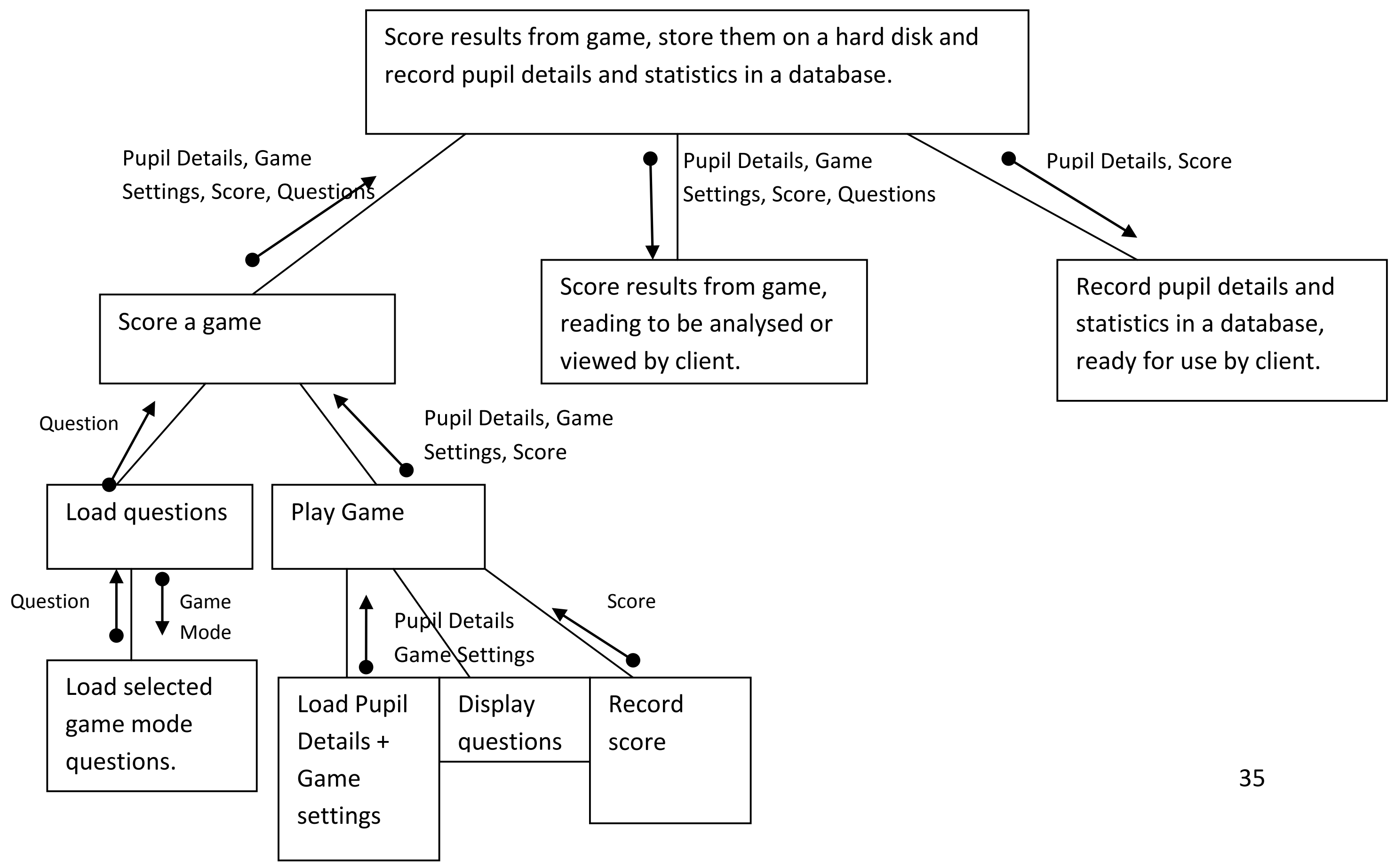
3. Hardware Specification

The following is a list of the hardware parts required to run the system effectively:

- A keyboard for input.
- A mouse for input. The combination of keyboard and mouse is the method which my
  client is the way the pupils are taught to use the computers therefore I will use this method to help make the system easy for use.
- A laptop or PC with a 17" screen, the large screen is vital as it will assist the pupils in being able to understand the system as it won't limit the size of icon.
- The laptop will need to contain at least 512MB RAM, preferably 1GB RAM, this will make the system run smoothly and at a fast rate so there is no wait between question.
- The Laptop or PC must have a operating system of windows XP or above.
- The Computer must have an Intel Pentium Dual Core processor or above, as this will aid the RAM in making the system look smooth and therefore professional.

4. Program Structure

4.1. Topdown Design Structure Charts


Score results from game, store them on a hard disk and record pupil details and statistics in a database.
Pupil Details, Game Settings, Score, Questions
Pupil Details, Game Settings, Score, Questions
Pupil Details, Score
Score a game
Score results from game, reading to be analysed or viewed by client.
Record pupil details and statistics in a database, ready for use by client.
Question
Pupil Details, Game Settings, Score
Load questions
Play Game
Question
Game Mode
Pupil Details Game Settings
Score
Load selected game mode questions.
Load Pupil Details + Game settings
Display questions
Record score

4.2. Algorithms in Pseudo Code

Current Score Counter:

```
CurrentScoreCounter → 0
For Question 1 to 30
If Answer → True
Then CurrentScoreCounter → CurrentScoreCounter +1
Test Score →  Counter
POST[TestScore]
```

Result:

```
Input TestScore
If TestScore >= 12
Then Result → True
If Result → True
Then Result = ‘Pass’
END IF
If TestScore >= 20
Then Result → True
If Result → True
Then Result → ‘Merit’
END IF
If TestScore >= 25
Then Result → True
If Result → True
Then Result → ‘Distinction’
END  IF
POST[AssesmentResults]
```

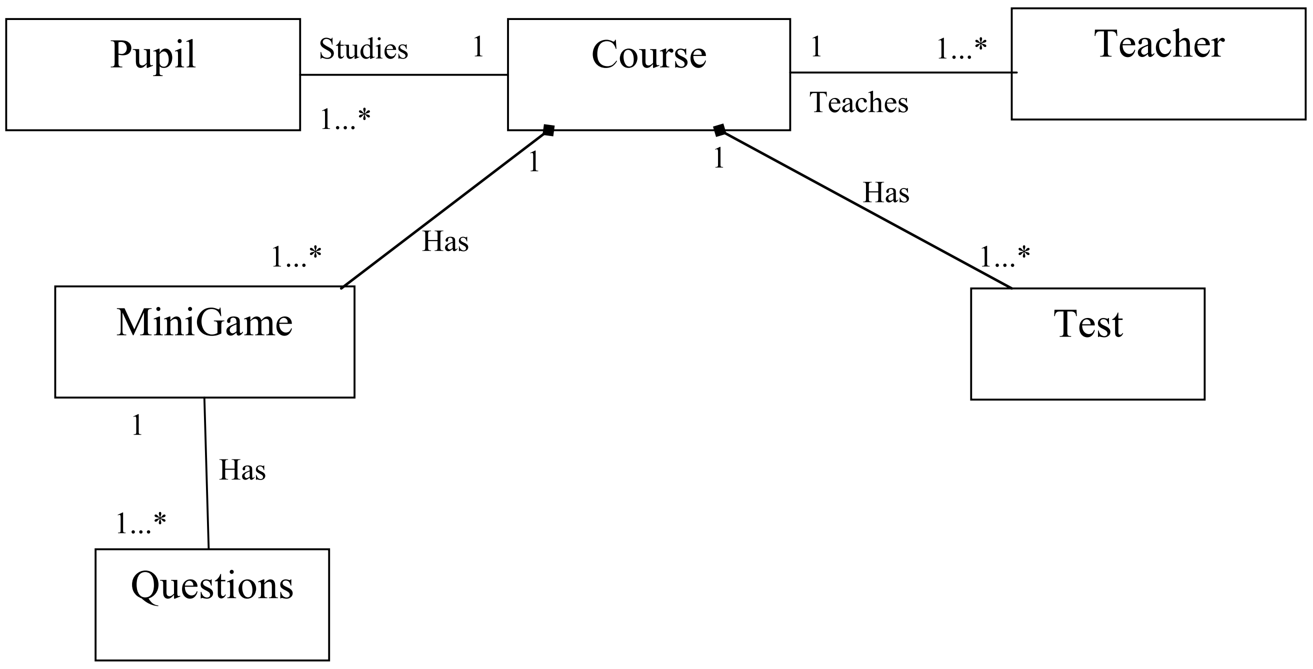
Pupil requires help:

```
Input TestScore
If TestScore  < 6
Then ExtraHelp → True
Else ExtraHelp →  False
POST[ExtraHelp]
```

Exceptional Score:

```
Input TestScore
If TestScore >= 30
Then Exceptional → True
Else Exceptional → False
POST[Exceptional]
```

4.3. Object Diagrams



4.4. Class Definitions

Pupil
firstName lastName SchoolYear EmergencyContact Level PupilNumber
Get firstName Edit firstName Get lastName Edit lastName Get SchoolYear Edit SchoolYear Get EmergencyContact Edit EmergencyContact Get Level Edit Level Assign PupilNumber Edit PupilNumber

Teacher
lastName Subject Title TeacherNumber
Get lastName Edit lastName Get Subject Edit Subject Get Title Edit Title Assign TeacherNumber Edit TeacherNumber

Question
QuestionNumber SchoolYear Level Difficulty
Assign QuestionNumber Edit QuestionNumber Get SchoolYear Edit SchoolYear Get Level Edit Level Get Difficulty Edit Difficulty

Test
TestName TestID Difficulty Level Question
Get TestName Edit TestName Get Level Edit Level Get Difficulty Edit Difficulty Assign TeacherNumber Edit TeacherNumber Get Question

Course
courseName
Modules
Level
ModuleNumber
CourseNumber
Get moduleName
Edit moduleName
Get Level
Edit Level
Assign moduleNumber
Edit moduleNumber
Assign CourseNumber
Edit CourseNumber

MiniGame
MiniGameName
Difficulty
Questions
MiniGameID
Level
Time
Get MiniGameName
Edit MiniGameName
Get Difficulty
Edit Difficulty
Get Questions
Edit Questions
Get Level
Edit Level
Get Time
Edit Time
Assign MiniGameID
Edit MiniGameID

5. Prototyping

5.1. Consideration of impact on design and development.

When designing my system it may be beneficial to the development to produce prototypes of certain modules or elements in order to check that they function as expected and fulfil the user's specifications.

In my system, it may be useful to prototype some of the following items:

The main menu screen to ensure it is easy to use and readable for children of all ages.
I will also test the different game modes to see whether the game modes are not too similar and to see whether the screens are clear enough for use.
I will test the test mode to check whether the format works well and whether the score is correct.
I will test the printout feature to see whether it prints out the report I requested.
As well as this I plan to test the security of the system as security is vital, as I have to adhere to the data protection act.

6. Definition of Data Requirements

6.1. Identification of all data input items

The following items need to be input for use in the system:

- Menu Choice
- Answers to the questions
- Number of Questions per game

6.2. Identification of all data output items

The following items are output during the game:

- User names
- Date
- Score
- Graphic (Correct or Incorrect)
- Question Number
- Question Remaining
- Time Limit
- Time Elapsed
- Percentage
- Total Score

The following items are output by the database part of the system:

- The User Name

- No of Games attempted by user
- No of questions answered by user
- No of correct questions by user
- No of incorrect questions by user
- Average Score of user
- Average Score
- Average time of user
- Average time
- Highest score achieved by user

6.3. Explanation of how data output items are generated

The output items from the score sheet part of the system are generated as follows:

User names Typed input by user

Date Fetched from system clock

Current user identified by PupilNumber

Questions remaining in the game Total Questions in GameMode – Questions Answered

Question number System automatically assigns innings number
to 1 and then changes it to 2 after the 1st question answered and etc...

Question pass rate Total Score / Questions Answered

User Number Automatically assigned in database

Player names Selected by user from database using PupilNumber, or if not
present in database, entered as typed input

Start time Fetched from system clock

End time Fetched from system clock

Score Calculated from result of each question , score is amended if necessary

Questions faced by each user Calculated by system, +1 added to on question counter after each question

Percentage of correct answers (Pupil score/questions faced) * 100

The following table demonstrates how data output items for the database part of the system are generated:

The players' names are fetched from database

No of Game attempted played by each player is fetched from database and amended after each game.

Highest score for each player fetched from database

Pass rate achieved in tests by each player (Test Score/totalnooftests)*100

6.4-Data Dictionary

This first part is for the results database.

Name	Data Type	Length	Validation	Example Data	Comment
PupilID	Integer	4 bytes	1 to 500	23	Automatically Generated
TeacherID	Integer	4 bytes	1 to 100	12	Automatically Generated
PupilName	Array of String	25 Chars	Must Exist	Smith, Tom Richard	Full Name, Surname, other names after.
TeacherName	String	25 Chars	Must Exist	Francis, Jane Elizabeth	Full Name, Surname, other names after.
Admin	Boolean			Admin/Non-Admin	Checks whether user is admin or non admin.
SchoolYear	Integer	1 bytes		1	School Year
DifficultyID	Intger	1 bytes		1	Automatically Generated
DifficultyName	String	7 chars		Easy, Medium, Hard	Difficulty at which the game is

					taken.
MiniGameChoice	String	50 Chars		The Card Game	Which mini game was taken.
Result	Integer	2 bytes	1 to 30	23	Score from the game
AverageScore	Integer	2 bytes	1 to 30	14	The average score from pupils on that mini game
Percentage	Real	4 bytes		78%	The percentage out of the number of questions.
Date	Date	3 bytes		20/10/12	The date the test was attempted.
TimePerQuestion	Integer	4 bytes	1 to 60	12	The time it took the user to answer the question.
PupilScoreSatisfactory	Boolean			TRUE	Whether pupil score a satisfactory score.
LevelID	Integer	2 bytes	1 to 5	3	The level at which the pupil is working at.

LevelName	String	25 Chars		Poor, Satisfactory, High	The name of the level the pupil is working at.
QuestionsAnswered	Boolean			FALSE	Whether the pupil answered all the questions
QuestionAttempted	Boolean			TRUE	Whether the pupil attempted the question.

This second part is for the pupil database.

