ELEMENTARY SCHOOL TUTORING SYSTEM

Table of contents

1. INTRODUCTION

1.1 Purpose of the Document

This document is primarily for use by the client, project management and developers.

1.2 Purpose of System

This system helps the student to improve their analytical skills, reading skills, critical thinking, problem solving on particular subjects. It also helps the student how to manage their time by allotting each subject a particular amount of time so that the student can manage the time to complete those questions. This helps mainly in boosting their confidence.

1.3 Scope of System

This system will ask questions to students one at a time and the next question will be allotted based upon the result of previous question. So they can enhance their ability in respective subjects and can improve grades in future tests.

This system tutors the students on different subjects and helps the students to improve their subject matter in which they are lacking the knowledge. The student has the choice to select any one subject from the list provided. So they can improve their knowledge in the given subjects.

1.4 References

Page 125, 126, and Appendix B: Glossary in Bernd Bruegge and Allen H. Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns and Java*, 3rd Edition, Prentice Hall, 2010 (ISBN 978-0136061250)

Page 847,848 Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw Hill, 2014 (ISBN 978-0078022128)

http://users.csc.calpoly.edu/~jdalbey/SWE/QA/nonfunctional.html

https://en.wikipedia.org/wiki/Functional_requirement

2. CURRENT SYSTEM LIMITATIONS

1. Some applications are platform dependent, so users with supported platform only can use the product.
2. Poor user interface in an application is very difficult to use and understand the system.
3. Current system produces the same level of questions irrespective of the students performance and give them result without analyzing how the student answering the questions.
4. In the current system tests are administered, they provide the grades manually and each subject is instructed by different teachers.
5. Unfortunately some applications will be stopped suddenly while in use. It will create inconvenience to the users.

3. Proposed System

3.1 Overview

This system mainly aims on the improvement of the student performance on particular subject. It provides good scale of achievement by providing questions that are specially focused on student ability level. So they can enhance their cognitive skills and build self confidence. Since it is a time based test student can learn how to manage time in test by allotting each question particular amount of time.

The system functions on the performance of students during the test based on the result of previous answers. So the difficulty of the next question will be given to the students based on the result.

3.2 Functional Requirements

3.2.1 Login Information

It provides the user, login information like username and password. If the user enters the correct information then it will display login successful else login failed.

3.2.2 Provides Subjects

It gives the information about the subjects that are available on the system. It provides with the list of subjects like math, reading, writing and student needs to select which subject he wants to perform.

3.2.3 Provides Questions

It provides with the different questions on the subject which the student has selected.

3.2.4 Provides Time

It displays the time on the top right of the screen so that the student is able to complete the test in a given time**.**

3.2.5 Provides Result

Based on the performance of the student in the test, it will evaluate the answers and provides results in the form of grading like A, A-, B, B-, and C.

3.2.6 Updating Subject Matter

If anything is needed to add or delete or make changes in the subject, then it will update with that information.

3.2.7 Updating Test questions

Based on the performance of the student in the test it will update the questions either in a difficult way or easy way. Questions are assigned after the evaluation of each previous question.

3.2.8 Updating Student List

It will update the students list daily to check whether the student is absent or present or any new student is added or removed.

3.3 Non-functional Requirements

3.3.1 Performance

This software should perform the same way irrespective to its Operating System environments.

3.3.2 Portability

As this software is to work on multiple platforms portability is an essential attribute and we ensure this by using JAVA as our programming language.

3.3.3 Security

This software keeps the information of a user in a secure way.

3.3.4 Accuracy

The capability of the software product to provide the right or agreed results or effects with the needed degree of precision.

3.3.5 Adaptability

This software is adapted for different specified environments without applying actions.

3.4 System Models

3.4.1 Scenarios

Scenario name	Create Teacher Account Successful
Participating actor instances	Teacher

Flow of events

1. A teacher activates the “Create Teacher Account” function of the system.
2. The system prompts the Teacher for their information (Name, Teacher ID, Phone and Email Address).
3. The Teacher enters all the requested information.
4. The system confirms that a Teacher with this information does not already exist in the system and creates a new account using the Teacher’s information.
5. The system prompts the Teacher for their password indicating the minimum password standards.
6. The Teacher enters their password.
7. The system confirms that the Teacher’s new password has met the standards and is accepted.

Scenario name	Create Student Account Successful
Participating actor instances	Student

Flow of events

1. A teacher activates the “Create Student Account” function of the system.
2. The system prompts the Student for their information (Name, Student ID, Class and Email Address).
3. The Student enters all the requested information.

4. The system confirms that a Student with this information does not already exist in the system and creates a new account using the Student's information.
5. The system prompts the Student for their password indicating the minimum password standards.
6. The Student enters their password.
7. The system confirms that the Student's new password has met the standards and is accepted.

Scenario name	Addition of Subject to Subject list
Participating actor instances	Teacher

Flow of events

1. A teacher activates the "Add Subject" function of the system.
2. The system prompts the Teacher for the name of the subject.
3. The Teacher enters all the requested information.
4. The system confirms that a subject with this name does not already exist in the system and creates a new subject.

Scenario name	Deletion of Subject from the Subject list
Participating actor instances	Teacher

Flow of events

1. A teacher activates the "Delete Subject" function of the system.
2. The system prompts the Teacher to select the subject for deletion.
3. The Teacher enters the name that has to be deleted.
4. The system confirms and deletes the subject.

Scenario name	Addition of Questions in a Subject
Participating actor instances	Teacher

Flow of events

1. A teacher activates the “Add Questions” function of the system.
2. The system prompts the Teacher to choose the subject from the subject list.
3. The Teacher chooses the subject from the subject list.
4. The system confirms and prompts the Teacher to add question and multiple choices for that question.
5. The Teacher enters the question, allots multiple choices and sets the correct answer for that question.
6. The system confirms the question and update the questions in the selected subject

Scenario name	Deletion of Questions in a Subject
Participating actor instances	Teacher

Flow of events

1. A teacher activates the “Delete Questions” function of the system.
2. The system prompts the Teacher to select the subject from the subject list.
3. The Teacher chooses the subject from the subject list.
4. The system confirms and prompts the Teacher to select the questions.
5. The Teacher selects the questions that have to be deleted.
6. The system confirms the selected list of questions and deletes the questions.

Scenario name	Generation of Result
Participating actor instances	Student

Flow of events

1. A Student activates the “Generate Result” function of the system.
2. The system prompts the Student to select the current test result or overall result of all tests.
3. The Student chooses the required option.
4. The system confirms the Student choice and generates the result.
5. The Teacher selects the questions that have to be deleted.
6. The system confirms the selected list of questions and deletes the questions.