Name	Data Type	Length	Validation	Example	Comment
PupilID	Integer	4 bytes	1 to 500	23	Automatically Generated
PupilName	Array of String	25 Chars	Must Exist	Smith, Tom Richard	Full Name, Surname, other names after.
PupilAddress	String	50 Chars	Must Exist	23 Chestnut rise, Newmarket, Cambridge	House name/number and road, town, county.
SchoolYear	Integer	1 bytes		1	The School Year that

					specific pupil is in.
Emergency Contact Name	String	25 Chars	Must Exist	Smith, Mary	Surname, Other names after
EmergencyContact Address	String	50 Chars	Must Exist	23 Chestnut rise, Newmarket, Cambridge	House name/number and road, town, county.
PreviousResults	Array of Integer	4 bytes	0 to 30	3,17,23,25,30	Shows last 5 previous results by that pupil
LevelID	Integer	2 bytes	1 to 5	3	The level at which the pupil is working at.
LevelName	String	25 Chars		Poor, Satisfactory, High	The name of the level the pupil is working at.
Percentage	Array of real	8 bytes	0 to 100	78%, 45%, 21%, 94%, 2%	Shows last 5 previous results by that pupil
ExtraHelp	Boolean			TRUE	Whether the pupil needs extra help

					or not.

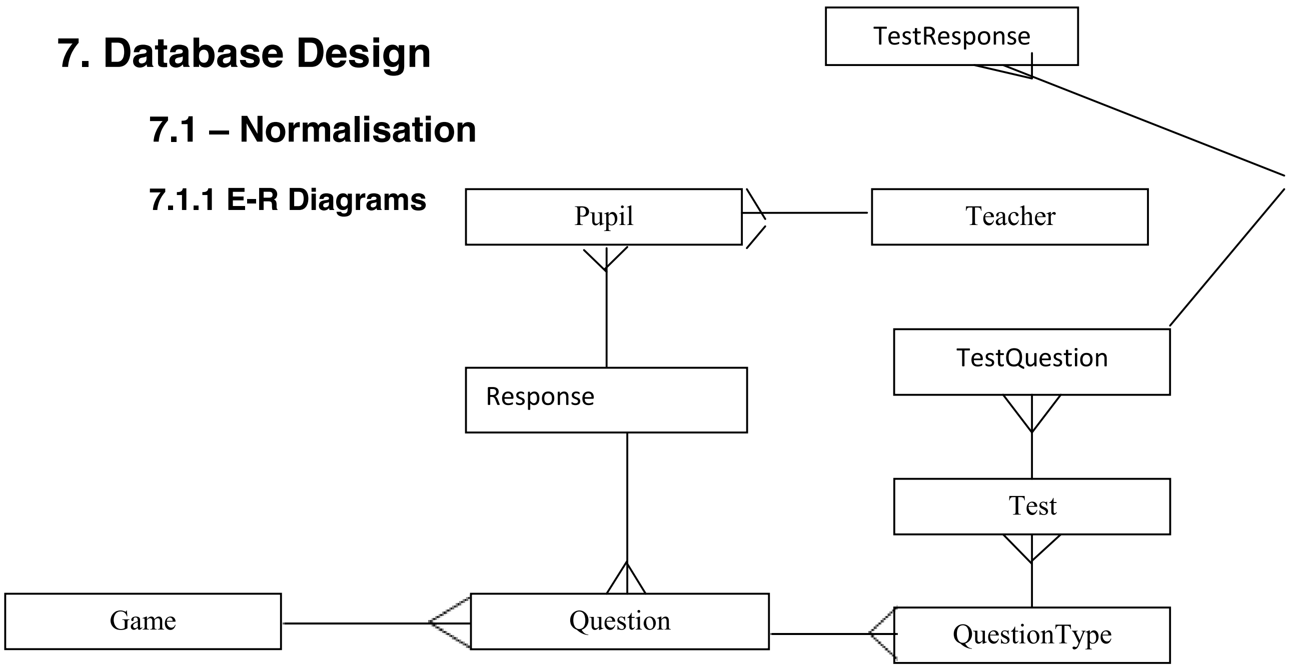
6.5. Identification of appropriate storage media

An appropriate method of storing the system and database files would be on the client's hard drive because secure, long term storage is required. To back up the system files the best medium would be a flash drive, which could store both the system files themselves and the data generated by the system. I have decided to use a USB storage device because my client will be using a laptop and so the need for portability is a important factor and the USB drive is the best device for this. It will also be transported a lot so will need to be robust which a USB is. USB devices are also available in different capacities starting from 1GB up to 64GB, this will suit my user because she can start with an 8GB and then upgrade if necessary.

7. Database Design

7.1 – Normalisation

7.1.1 E-R Diagrams



7.1.1.1 - Entity Descriptions

Pupil (<u>PupilID</u>, FirstName, LastName, Gender, Address Line 1, Address Line 2, Town, PostCode, EmergencyContactName, EmergencyContactTelNo,)

Teacher (<u>TeacherID</u>, FirstName, LastName, Gender, Email)

Question (QuestionID, DifficultyName, LevelNumber, QuestionSet, AnswerSet, *GameID, QuestionTypeID*)

QuestionType (QuestionTypeID, QuestionTypeName*)*

Game (GameID, GameTypeName, GameTypeNumber, OverallTimeTaken, Score, Grade, Percentage*)*

Test (TestID, TestQuestionSet, TestAnswerSet, TestTimeLimit, Test Total Score, GradeBoundaries, *QuestionTypeID*)

TestQuestion (TestQuestionID, TestQuestionSet, *TestID*)

TestResponse (TestResponseID, TestAnswerSet, UserResponse, Match, TimeTaken, *TestQuestionID*)

Response (ResponseID, AnswerSet, UserResponse, Match, TimeTaken, *QuestionID, PupilID*)

7.1.2 – UNF – 3NF

🔑 = Primary Key * = Foreign Key

UNF		1NF		2NF		3NF
PupilID	🔑	PupilID	🔑	PupilID	🔑	PupilID
FirstName		FirstName		FirstName		FirstName
LastName		LastName		LastName		LastName
Gender		Gender		Gender		Gender
DateofBirth		DateofBirth		DateofBirth		DateofBirth
Address Line 1		Address Line 1		Address Line 1		Address Line 1
Address Line 2		Address Line 2		Address Line 2		Address Line 2
Town		Town		Town		Town
PostCode		PostCode		PostCode		PostCode
EmergencyContactName		EmergencyContactName		EmergencyContactName		EmergencyContactName
EmergencyContactTelNo		EmergencyContactTelNo		EmergencyContactTelNo		EmergencyContactTelNo
Gender		TeacherID				
TeacherFirstName		Gender	🔑	TeacherID	🔑	TeacherID
TeacherLastName		TeacherFirstName		Gender		Gender

Email		TeacherLastName		TeacherFirstName		TeacherFirstName
QuestionID		Email		TeacherLastName		TeacherLastName
DifficultyName				Email		Email
LevelNumber	🔑	QuestionID				
QuestionSet	🔑	PupilID	🔑	QuestionID		
AnswerSet		DifficultyName	🔑	PupilID	🔑	Question
QuestionTypeName		LevelNumber		Score	🔑	PupilID
GameTypeName		QuestionSet		Grade		Score
GameTypeNumber		AnswerSet		Percentage		Grade
OverallTimeTaken		QuestionTypeName		TimeTaken		Percentage
Score		GameTypeName		Match		TimeTaken
Grade		GameTypeNumber				TimeTakenPerQuestion
Percentage		OverallTimeTaken	🔑	GameID		
TestQuestionSet		Score	🔑	QuestionID	🔑	QuestionID
TestAnswerSet		Grade		QuestionTypeName		Difficulty
TestTimeLimit		Percentage		GameTypeName		Level
TestTotalScore		TestQuestionSet		Difficulty		QuestionSet
GradeBoundaries		TestAnswerSet		Level		AnswerSet
UserResponse		TestTimeLimit		QuestionSet		GameID*
Match		TestTotalScore		AnswerSet		
TimeTaken		GradeBoundaries		TestQuestionSet	🔑	GameID
		UserResponse		TestAnswerSet		GameTypeName
		Match		TestTimeLimit		GameTypeNumber
		TimeTaken		TestTotalScore		
					🔑	QuestionTypeID
						QuestionTypeName

						QuestionID*
					🔑	TestID
						Testquestionset
						Testanswerset
						Timeallowed
						Timetaken
						Gradeboundaries
						Testtotalscore
						Percentage
						Questiontypeid*

7.2- SQL Queries

Year 1 Pupil Search, Who's average score is higher than average.

I am using the python format to execute my SQL Queries.

```
"'Select {0}, {1}, {2}, {3}, {4}

From {5}, {6}

Where {3} >= {4} and {2} = "Year 1"

Order By {3} ASC"""

.format (PupilID, lastName, SchoolYear, PupilAverageScore, AverageScore, Pupil, Result)
```

Here I selected different entities from different tables using a .format statement.

Here I enter what the results of the query have to match to be correct. I then order the results of the query by the highest average score.

I get these parameters from my database they are entities in the tables.

Adding a new pupil to the Pupil table.

```
"""Insert into Pupil (firstName, lastName, DateofBirth, SchoolYear)

Values ('{0}',' {1}',' {2}',' {3}')""".format (firstName, lastName, DateofBirth, SchoolYear)
```

Here I add a pupil to the database, I do this using an insert table, I provide what entities I want to add and then I add them using the values statement and a .format statement with the same entities as before.

Updating a pupil record in the database.

Here I update the pupil database, and change the school year and level of year 1 pupils to the expected year 2 level.

```
“””UPDATE Pupil
SET SchoolYear = '2', Level = '3'
WHERE SchoolYear = '1'”””
```

Deleting a pupil from the database.

```
“””DELETE count (*) FROM Pupil
WHERE EXISTS
 (select Pupil.lastName, Pupil.SchoolYear
   from Pupil

  where Pupil.SchoolYear = ‘6’
   and Pupil.lastName = ‘Johnson’)”””
```

Here I check every record in the table for pupils in year 6 and with the last name Johnson, and checked whether this record exists if say I checked against a criteria.

I check for when School year equals 6, and when the pupil’s last name equals Johnson. I then delete the record if it exists.

8. Security and Integrity of System and Data

8.1. Security and Integrity of Data

To prevent the loss of the data it should be backed up on another suitable storage facility, e.g. a paper copy on request of client. To improve the integrity of the data at the entry stage, which is vital in this particular system, drop down boxes and radio buttons will be used in the interface as far as possible to try and ensure that data input is accurate. If typed input is required, stringent validation and verification checks will be in place to try and prevent errors. I will use referential integrity in the database I will do this to ensure that the user cannot add an item into the database that will not work with the database; I will also do this to ensure that when an item is deleted from one section of the database any sections linked with that section will also be amended. This process will stop any chance of there being duplicates and will also keep the database up to date and ensure that only relevant data will be stored which is in compliance with the data protection act.

8.2. System Security

To protect the integrity of the database part of the system, there will be two access levels called administrator and user. The user will only be able to play the games and take test, but the administrator will be able to access and edit the database if required. To log on as an administrator, a password will be needed. Hopefully, if the data has been input correctly, it will be unnecessary to alter the database anyway, but the function to do so must be built in case of error made during input. The data most be stored securely and regularly checked through to make sure all information is relevant and up to date due to the Data Protection Act.

9. Validation

As far as possible, the need for validation in my system has been avoided by the use of drop- down menus or radio buttons, but the following table shows where it will be necessary and the form it will take:

Presence checks to confirm the entry of team names effectively.	A pupil name must be entered to record the matches
Length checks to ensure the lengths of team names are less than 40 characters	The pupil names must be less than forty characters because the string will have to be limited and forty characters is a reasonable maximum length for the name of a pupil.

10.Testing

10.1. Outline Plan

10.1.1. Identification of suitable test strategies

The best method of testing my system would be a mixed testing strategy as the system will de implemented using a suitable combination of top-down and bottom-up design techniques.

Test Series Purpose

1. Test the flow of control. Check the user interface functions correctly and that links between interface forms function as expected. (Top-down testing)

2. Test the validation of input data. It is essential that the validation functions correctly as there is very little scope for error correction once erroneous data is input into the system. (Bottom-up testing)

3. Test that input data is inserted into the correct variables by the system. (Black box and unit testing).

4. Test that all calculations are performed correctly by the results program so the match can be scored effectively.(White box testing)
5. Test that the score data is transferred successfully to the database and is saved into the correct fields. (System testing)

6. Test that the completed scores have been saved successfully for later viewing by the user.(System testing)

7. Test that the system fulfils the objectives stipulated in the analysis section. (Acceptance testing).

10.2. Detailed Plan

10.2.1 – Normal

1.1 Check system, The log in window works correctly and try some data that would expect to be allowed and some that shouldn't. Should load when the enter button is clicked.

1.2 Check that The play Card Game button and check to see that the card game function is loaded and game works and is playable. Should load when the Card Game button is clicked.

1.3 Check that The input Click on the Maximum 'confirm' form for confirm number of button on maximum button questions per game will set the max number of questions correctly should load when the confirm button is clicked.

1.4 Check that The play Dice Game button and check to see that the card game function is loaded and game works and is playable. Should load when the Dice Game button is clicked.

1.5 Check that The play Problem Solving button and check to see that the card game function is loaded and game works and is playable. Should load when the Problem Solving button is clicked.

1.6 Check that The play Multiplication Problems button and check to see that the card game function is loaded and game works and is playable. Should load when the Multiplication Problems button is clicked.

1.7 Check that The play Test button and check to see that the card game function is loaded and game works and is playable. Should load when the Test button is clicked.

1.8 Check that The Access Database button works and check to whether a request for an admin log in appears and whether if entered correctly user has access to the Database. Should load when Access database button is clicked.

1.9 Check that on all mini game modes the interface is correct, when the time elapsed being correct and the submit button and line edits are working correctly.

1.10 Check that the add new pupil function works correctly and check in the database to see whether information has been added correctly.

1.11 Check that the search for pupil function works correctly and try different data, including a pupil who doesn't exist in the database to see if the search function works

1.12 Check that the delete pupil function works, this should re-prompt the user for the admin password, then delete the pupil from the database. Check the correct pupil was deleted and no data has merged into one record.

1.13 Check whether the view average score button works, this should load the average score of all pupils that have played on the game on any game mode excluding Test.

1.14 Check whether the compare average score button works, this should when clicked allow the client to compare a specific pupil's average score with the average score and the system should return whether it is better or worse than the average. On all game modes excluding Test

1.15 Check whether the view average time button works, this should load the average time it took for of all pupils that have played on the game on any game mode excluding Test.

1.16 Check whether the compare average time button works, this should when clicked allow the client to compare a specific pupil's average time with the average time and the system should return whether it is better or worse than the average. On all game modes excluding Test

1.17 Check whether the view average test score button works, this should load the average test score of all pupils that have taken the test Test.