Scenario name	Generation of Student Result
Participating actor instances	Teacher

Flow of events

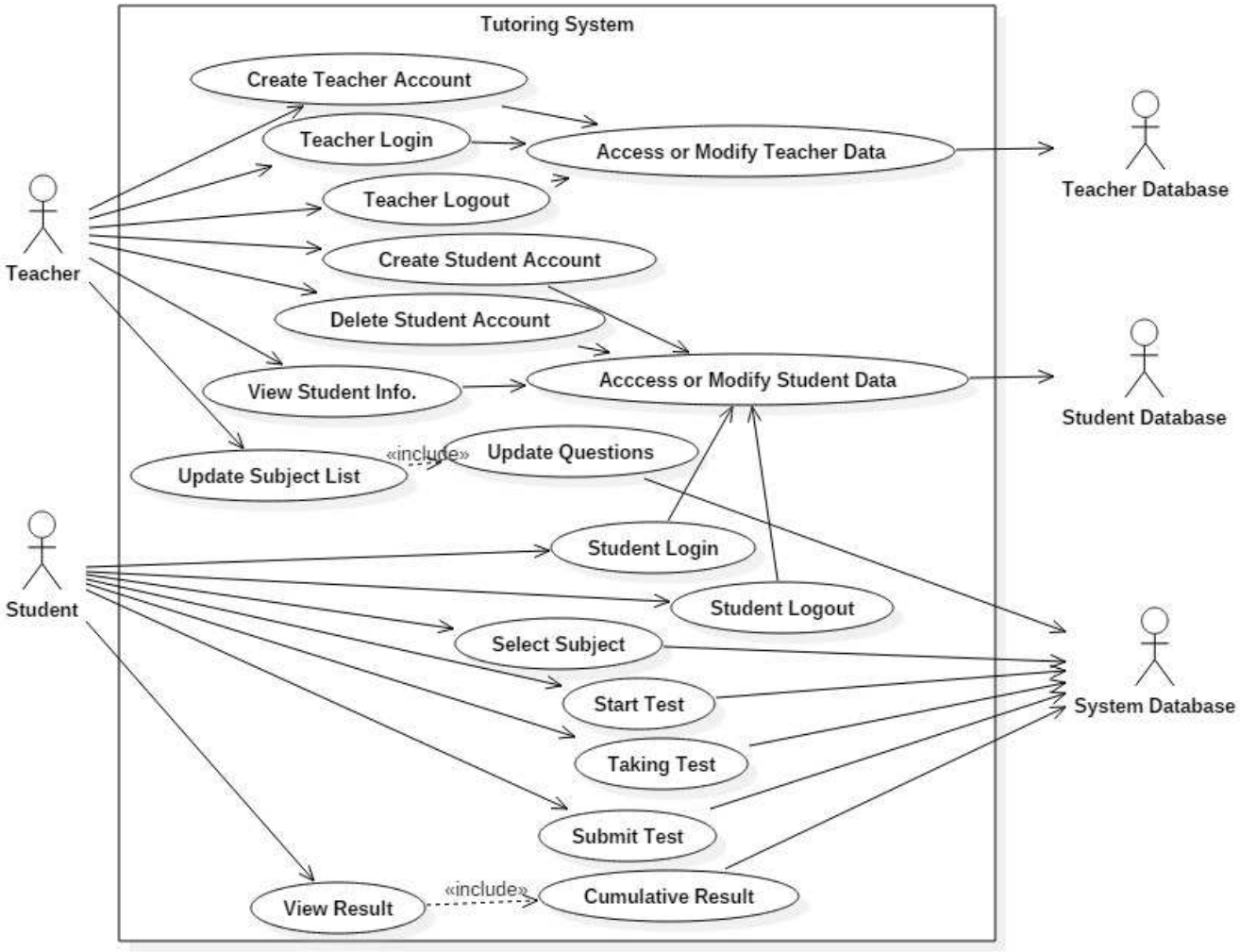
1. A Teacher activates the “Generate Student Result” function of the system.
2. The system prompts the Teacher to select the required Student result from the list of Students.
3. The Teacher selects the required Student result.
4. The system confirms the request and generates the result of the selected Student.

Scenario name	Deletion of Student Account
Participating actor instances	Teacher

Flow of events

1. A Teacher activates the “Delete Student Account” function of the system.
2. The system prompts the Teacher to select the Student for deletion.
3. The Teacher selects the Student from the list of Students.
4. The system confirms and deletes the Student Account.

3.4.2 Use Case Model



Use case name	Create Teacher Account
Participating actor instances	Teacher

Flow of events

Main Flow:

1. The use case begins to create account for Teacher.
2. The Teacher enters all the data like Name, Teacher ID, phone and email id.
3. The system accepts data and creates the Teacher account successfully.
4. The use case ends.

Alternate path #1:

3.a. The Teacher unable to create Teacher account and sends the information to the system database.

3.b. Then system database double checks the information and helps the Teacher to create account.

Entry condition:	the Teacher must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the Teacher to try the operation again.

Use case name	Teacher Login
Participating actor instances	Teacher

Flow of events

Main Flow:

1. A Teacher activates the “Teacher Login” function of the system.
2. The system prompts the Teacher to enter their email address and password. The password shall be x-ed out for security.
3. The Teacher enters their user information.
4. The system confirms that a Teacher account matching this information exists in the Teacher Database.
5. The system allows access to the Teacher account and displays information regarding Teacher. Use case ends.

Alternate Flow #1:

4a. the system cannot confirm that a Teacher account matching this login information exists.

4a1. the system prompts the Teacher to either try to login again, have their password emailed to them or create a new account.

4a2. the Teacher requests to try to enter their login info again.

4a3. the system returns to Step 2.

Alternate Flow #2:

4a2a. The Teacher requests their password to be sent to their email.

4a2a1. the system retrieves the Teachers password from the Teachers Database, emails it to the Teacher mail id or sends a password to the mobile. Use case ends.

Alternate Flow #3:

4a2b. The Teacher requests to create a new account.

4a2b1. the system prompts the Teacher to enter information for a new account.

4a2b2. The Teacher enters their account information.

4a2b3. the system proceeds to create a customer account includes Create Teacher account use case.

4a2b4. the system successfully creates a new customer account.

4a2b5. the system returns to Step 2.

Alternate Flow #4:

4a2b4a. the system does not successfully create a new Teacher account.

4a2b4a1. the system indicates to the Teacher to school service to resolve problem. Use case ends.

Entry condition:	must be Teacher to login
Exit condition:	none
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the Teacher to try the operation again.

Use case name	Teacher Logout
Participating actor instances	Teacher

Flow of events

Main Flow:

1. The use case begins when Teacher decides to logout.

2. The application sends a request to Teacher database that the Teacher wants to logout.
3. The Teacher database Logs out the Teacher.
4. The application informs the Teacher that Teacher has been logged out.
5. The use case ends.

Entry condition:	the Teacher must login.
Exit condition:	the use case ends in success, the Teacher is no longer Logged User. Teacher can login again.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the Teacher to try the operation again.

Use case name	View Student Info
Participating actor instances	Teacher

Flow of events

Main Flow:

1. The use case begins when teachers decides to view Student Info.
2. The teacher can view the Student Info and tests results of Student.
3. The use case ends.

Alternate path #1:

2.a. The teacher unable to view Student Info then sends the information to the Student Database.

2.b. Then Student database double checks the information provided by teacher and helps the teacher to view the result of Student.

Entry condition:	the teacher must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the Teacher to try the operation again.

Use case name	Create Student Account
Participating actor instances	Teacher

Flow of events

Main Flow:

1. The use case begins when teachers decides to create account for Student.
2. The teacher enters all the data of Student like Name, Student ID, Class and Student email id.
3. The system accepts data and creates the Teacher account successfully.
4. The use case ends.

Alternate path #1:

3. a. The system unable to create Student account then sends the information to the system database.

3. b. Then system database double checks the information by the Student Database and helps the teacher to create account.

Entry condition:	the teacher must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the teacher to try the operation again.

Use case name	Student login
Participating actor instances	Student

Flow of events

Main Flow:

1. A Student clicks the “Student Login” function of the system.
2. The system prompts the Student to enter their Student ID and Password. The password shall be x-ed out for security.
3. The Student enters the information.
4. The system confirms that the Student account matching this information exists in the Student Database. Includes Access/Modify Student Database use case.
5. The system allows access to the Student account and displays information regarding Student. Use case ends.

Alternate Flow #1:

4a. the system cannot confirm that a Student account matching this login information exists.

4a1. the system prompts the Student to either try to login again, have their password emailed to them or create a new account.

4a2. The Student requests to try to enter their login info again.