1.18 Check whether the compare average test score button works, this should when clicked allow the client to compare a specific pupil's average test score with the average test score and the system should return whether it is better or worse than the average.

1.19 Check whether the view average test time button works, this should load the average time it took for of all pupils to finish the test.

1.20 Check whether the compare average test time button works, this should when clicked allow the client to compare a specific pupil's average test time with the average test time and the system should return whether it is better or worse than the average.

1.21 Check whether when a pupil answers a questions correct the correct graphic appears and the score gets increased by 1, this should occur in all game modes excluding test.

2.1 Check whether when a pupil answers a questions incorrect the incorrect graphic appears this should occur in all game modes excluding test.

2.2 Check that in test the current score and graphic doesn't appear after questions.

2.3 Check that on each question answer has been given and the answer is acceptable, this applies to all game modes.

10.2.2 – Erroneous

2.4 Test Log in window – I will do this by entering a pupil number that doesn't exist, an error message should be returned stating that the pupil number was not found and the screen should return to the log in window.

2.5 Test Log in Window – I will do this by first entering a pupil name that doesn't exist, an error message should be returned stating the pupil name wasn't found, and then I will test the log in window by not entering anything I will use a presence check to check that the pupil name has been entered if it has not an error message will appear stating a value must be entered, I will use a length check to see if the name is an acceptable and suitable name.

2.6 Test the Access Database, I will test this by first attempting to access the database with the incorrect log in details, this should return an error message stating that only the admin can access the database, and then once I am in the database I will test all the functions in the database.

2.7 Test Add function in access database, I will test this by trying to enter blank spaces in the fields, I will use a presence check to see if a value has been entered I will then use a length check to see if the imputed data is suitable.

2.8 Test delete function in access database, I will test this by trying to delete a record without the admin password which is required to delete a record.

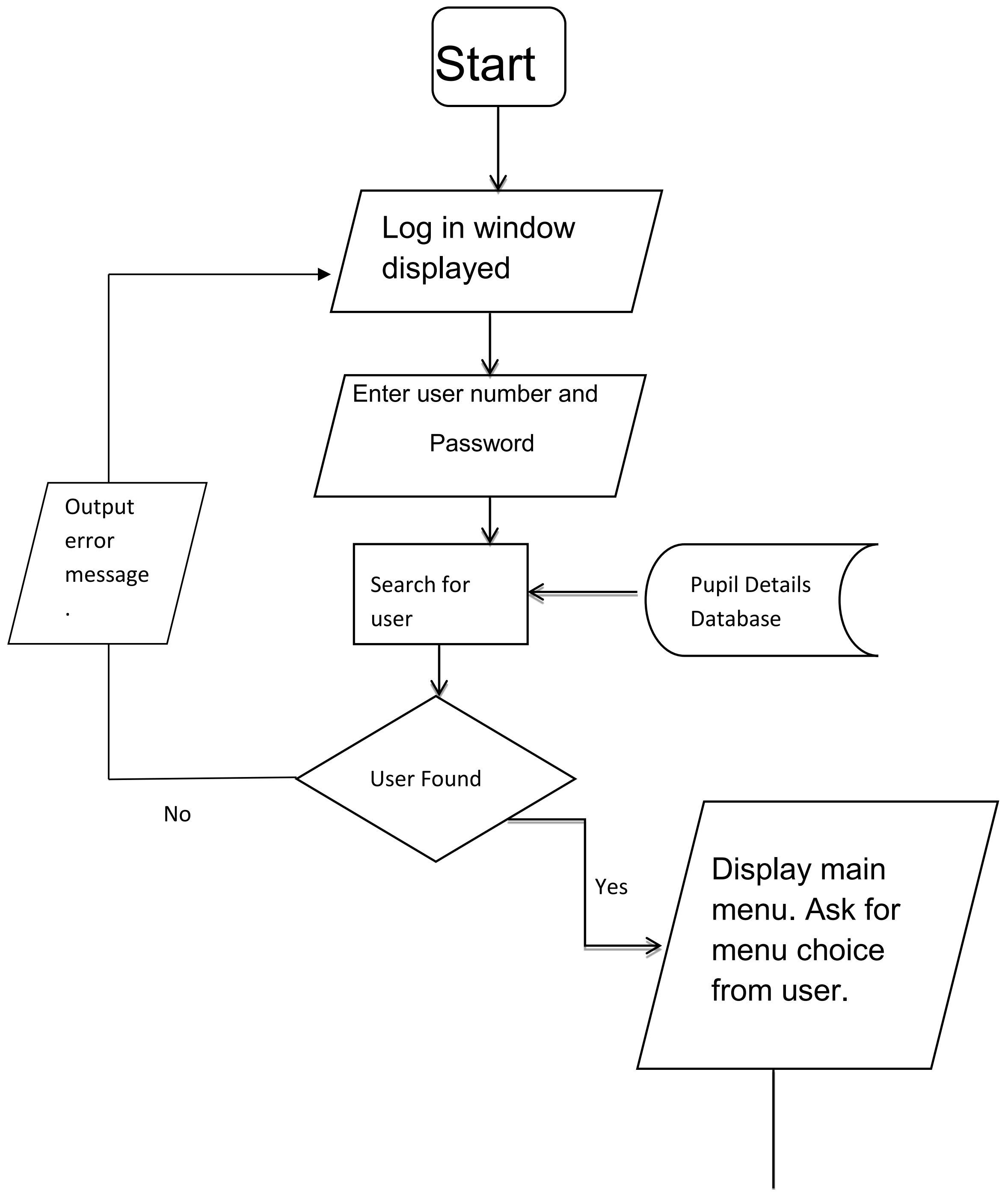
2.9 Test search function in access database, I will test this by trying to search the database by entering a blank, this should return an error message and re-prompt me to enter a value I will then test many data sets to see the search returns what I expect it to return.

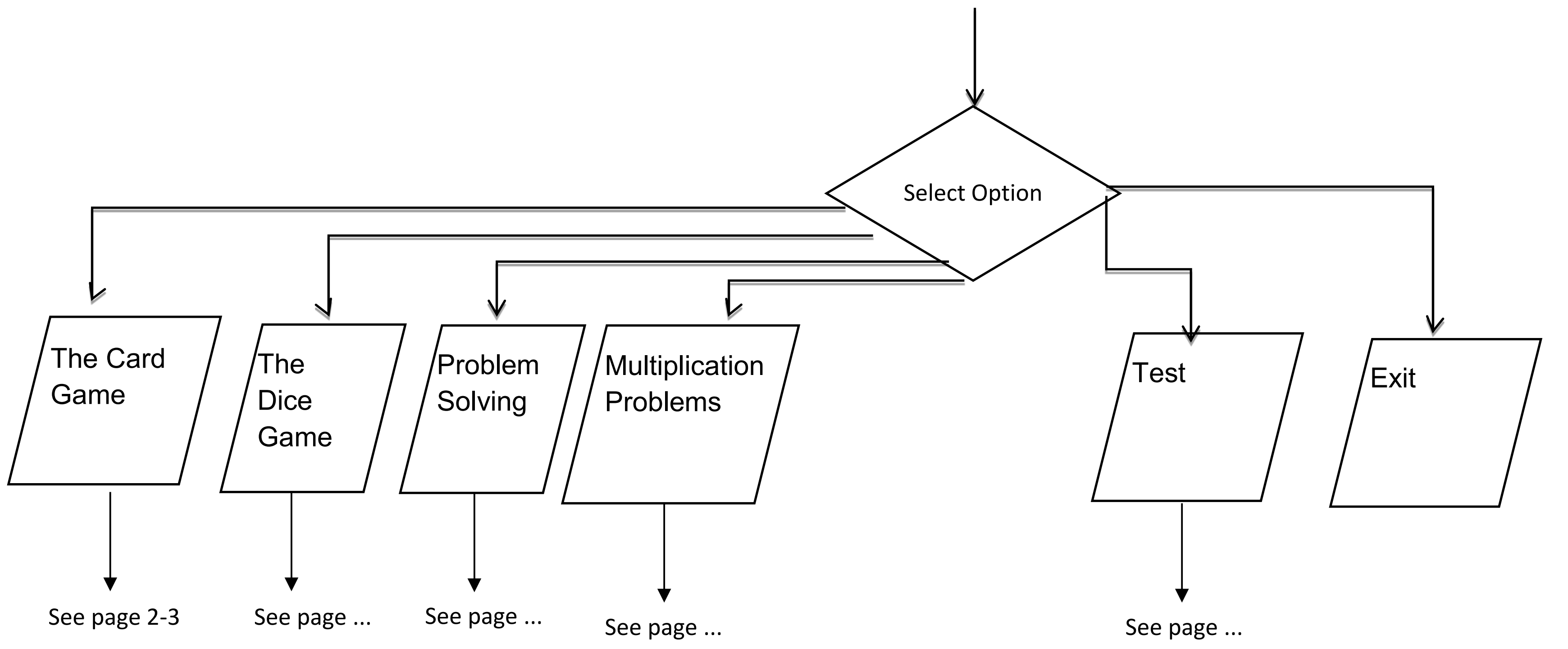
10.2.3 – Boundary/Extreme

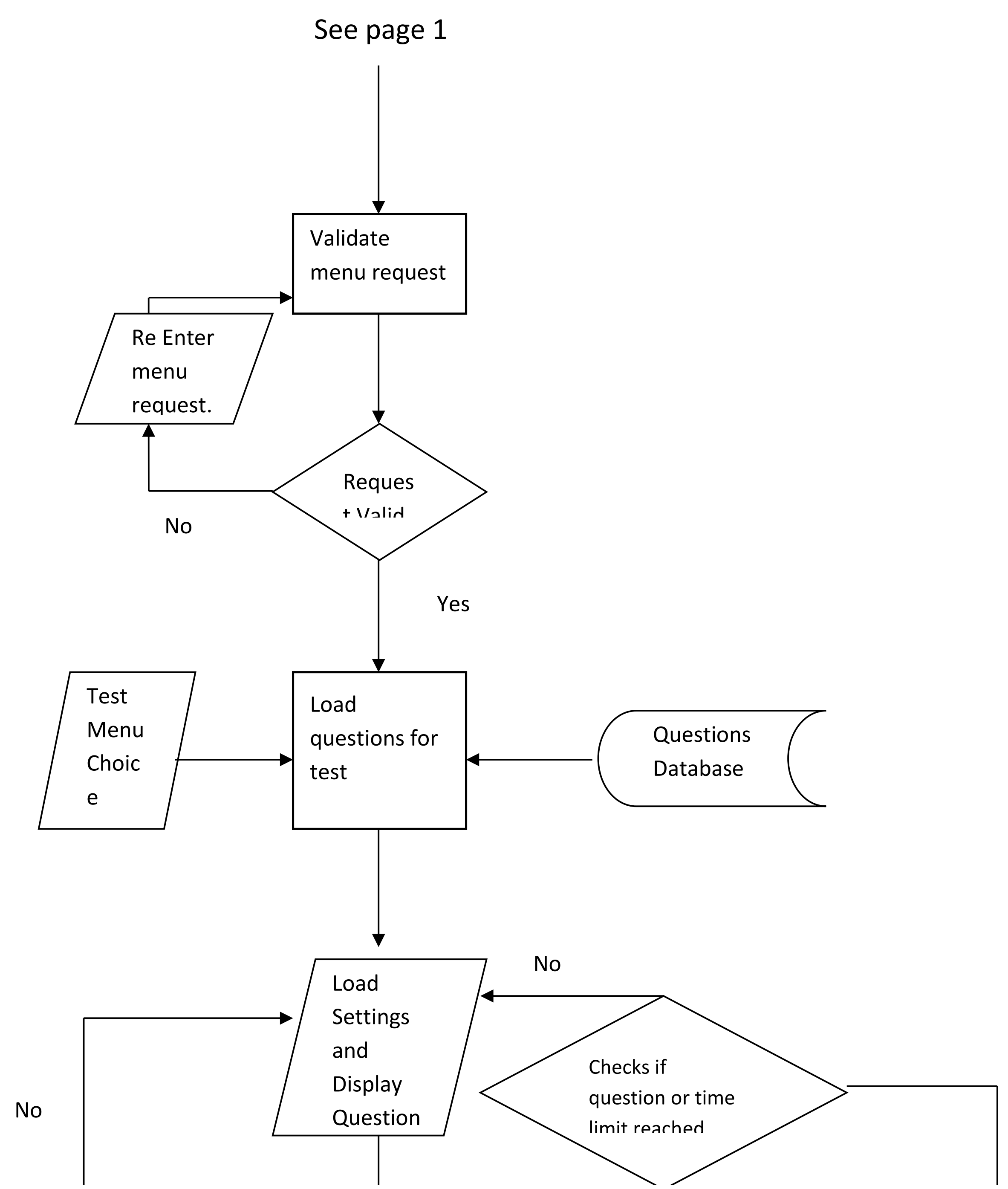
2.1 I will test the pupil number which is only allowed to be 1 to 500, I will enter 0 and 501 these values should return error messages, and I will also test with an empty number which should also return invalid.