4a3. the system returns to Step 2.

Alternate Flow #2:

4a2a. The Student requests their password to be sent to their email.

4a2a1. the system retrieves the Student password from the Student Database, emails it to the Student mail id includes Access or Modify Student Database use case. Use case ends.

Alternate Flow #3:

4a2b. The Student requests to create a new account.

4a2b1. the system prompts the Student to enter information for a new account.

4a2b2. The Student enters their account information.

4a2b3. the system proceeds to create a Student account include Create Student Account use case.

4a2b4. the system successfully creates a new Student account.

4a2b5. the system returns to Step 2.

Alternate Flow #4:

4a2b4a. the system does not successfully create a new Student account.

4a2b4a1. the system prompts the Student to take assistance of teacher to resolve problem. Use case ends.

Entry condition:	must be Student to login
Exit condition:	none
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the Student to try the operation again.

Use case name	Student Logout
Participating actor instances	Student

Flow of events

Main Flow:

1. The use case begins when Student decides to logout.
2. The application sends a request to Student Database that the user wants to logout.
3. The Student Database System Logs Out the User.
4. The application informs the user that he has been logged out.
5. The use case ends.

Entry condition:	the Student must login.
Exit condition:	the use case ends in success, the user is no longer Logged User. He can login again.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the Student to try the operation again.

Use case name	Update Subject List
Participating actor instances	Teacher

Flow of events

Main Flow:

1. The use case begins to update the subjects.
2. The system prompts the teacher to select the subjects from the list provided.

3. After selecting the subject from the list the teacher needs to update the subjects.
4. The Teacher can add or delete the subject or can make any changes in the subject list.
5. The use case ends.

Entry condition:	the teacher must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the teacher to try the operation again.

Use case name	Update Questions
Participating actor instances	Teacher

Flow of events

Main Flow:

1. The use case begins to update the questions.
2. The system prompts the teacher to select the question from the subject that has to be changed.
3. After selecting the Teacher can add questions or delete questions or modify questions or make suitable corrections in the multiple choices in the answers for a question in a selected subject from the subject list.
4. The use case ends.

Entry condition:	the teacher must login.
Exit condition:	none.

Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the teacher to try the operation again.

Use case name	Start Test
Participating actor instances	Student

Flow of events

Main Flow:

1. The use case begins to start test.
2. The system prompts the student to take the test.
3. The use case ends.

Alternate path #1:

1. The student unable to take the test then the information is send to the system database.
2. The information sent to the system database verifies and allows the student to start the test.

Entry condition:	the student must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the student to try the operation again.

Use case name	Submit Test
Participating actor instances	Student

Flow of events

Main Flow:

1. The use case begins to submit test.
2. After completion of the test, the system prompts the student to submit the test.
3. The use case ends.

Entry condition:	the student must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the student to try the operation again.

Use case name	View Result
Participating actor instances	Student

Flow of events

Main Flow:

1. The use case begins to view result.
2. After submission of the test, the system prompts the student to view the result.
3. The system provides the grade for the current test and also overall grades that the student has accumulated
4. The use case ends.

Entry condition:	the student must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the student to try the operation again.

Use case name	Select Subject
Participating actor instances	Student

Flow of events

Main Flow:

1. The use case begins to select subject.
2. The system prompts the student to select the subject from the list provided by the system.
3. The student selects the subject from the list.
4. The system confirms the subject.
5. The use case ends.

Entry condition:	the student must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the student to try the operation again.

Use case name	Access/Modify Teacher data
Participating actor instances	Teacher

Flow of events

Main Flow:

1. The use case begins to access/modify teacher data.
2. To change the information to the teacher account is done and teacher can access the account information.
3. The system prompts for the changes that have to be saved.
4. Teacher saves the changes made.
5. The use case ends.

Entry condition:	the teacher must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the teacher to try the operation again.

Use case name	Access/Modify Student data
Participating actor instances	Student, Teacher

Flow of events

Main Flow:

1. The use case begins to access/modify student data.

2. To change the information to the Student account is done by either student or teacher and either student or teacher can access the account information.
3. The system prompts for the changes that have to be saved.
4. Student or Teacher saves the changes made.
5. The use case ends.

Entry condition:	the student or teacher must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the student or teacher to try the operation again.

Use case name	Delete Student Account
Participating actor instances	Teacher

Flow of events

Main Flow:

1. The use case begins to delete student account.
2. The system prompts the teacher to select the student account which has to be deleted.
3. After selecting and deleting the account.
4. The use case ends.

Entry condition:	the teacher must login.
Exit condition:	none.

Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the teacher to try the operation again.

Use case name	Taking Test
Participating actor instances	Student

Flow of events

Main Flow:

1. The use case begins to taking test.
2. The system prompts the student to take the test.
3. The Student takes the test of the selected subject.
4. The use case ends.

Entry condition:	the student must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the student to try the operation again.

Use case name	View Cumulative Result
Participating actor instances	Student

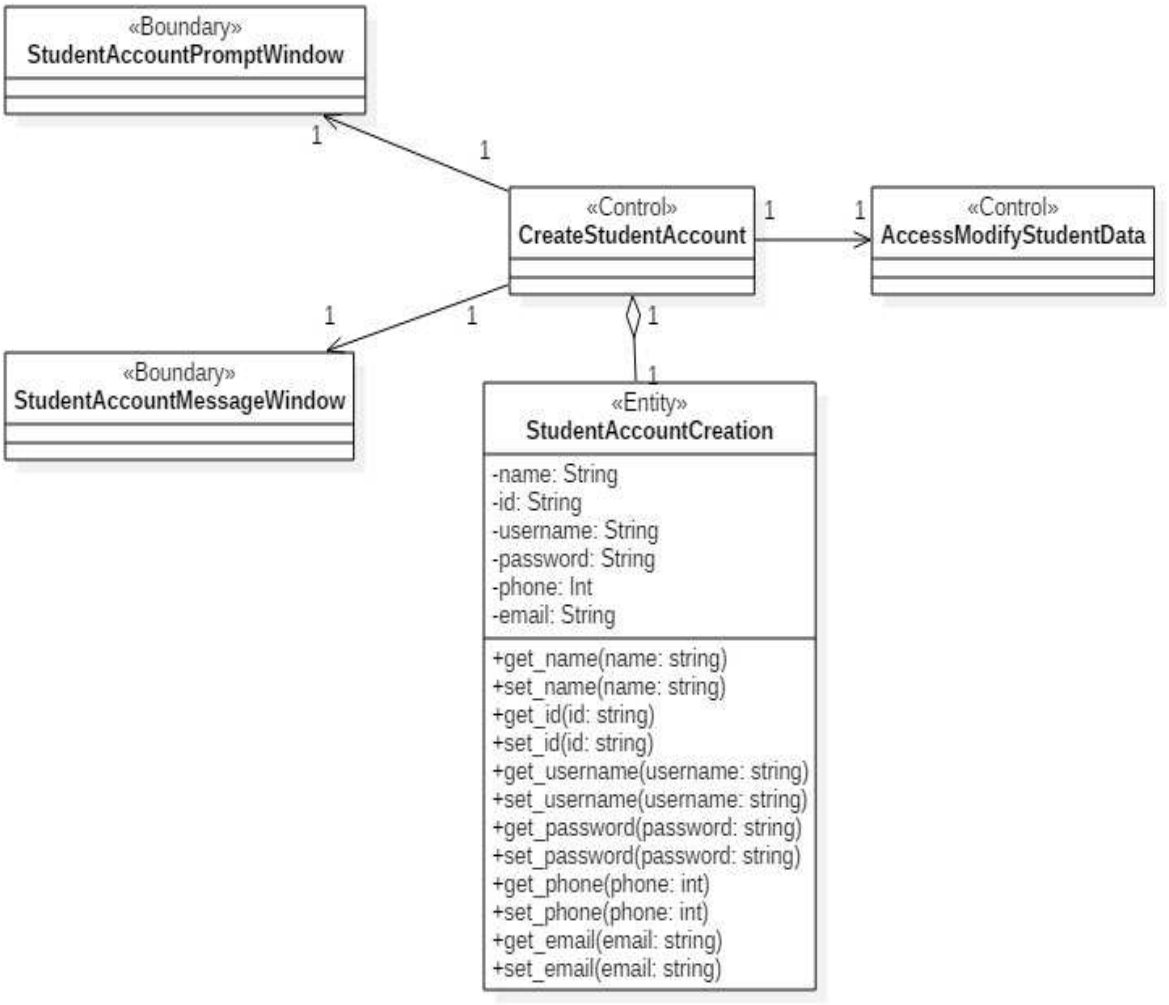
Flow of events

Main Flow:

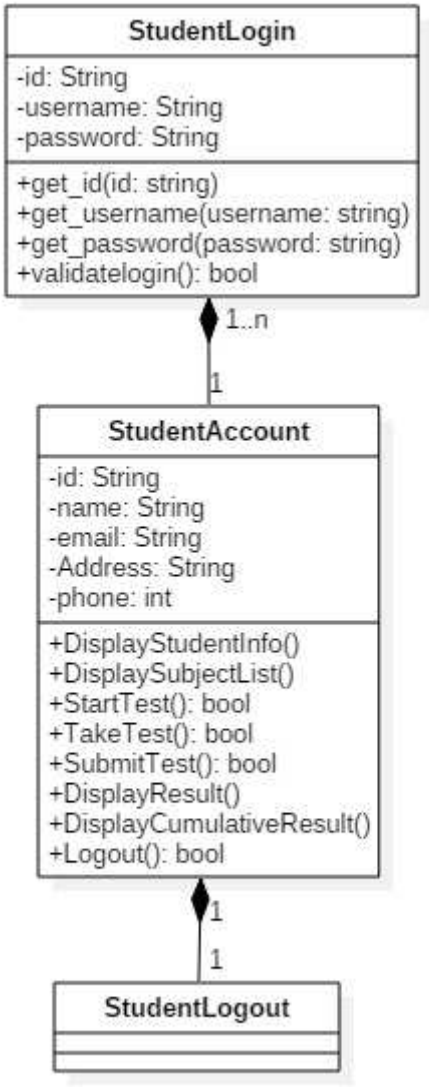
1. The use case begins to view cumulative result.
2. The student select the option to view the cumulative result.
3. After selecting the system provides the cumulative result.
4. The use case ends.

Entry condition:	the student must login.
Exit condition:	none.
Quality Requirements:	Any system operations that exceed 30 seconds in duration shall produce a timeout error reported to the student to try the operation again.

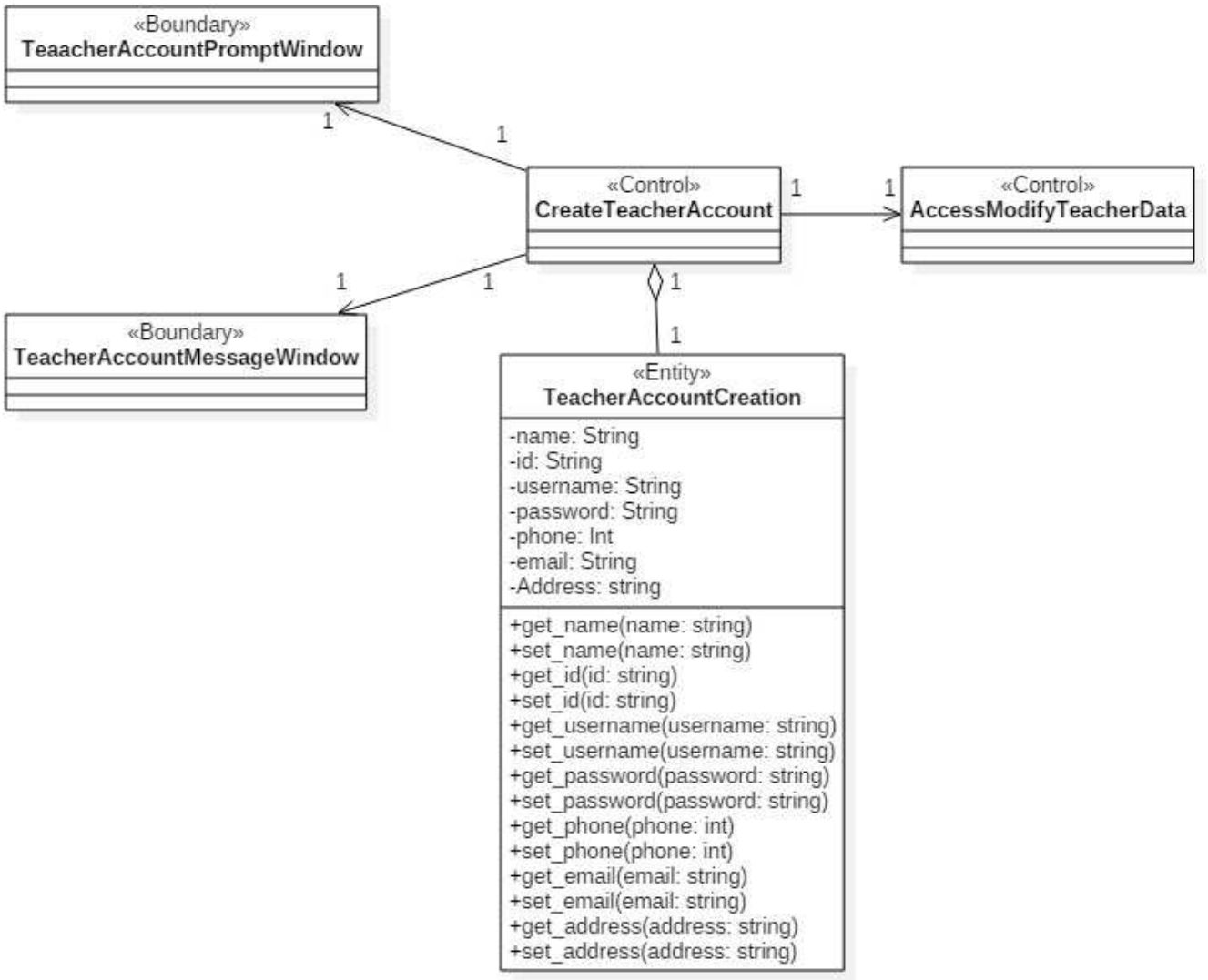
3.4.3 Object Model



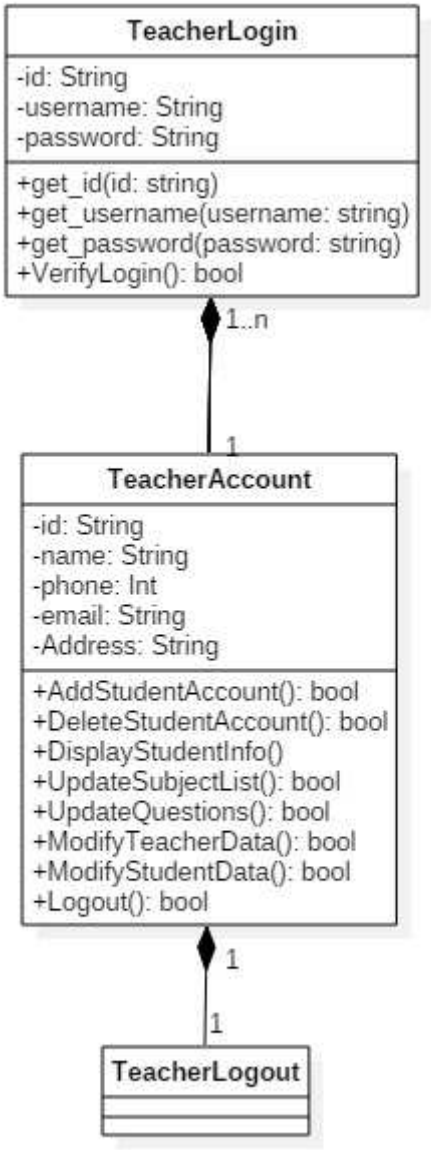
Student account creation class diagram



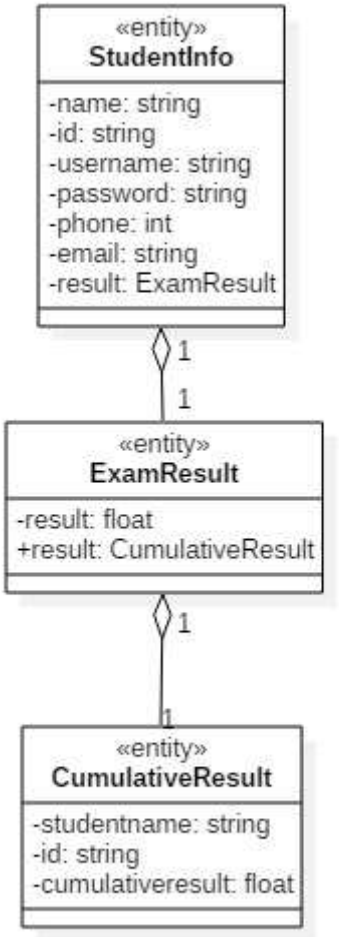
Student Login/Logout class diagram



Teacher Account Creation class diagram