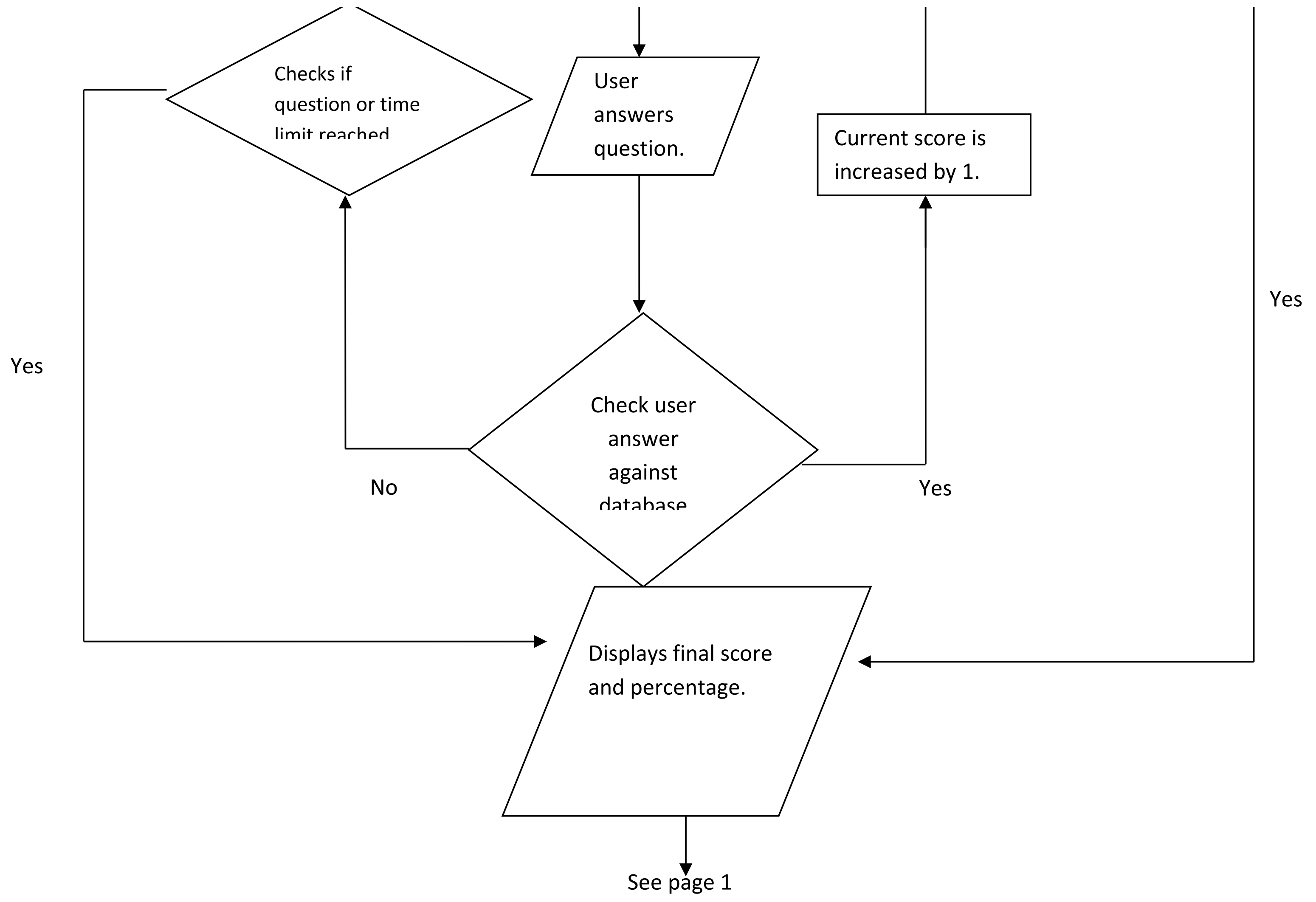
2.2 I will test the pupil name which is only allowed to be 25 characters I will enter 25 and 26, 25 should be accepted but 26 should be refused and the user should be re-prompt.

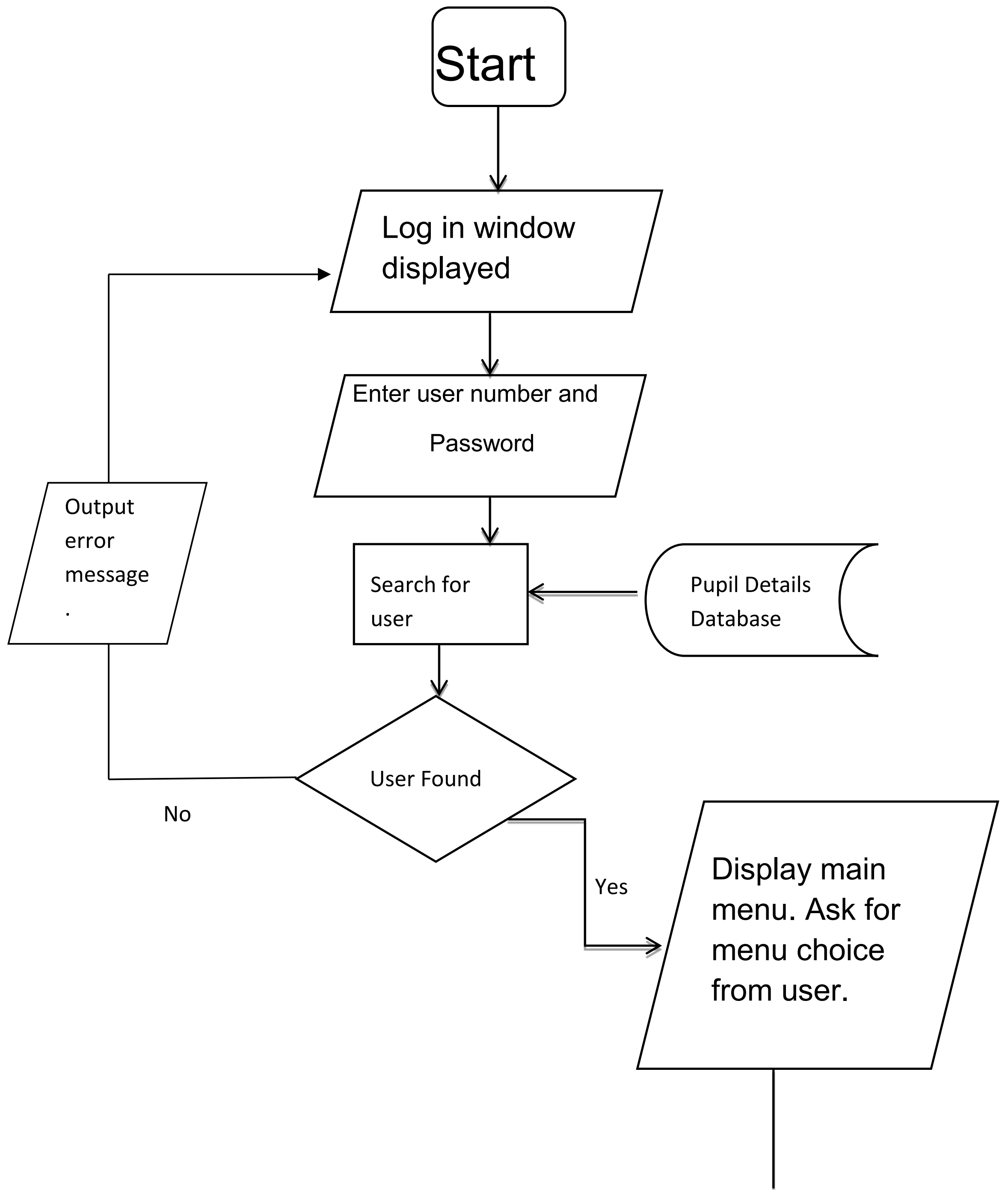
2.3 I will test the max number of questions it is only allowed to be 30 maximum I will enter 0, 30, and 31. 30 should be accepted the other values should be seen as invalid and not allowed, User should be re-prompted.

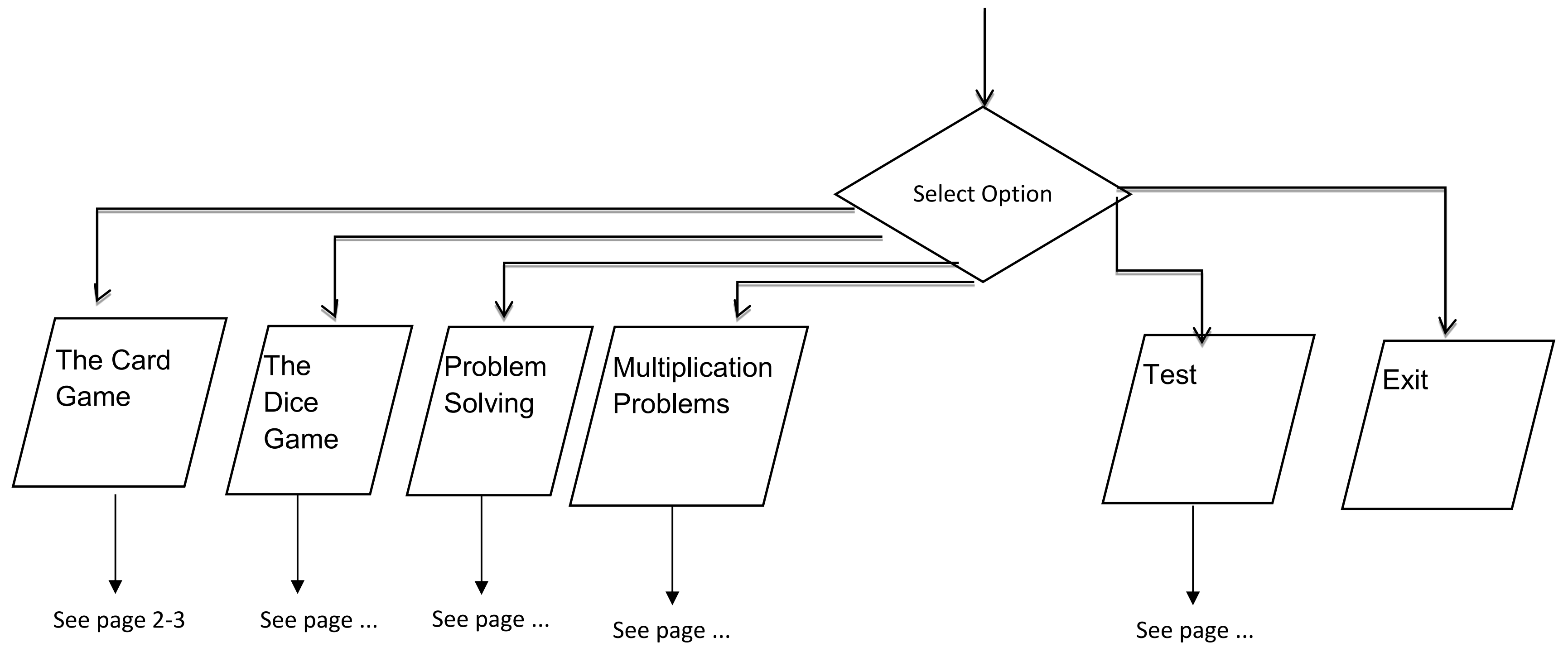

Start
Log in window displayed
Enter user number and Password
Search for user
Pupil Details Database
User Found
No
Output error message.
Yes
Display main menu. Ask for menu choice from user.


Select Option
The Card Game
The Dice Game
Problem Solving
Multiplication Problems
Test
Exit
See page 2-3
See page ...
See page ...
See page ...
See page ...

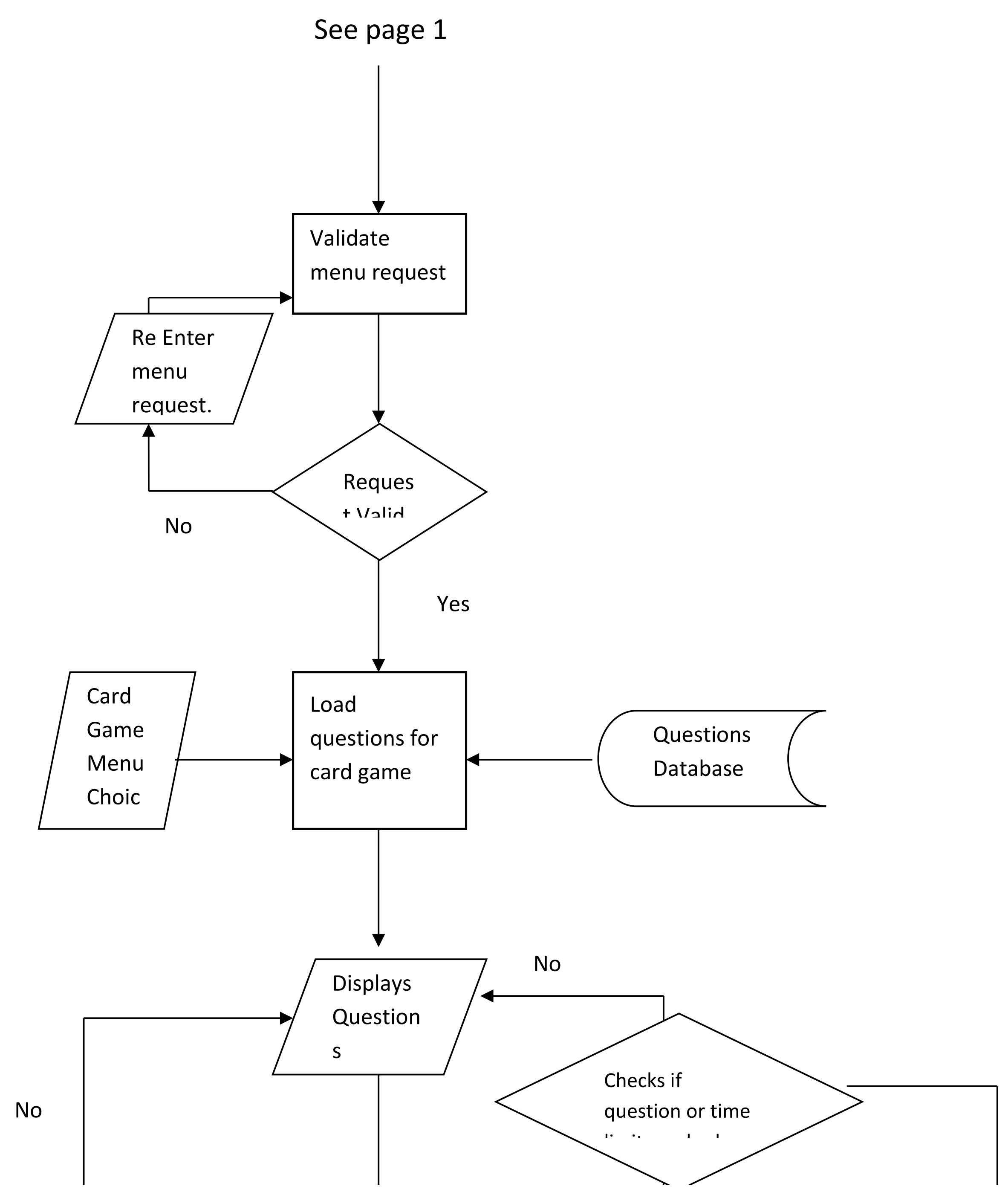

See page 1
Validate menu request
Re Enter menu request.
Reques t Valid
No
Yes
Test Menu Choic e
Load questions for test
Questions Database
Load Settings and Display Question
No
Checks if question or time limit reached
No