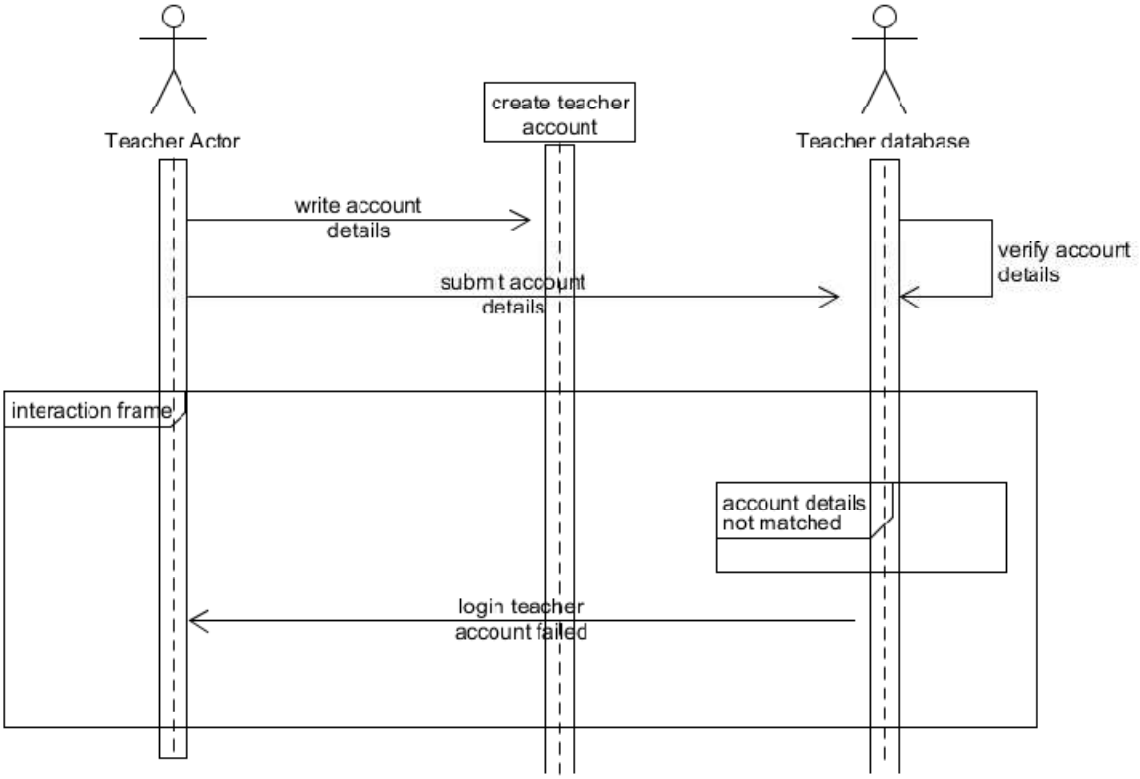


Teacher Login/Logout class diagram

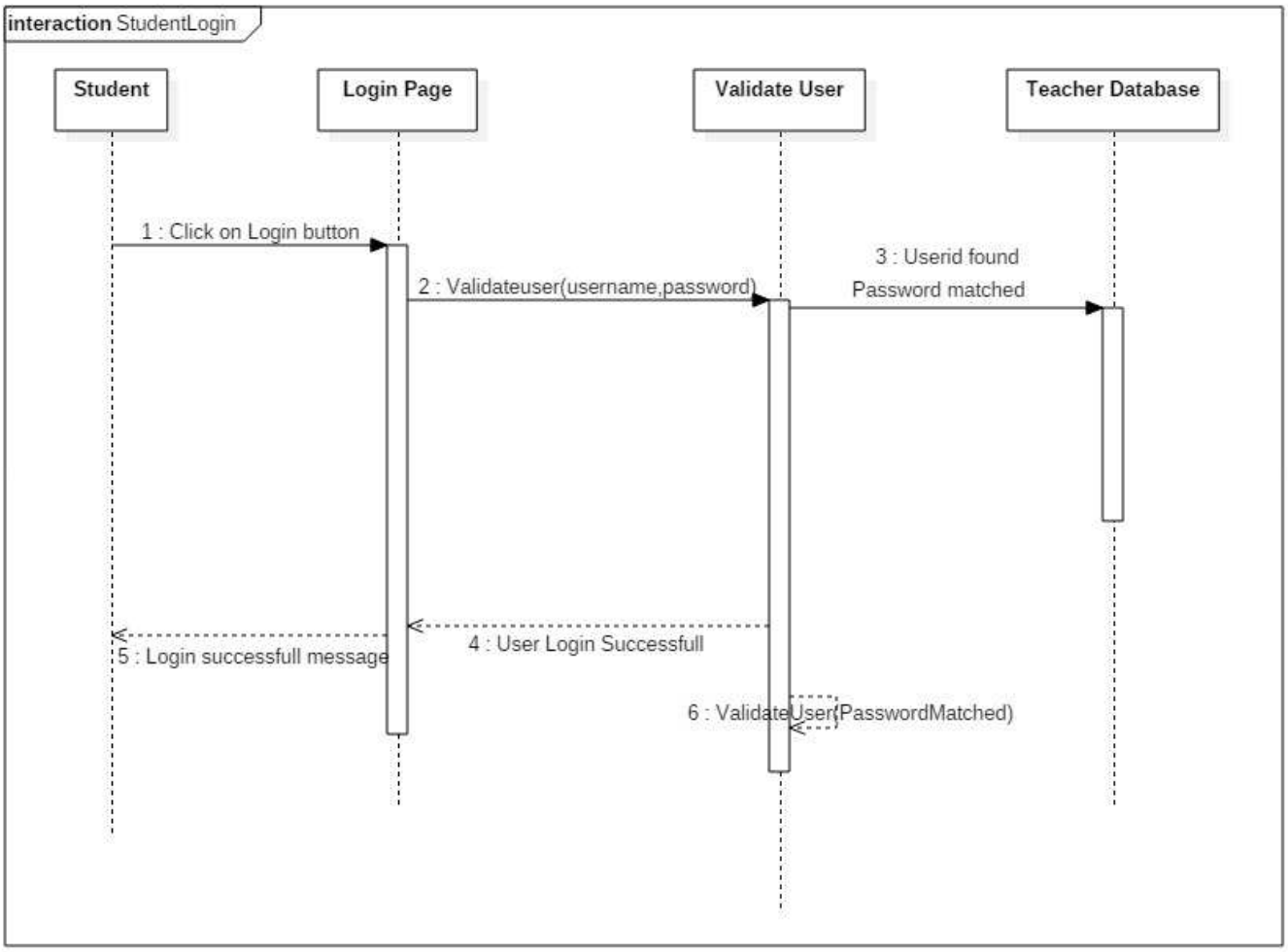


Student Info class diagram

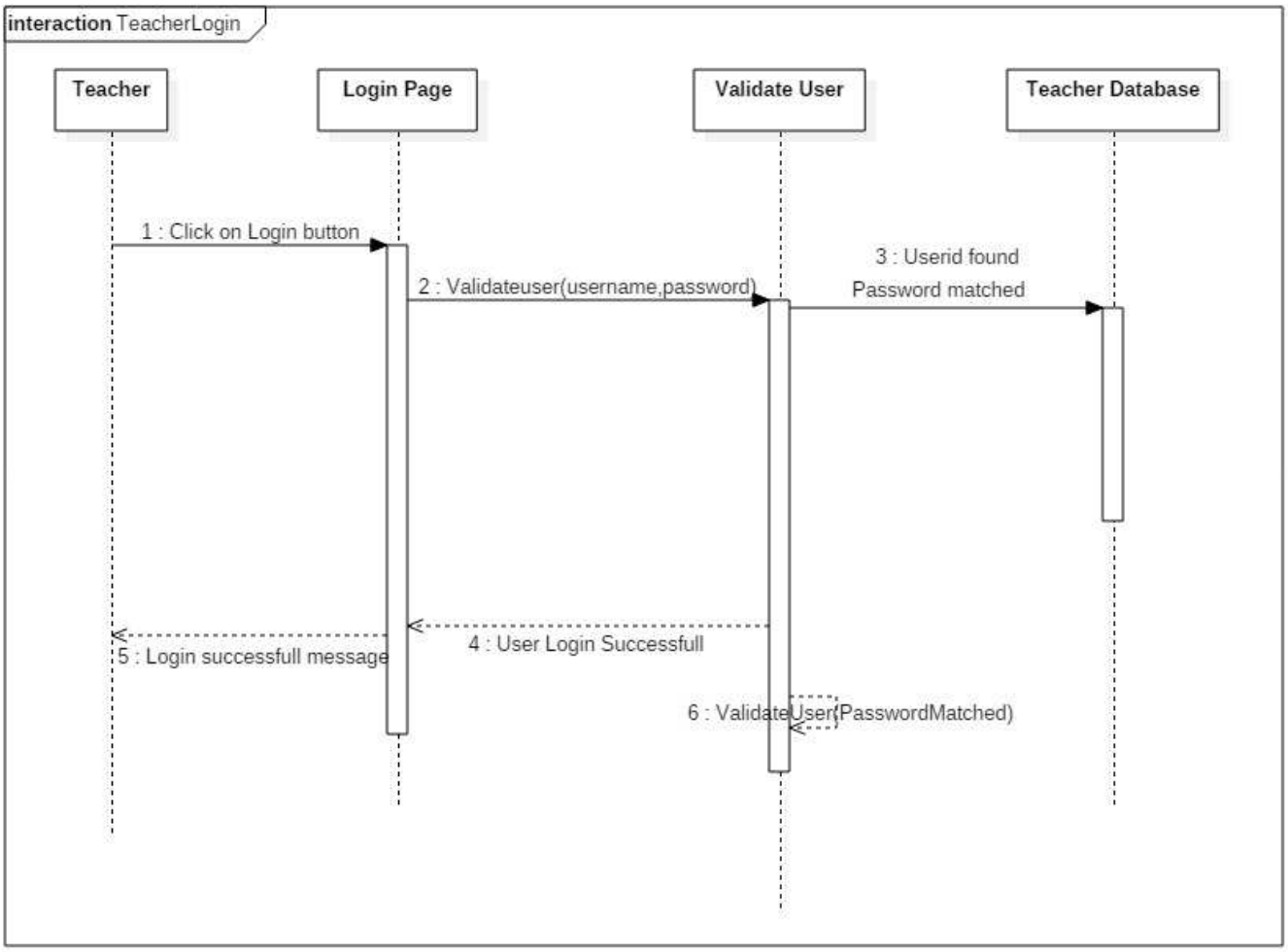
3.4.4 Analysis Dynamic Model



Teacher Account Creation sequence diagram



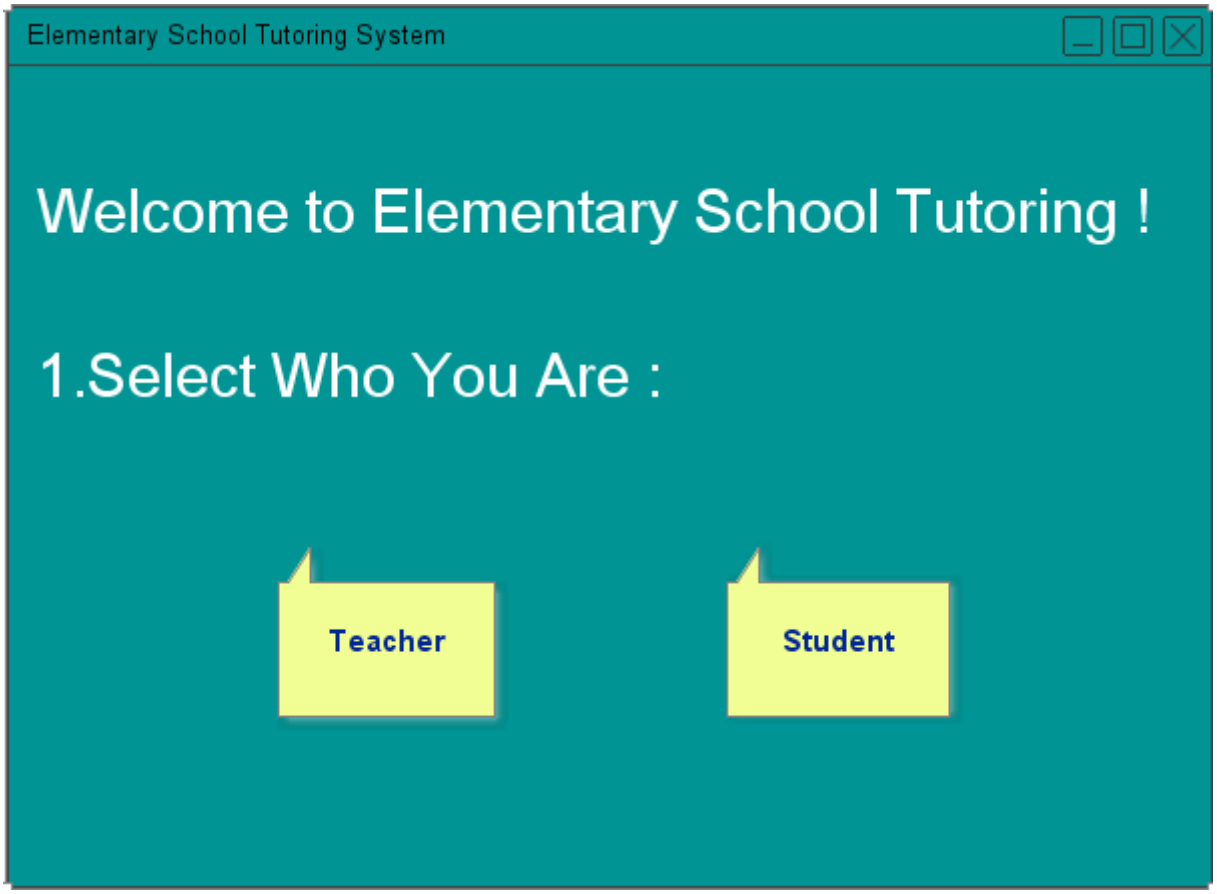
Student Login sequence diagram



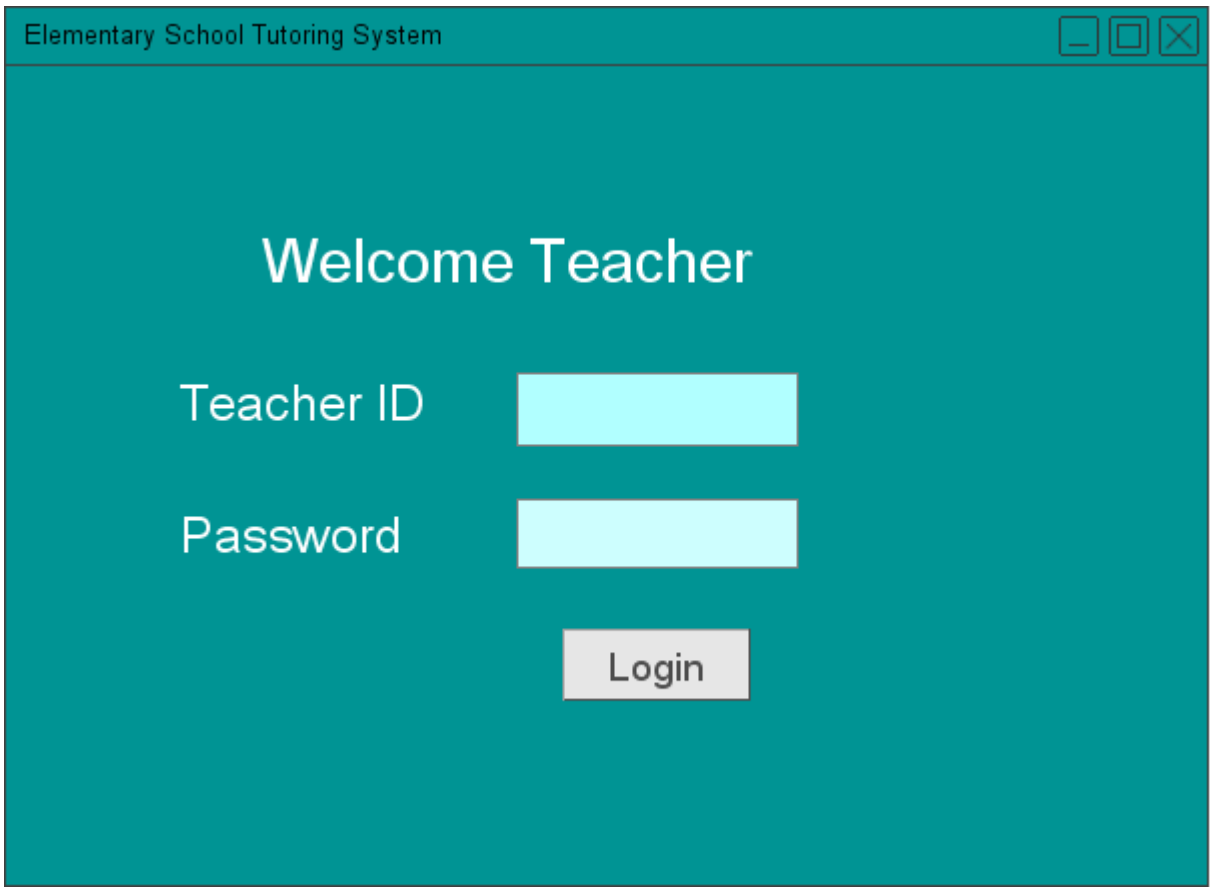
Teacher Login sequence diagram

3.4.5 User Interface

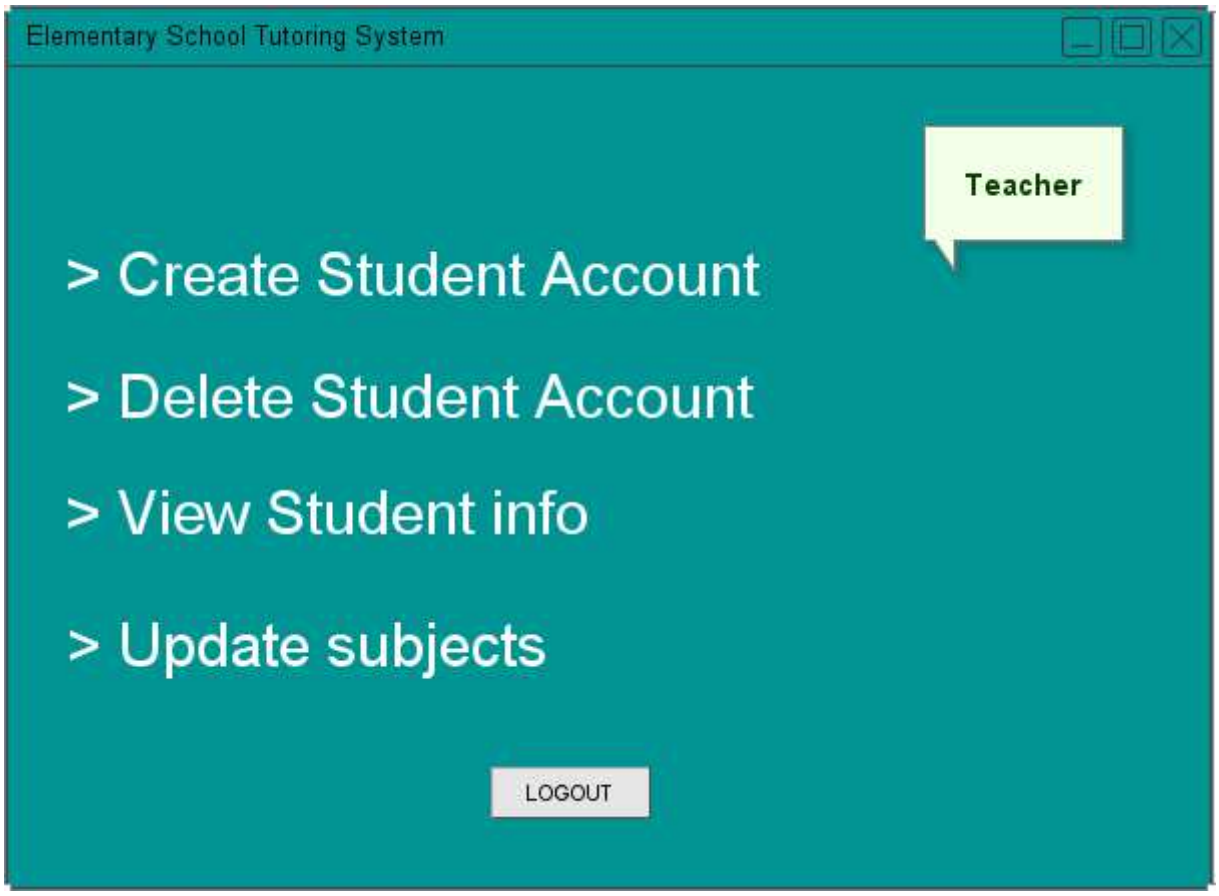
Elementary School Tutoring System>