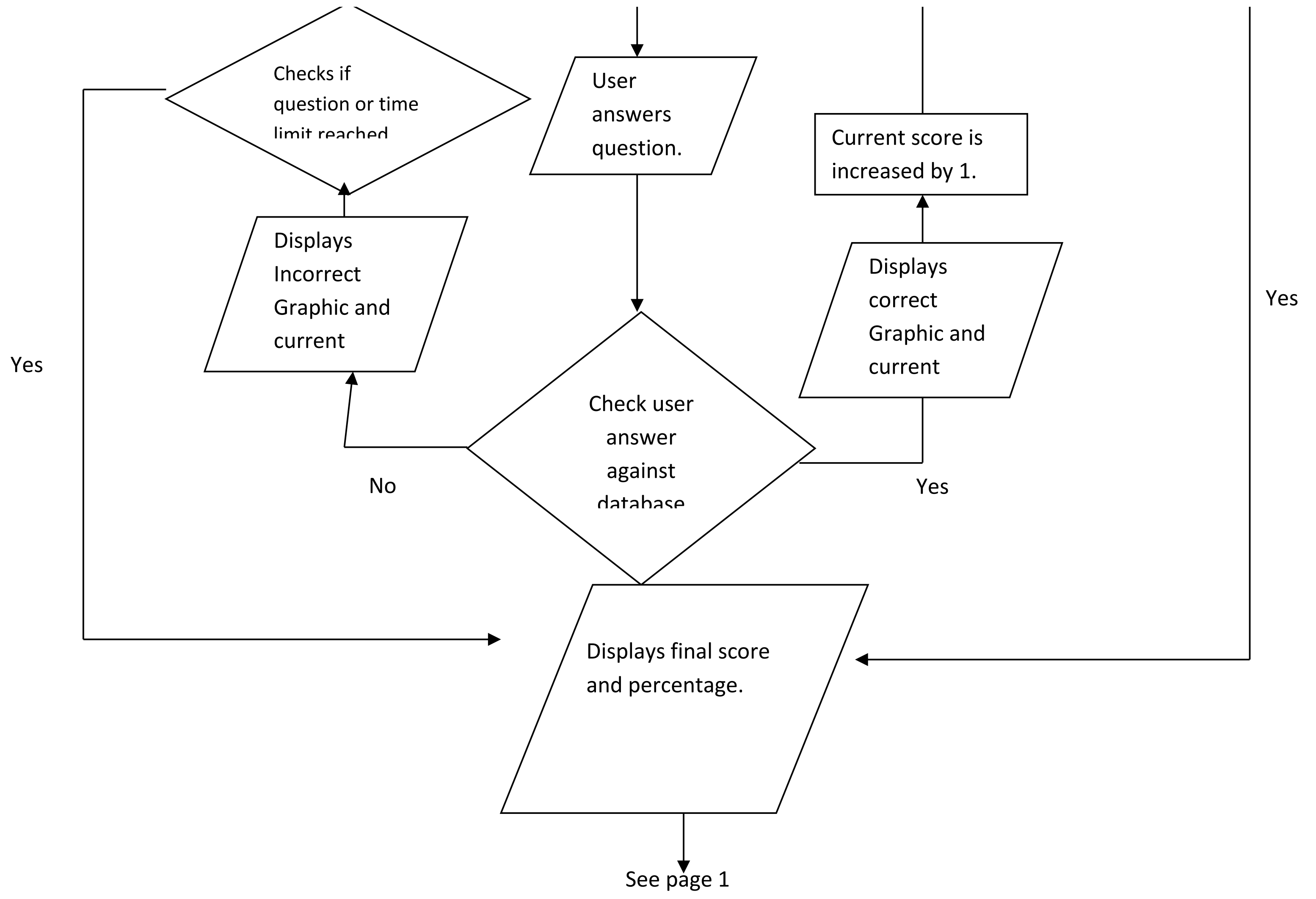

Checks if question or time limit reached
User answers question.
Current score is increased by 1.
Yes
Yes
Check user answer against database
No
Yes
Displays final score and percentage.
See page 1

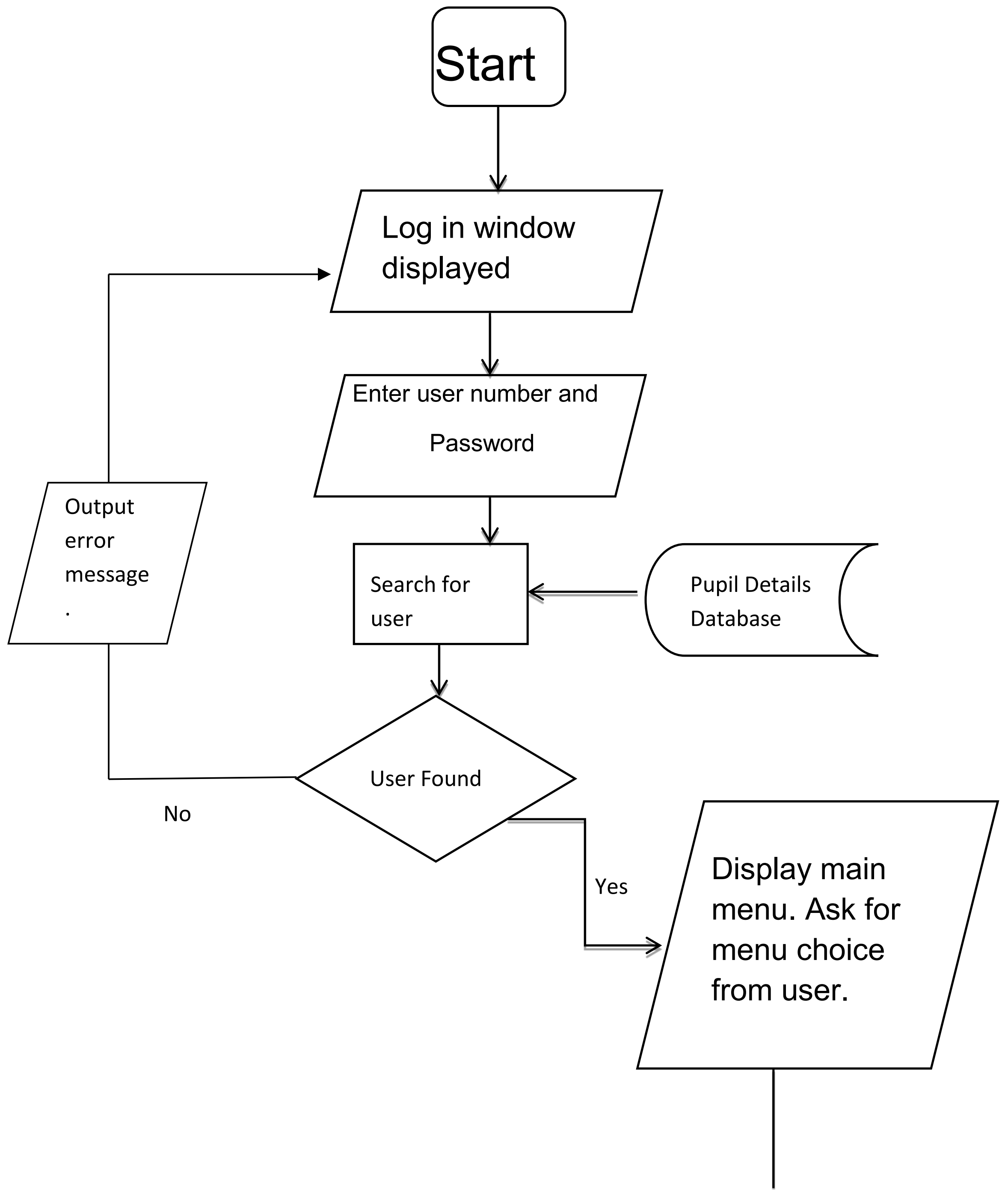

Start
Log in window displayed
Enter user number and Password
Search for user
Pupil Details Database
User Found
No
Output error message.
Yes
Display main menu. Ask for menu choice from user.

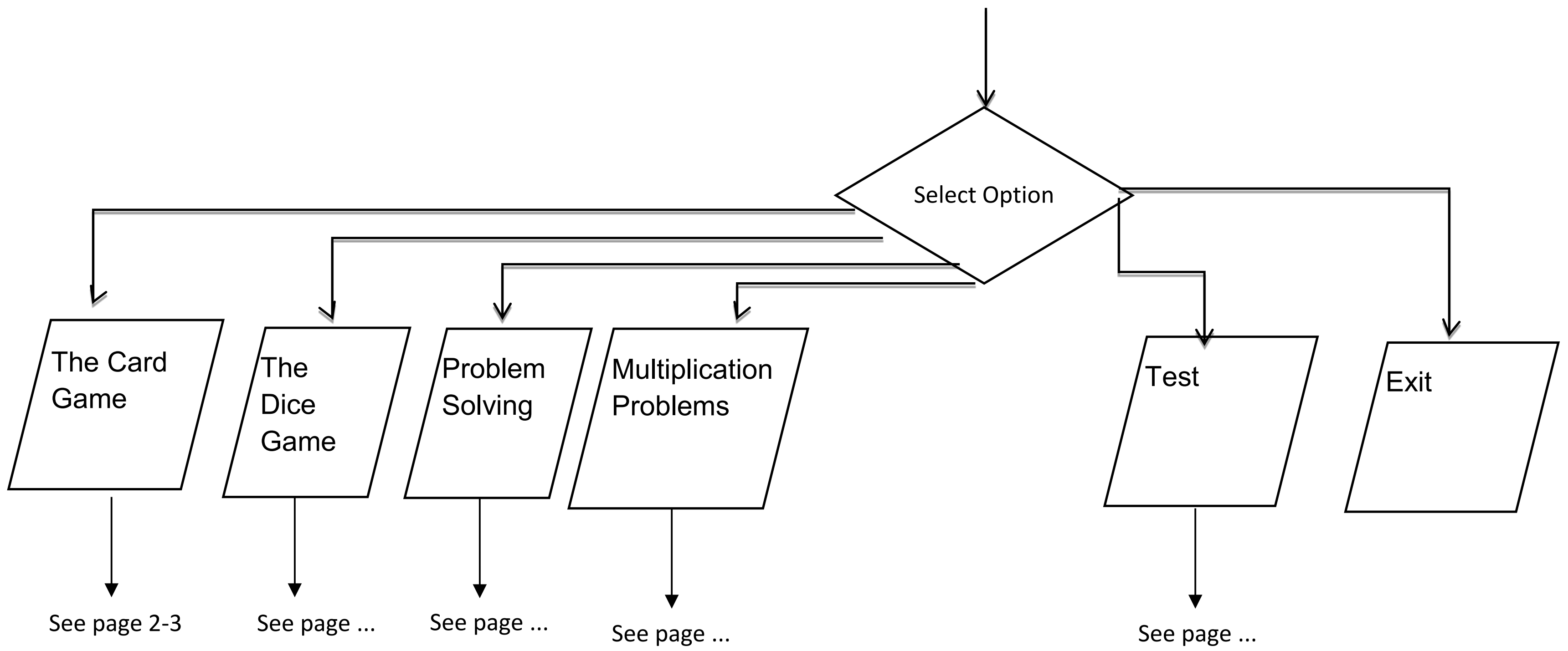


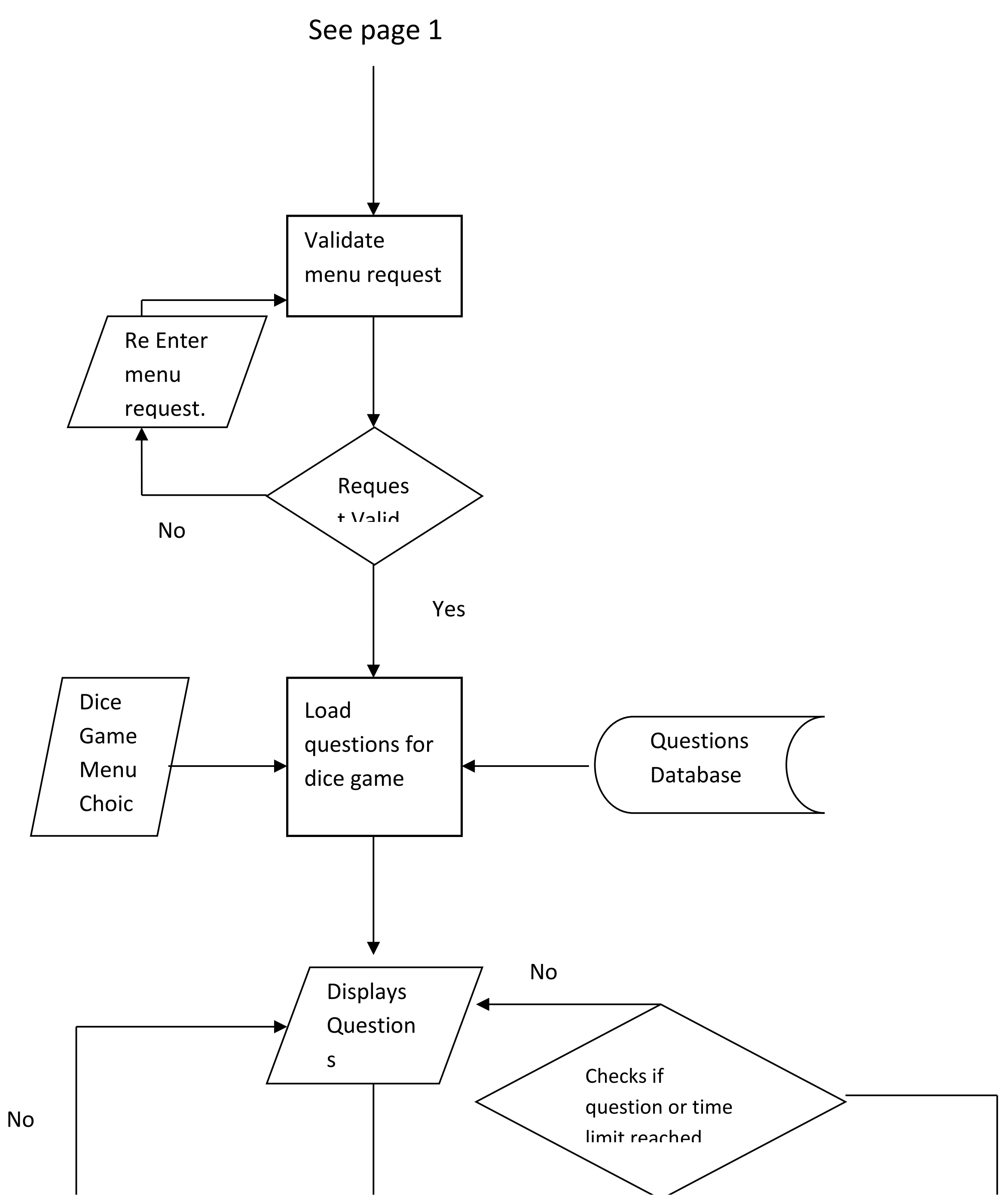
Select Option
The Card Game
The Dice Game
Problem Solving
Multiplication Problems
Test
Exit
See page 2-3
See page ...
See page ...
See page ...
See page ...

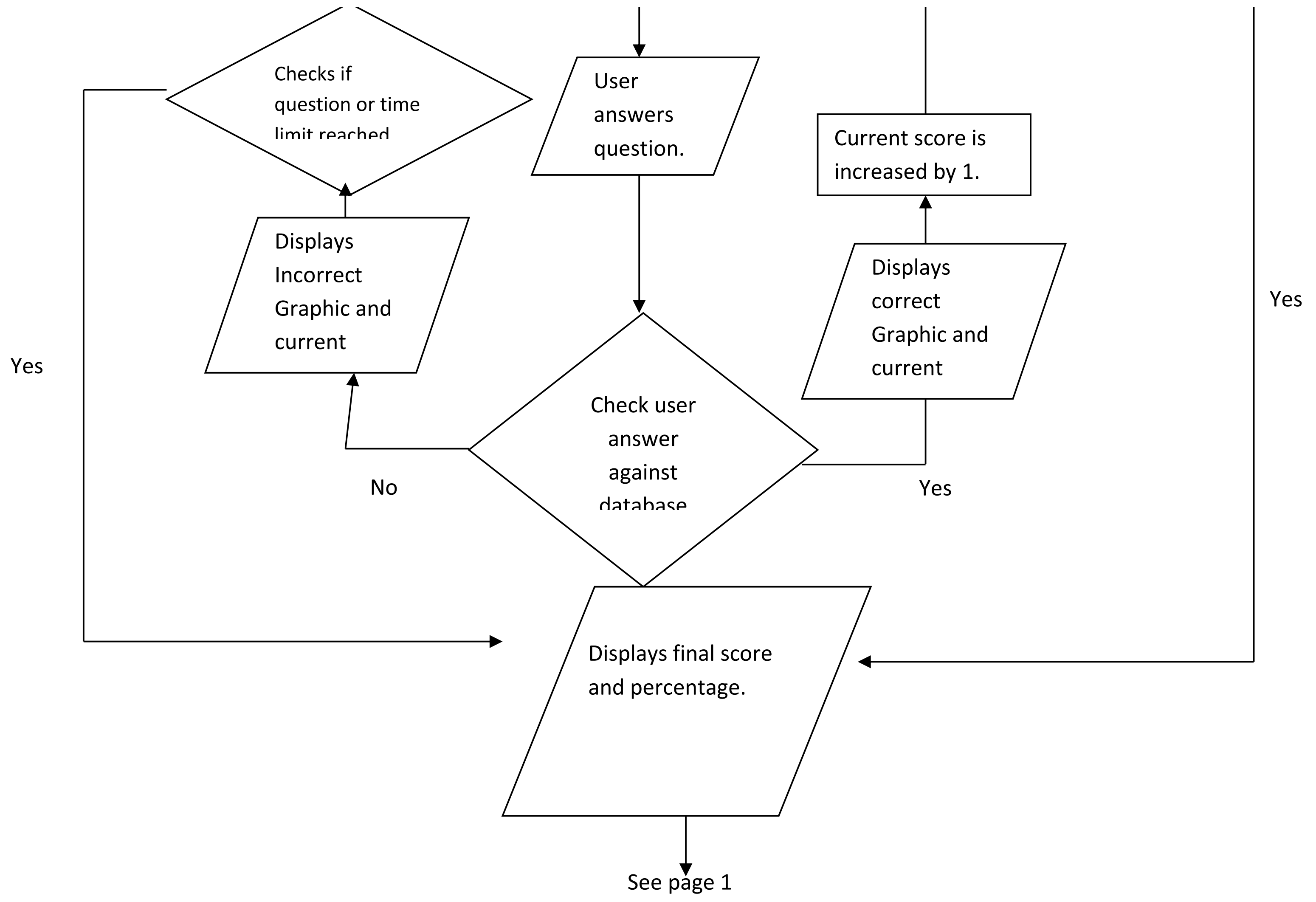

See page 1
Validate menu request
Re Enter menu request.
Reques t Valid
No
Yes
Card Game Menu Choic
Load questions for card game
Questions Database
Displays Question s
No
No
Checks if question or time

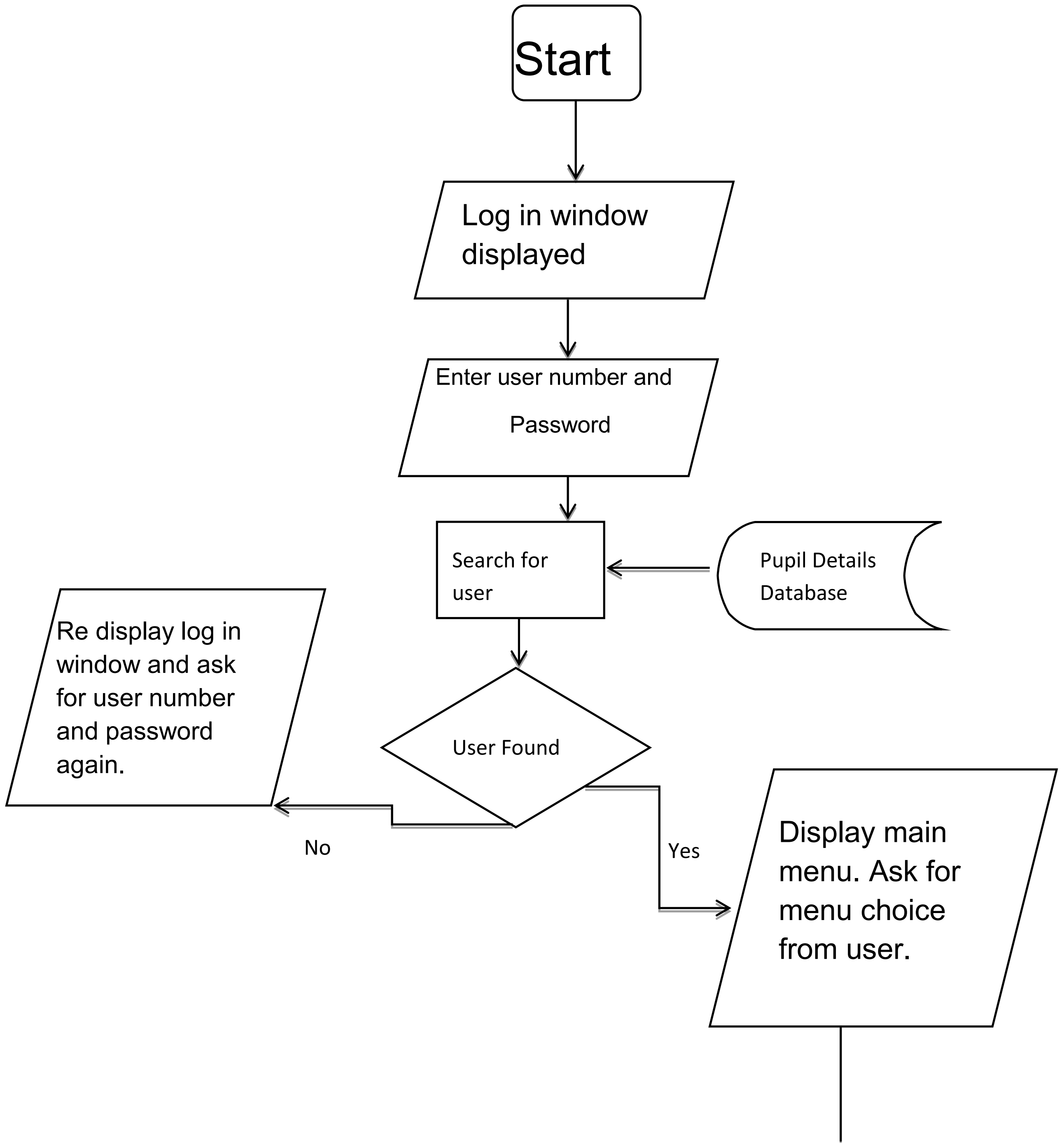

Checks if question or time limit reached
User answers question.
Current score is increased by 1.
Displays Incorrect Graphic and current
Displays correct Graphic and current
Yes
Yes
Check user answer against database
No
Yes
Displays final score and percentage.
See page 1

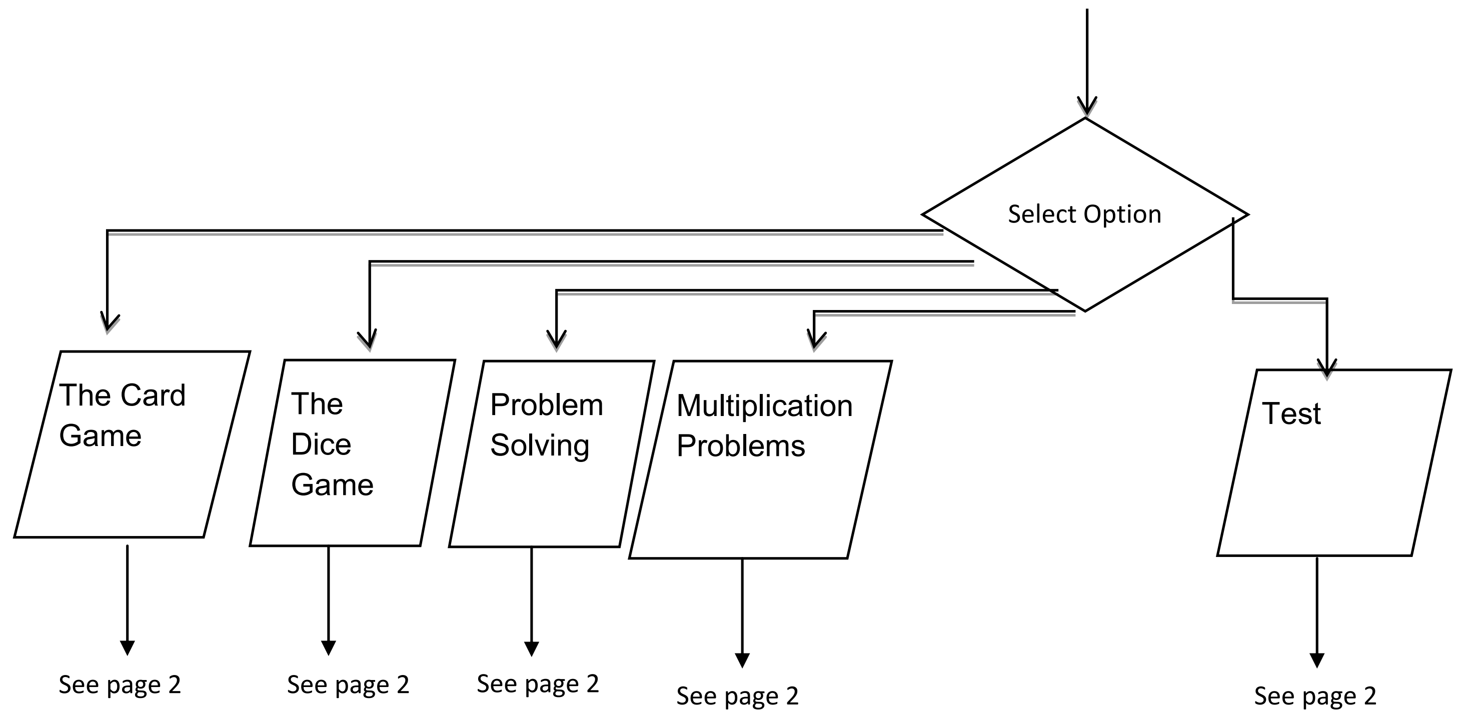

Start
Log in window displayed
Enter user number and Password
Search for user
Pupil Details Database
User Found
No
Yes
Output error message.
Display main menu. Ask for menu choice from user.

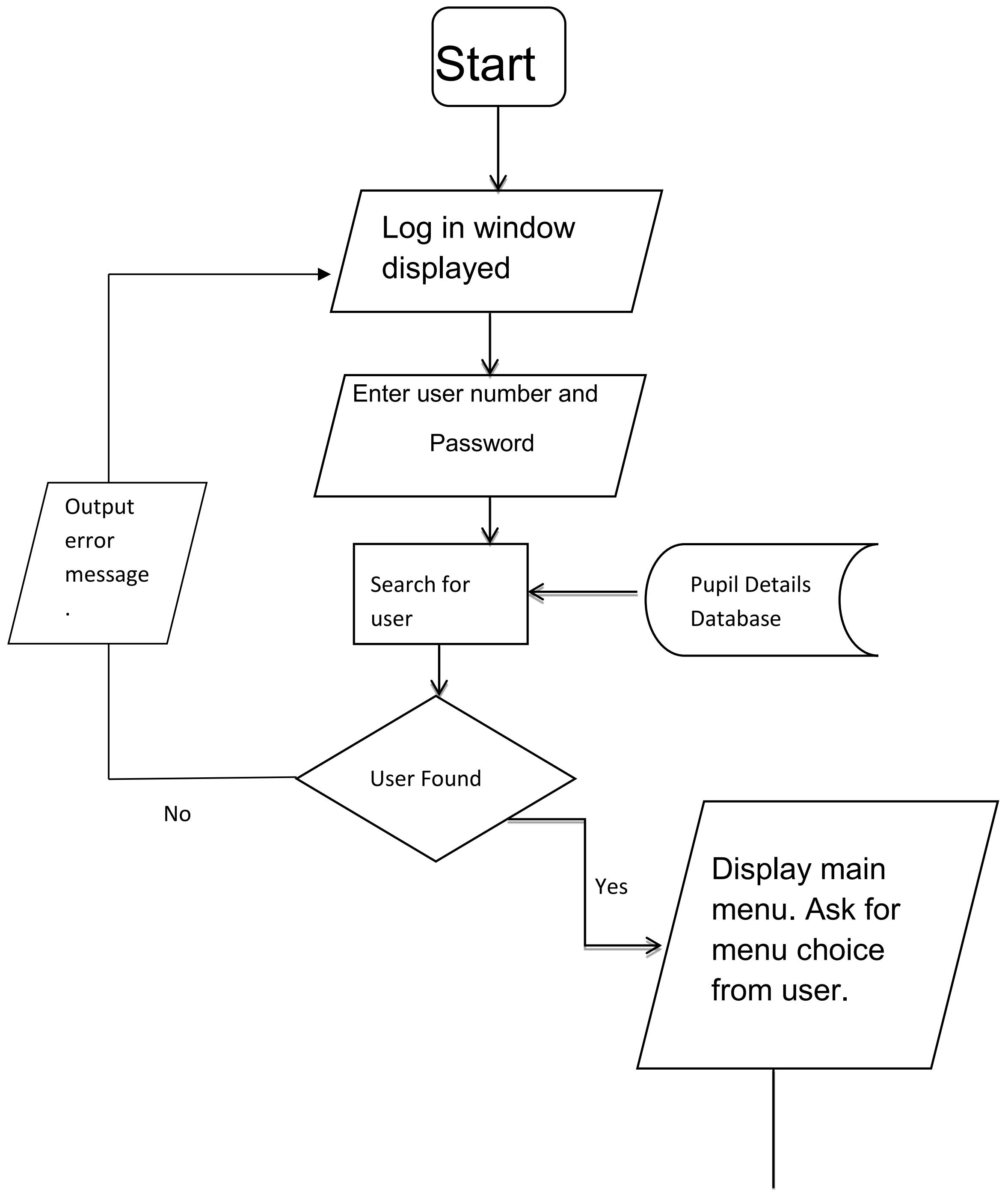

Select Option
The Card Game
The Dice Game
Problem Solving
Multiplication Problems
Test
Exit
See page 2-3
See page ...
See page ...
See page ...
See page ...

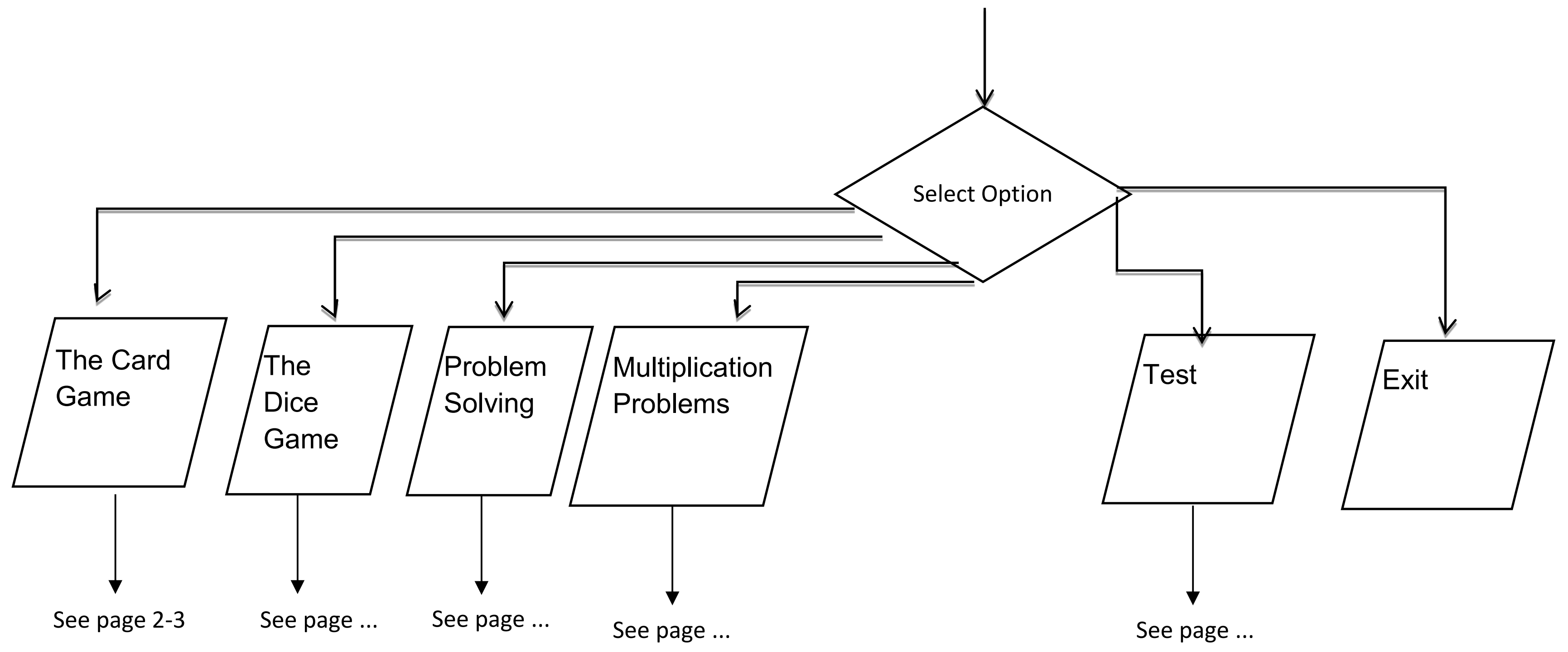

See page 1
Validate menu request
Re Enter menu request.
Reques t Valid
No
Yes
Dice Game Menu Choic
Load questions for dice game
Questions Database
Displays Question s
No
Checks if question or time limit reached
No


Checks if
question or time
limit reached
User
answers
question.
Current score is
increased by 1.
Displays
Incorrect
Graphic and
current
Displays
correct
Graphic and
current
Yes
Yes
Check user
answer
against
database
No
Yes
Displays final score
and percentage.
See page 1

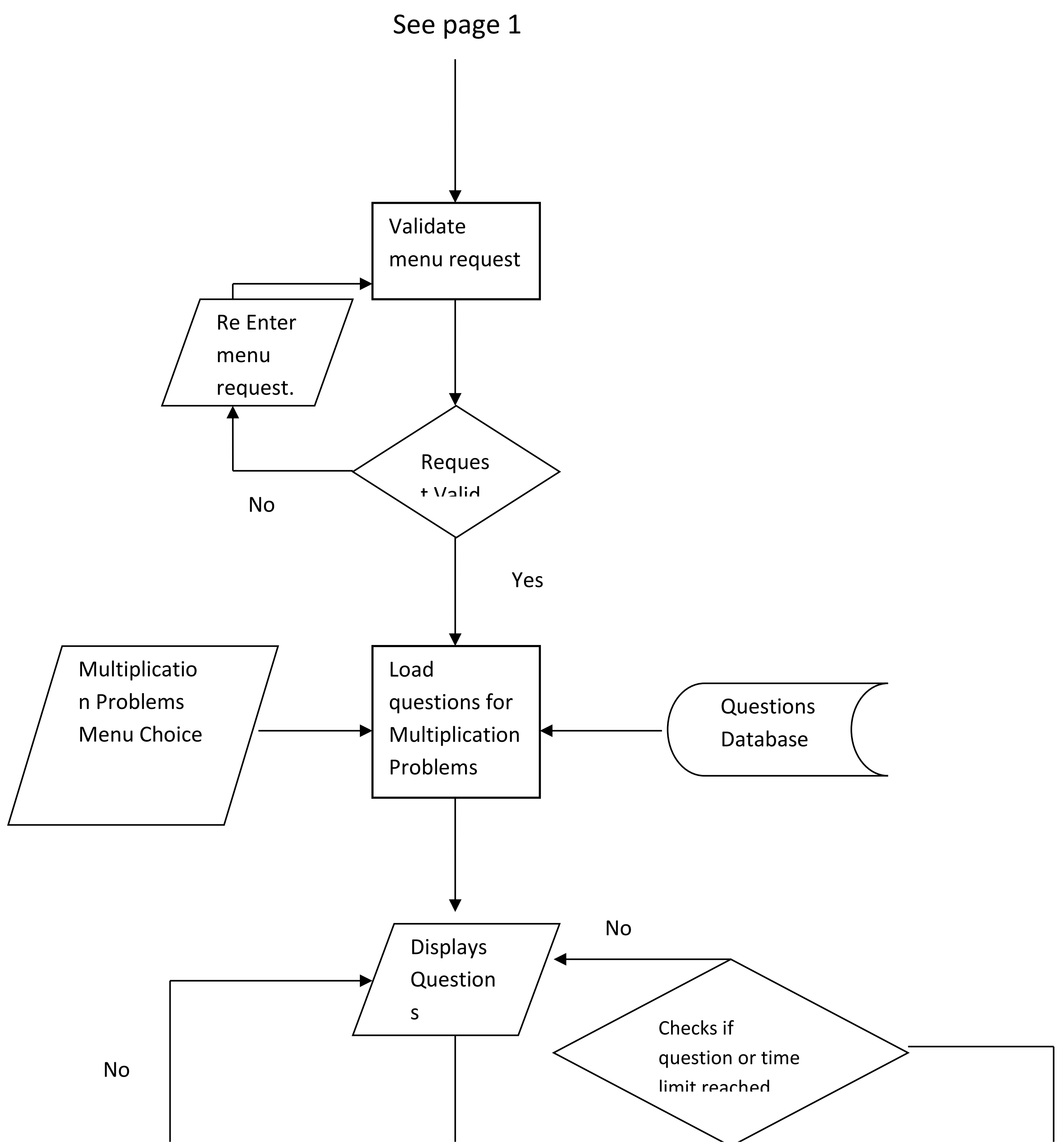

Start
Log in window displayed
Enter user number and Password
Search for user
Pupil Details Database
User Found
No
Yes
Re display log in window and ask for user number and password again.
Display main menu. Ask for menu choice from user.

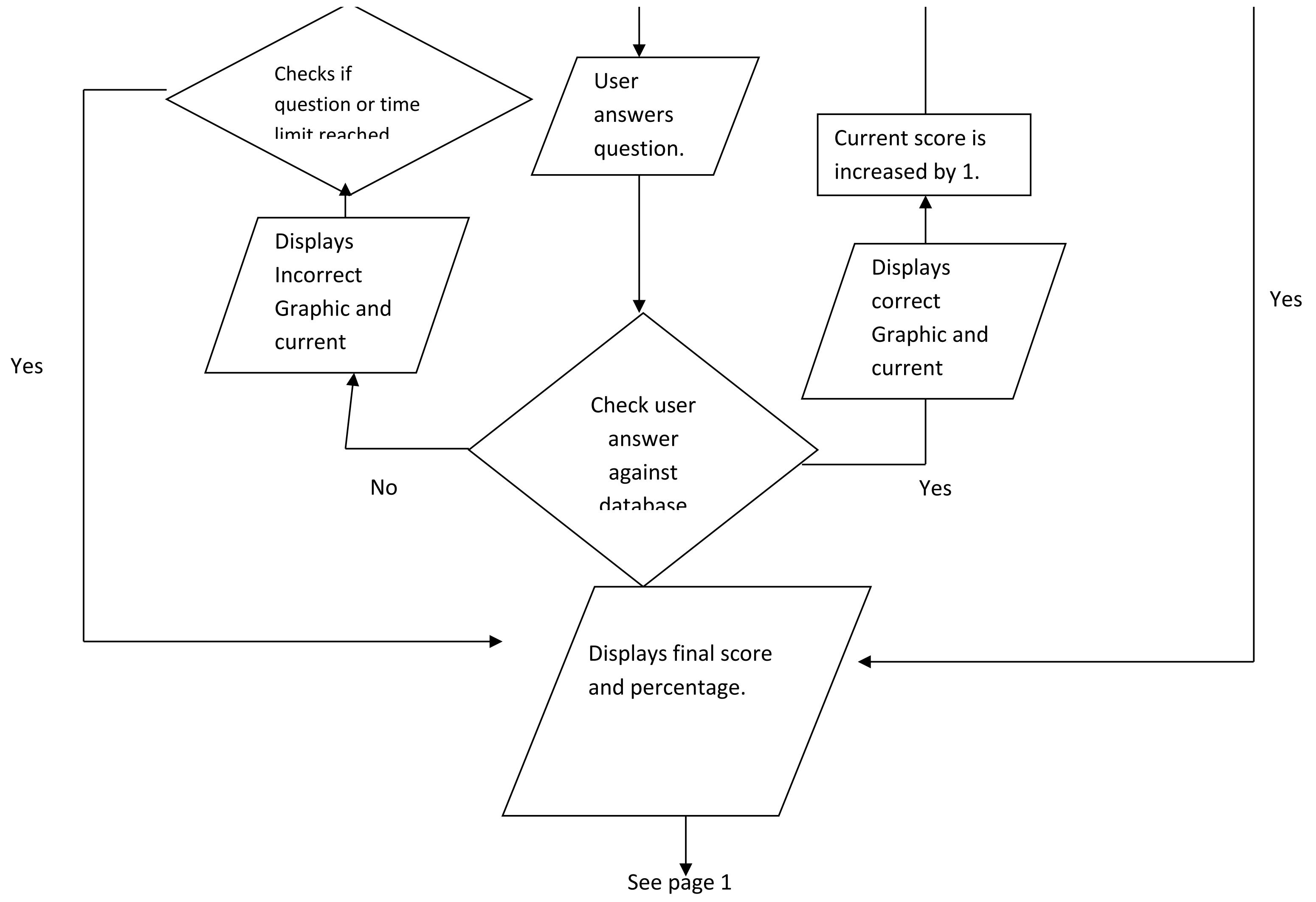

Select Option
The Card Game
The Dice Game
Problem Solving
Multiplication Problems
Test
See page 2
See page 2
See page 2
See page 2
See page 2

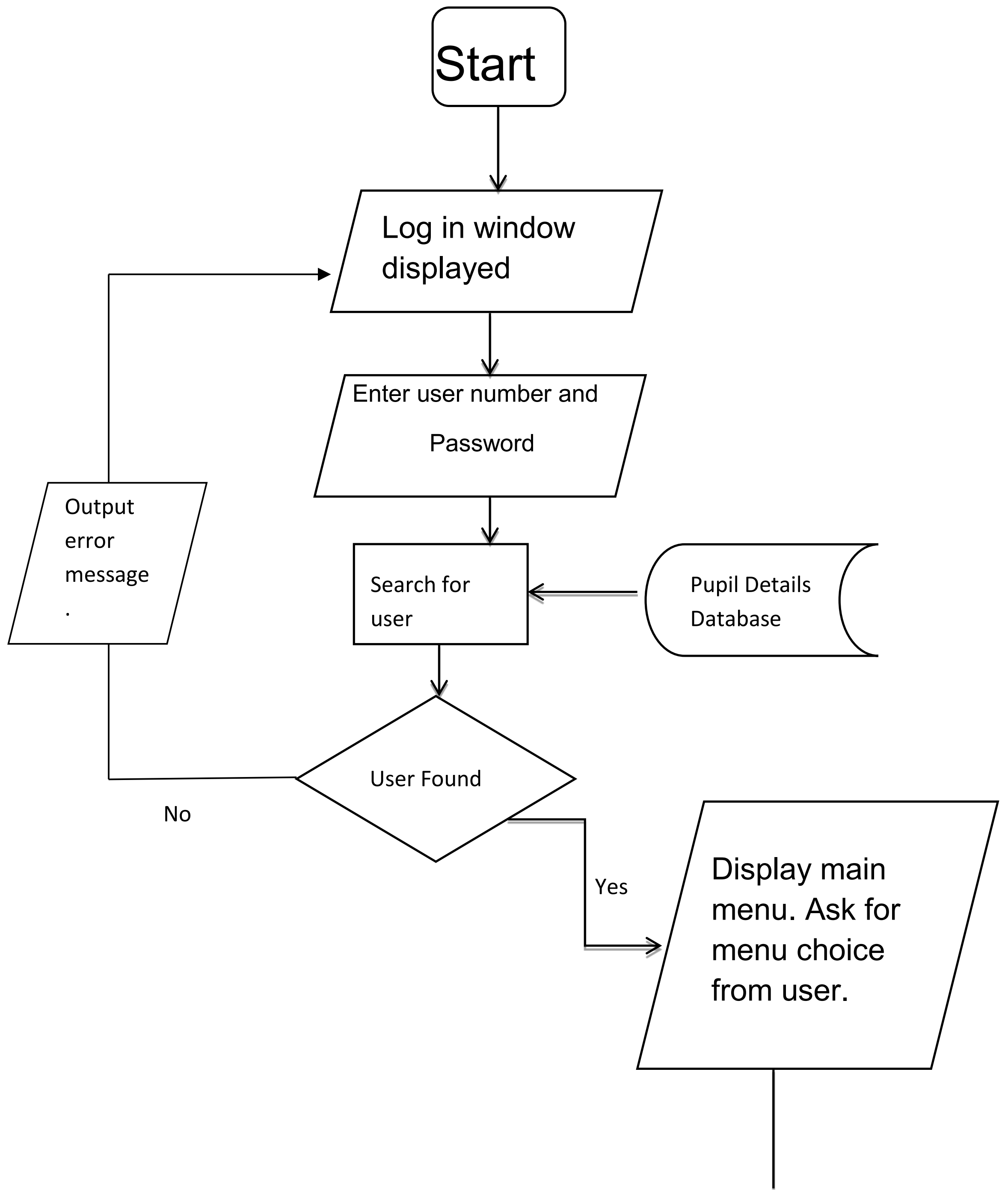

Start
Log in window displayed
Enter user number and Password
Search for user
Pupil Details Database
User Found
No
Output error message.
Yes
Display main menu. Ask for menu choice from user.

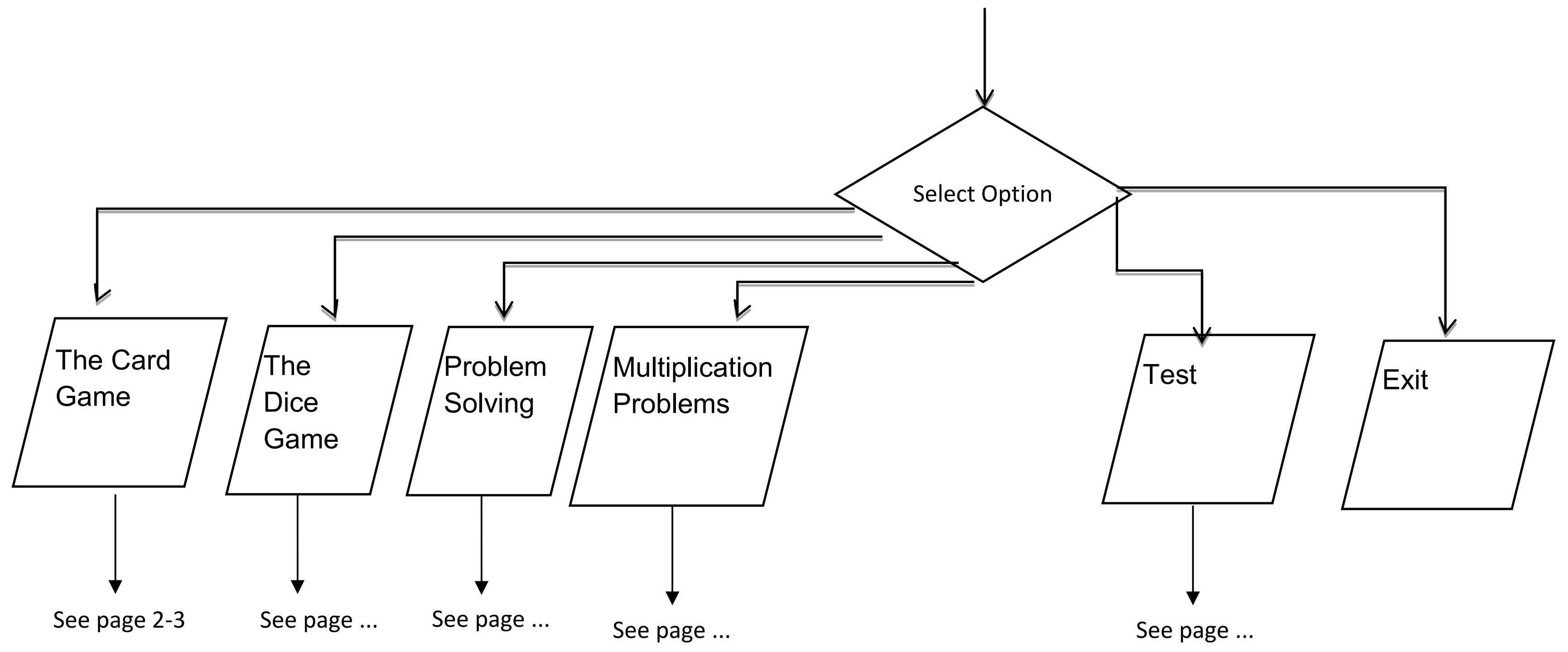


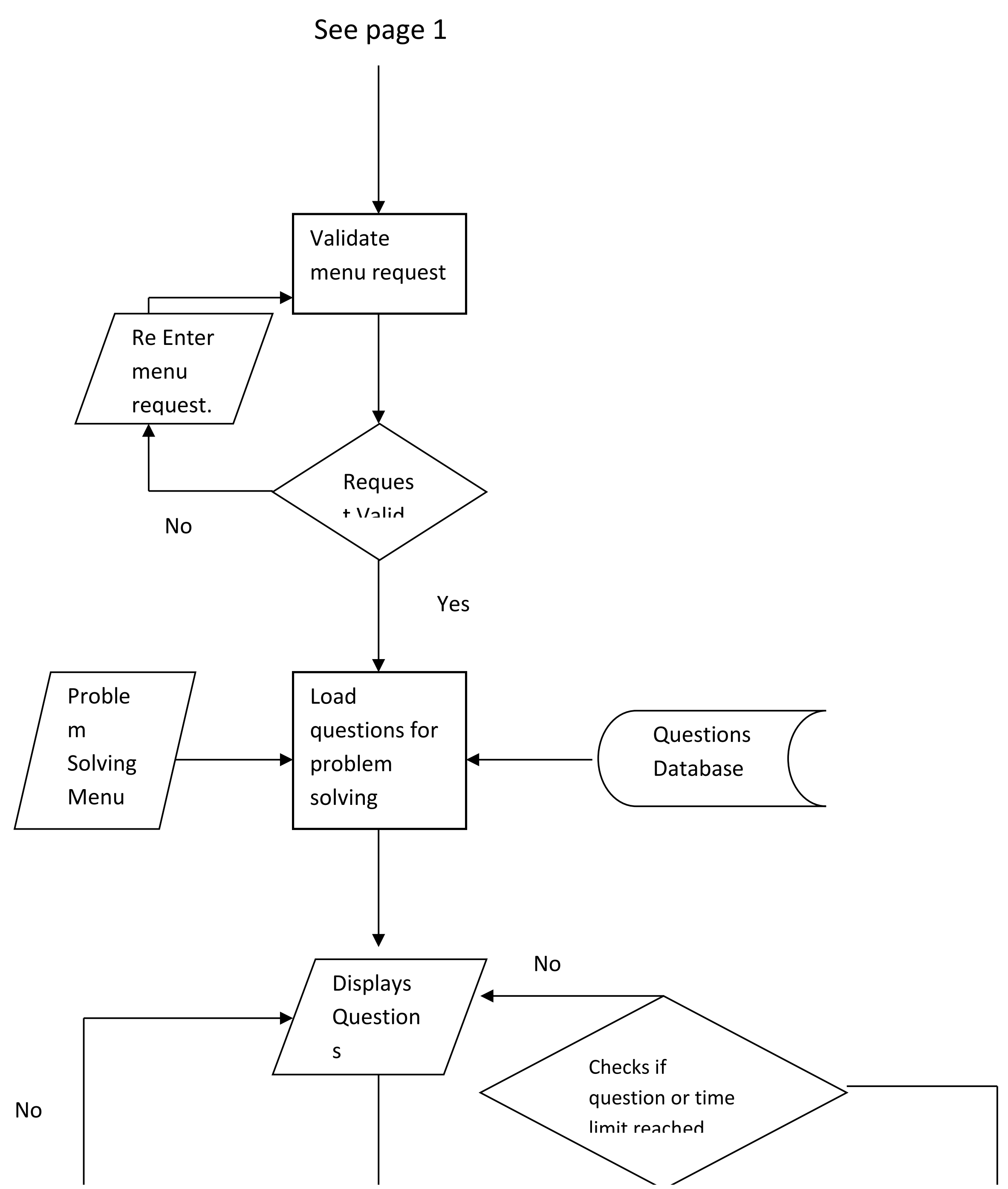
Select Option
The Card Game
The Dice Game
Problem Solving
Multiplication Problems
Test
Exit
See page 2-3
See page ...
See page ...
See page ...
See page ...

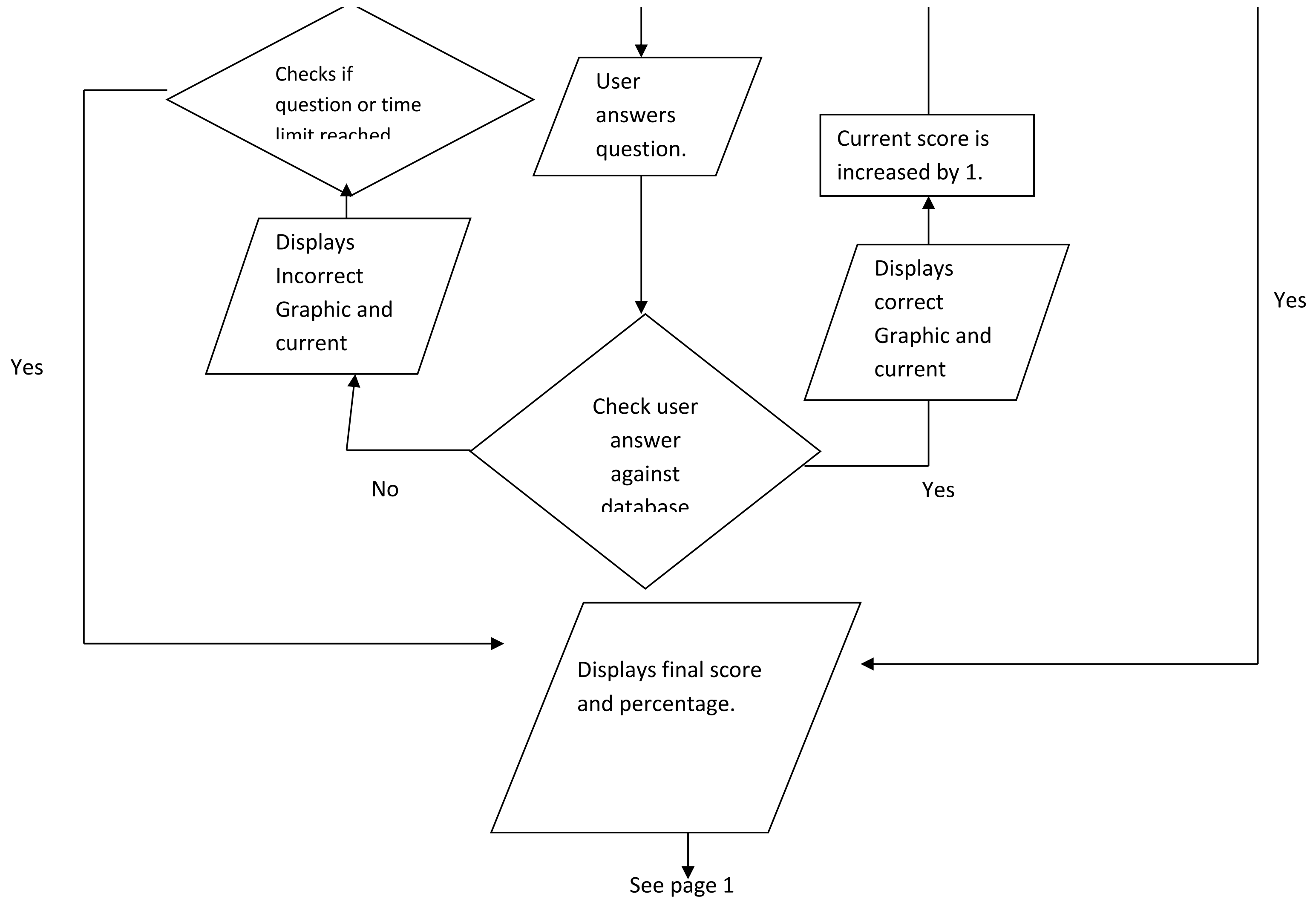

See page 1
Validate menu request
Re Enter menu request.
Reques t Valid
No
Yes
Multiplicatio n Problems Menu Choice
Load questions for Multiplication Problems
Questions Database
Displays Question s
No
Checks if question or time limit reached
No


Checks if
question or time
limit reached
User
answers
question.
Current score is
increased by 1.
Displays
Incorrect
Graphic and
current
Displays
correct
Graphic and
current
Yes
Yes
Check user
answer
against
database
No
Yes
Displays final score
and percentage.
See page 1


Start
Log in window displayed
Enter user number and Password
Search for user
Pupil Details Database
User Found
No
Output error message.
Yes
Display main menu. Ask for menu choice from user.


Select Option
The Card Game
The Dice Game
Problem Solving
Multiplication Problems
Test
Exit
See page 2-3
See page ...
See page ...
See page ...
See page ...


See page 1
Validate menu request
Re Enter menu request.
Reques t Valid
No
Yes
Proble m Solving Menu
Load questions for problem solving
Questions Database
Displays Question s
No
Checks if question or time limit reached
No


Checks if question or time limit reached
User answers question.
Current score is increased by 1.
Displays Incorrect Graphic and current
Displays correct Graphic and current
Yes
Yes
Check user answer against database
No
Yes
Displays final score and percentage.
See page 1