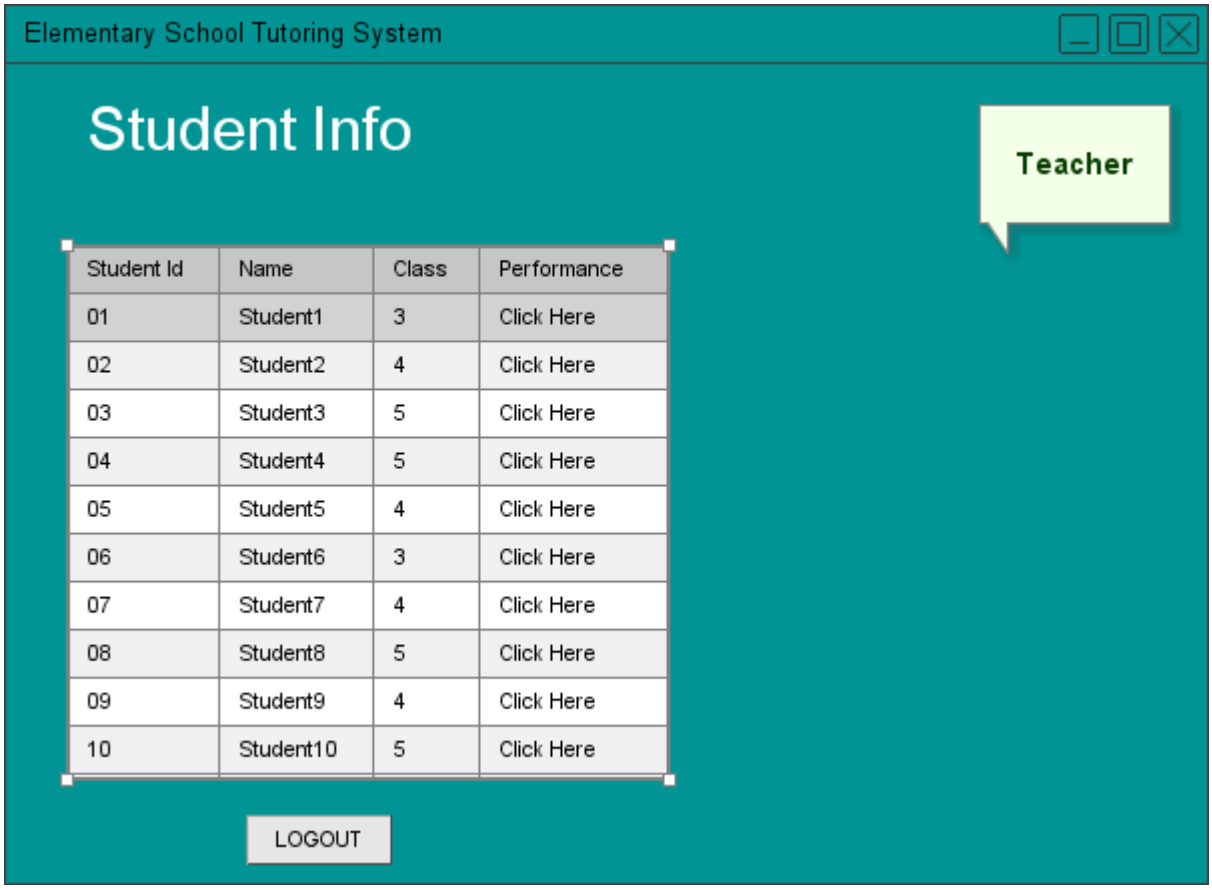
Elementary School Tutoring System> Teacher Login Page>



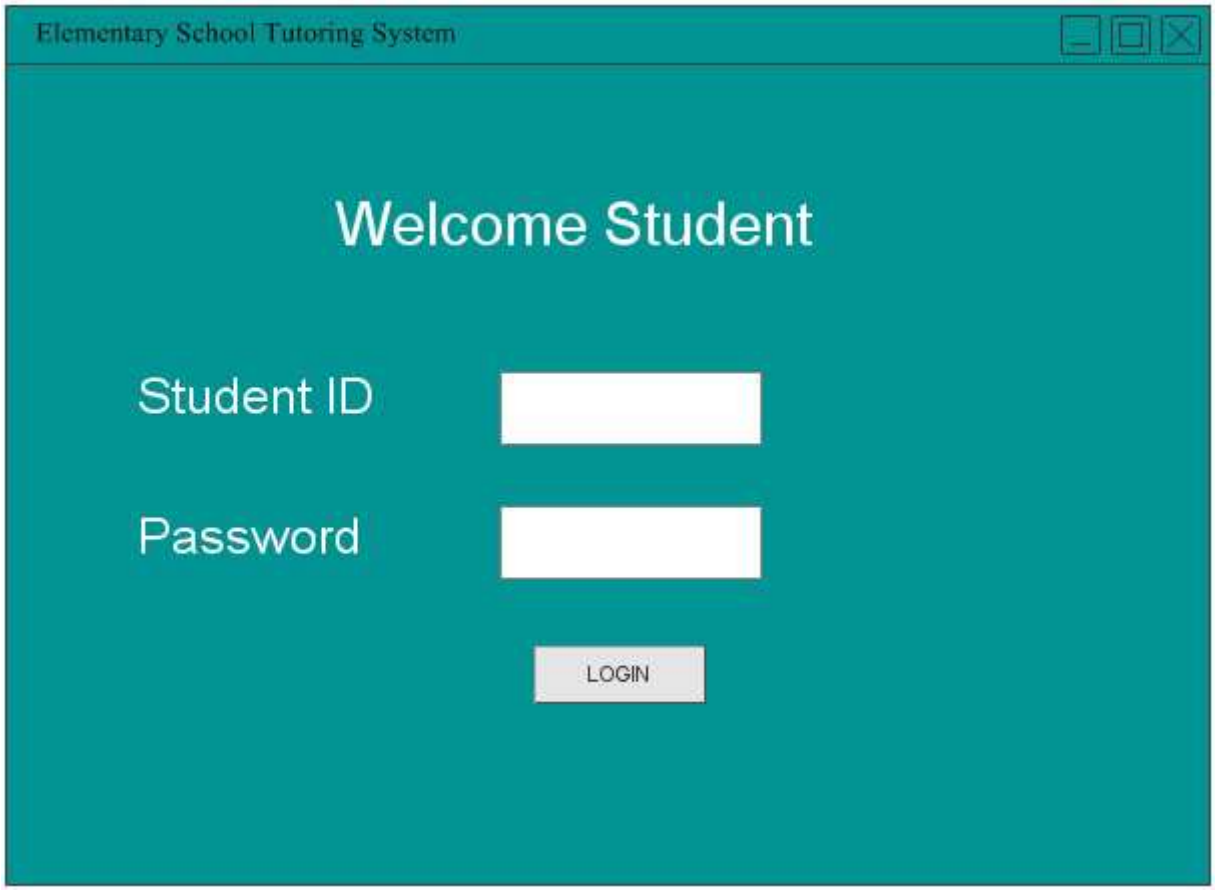
Elementary School Tutoring System> Teacher Login Page> Options to be selected by the teacher>



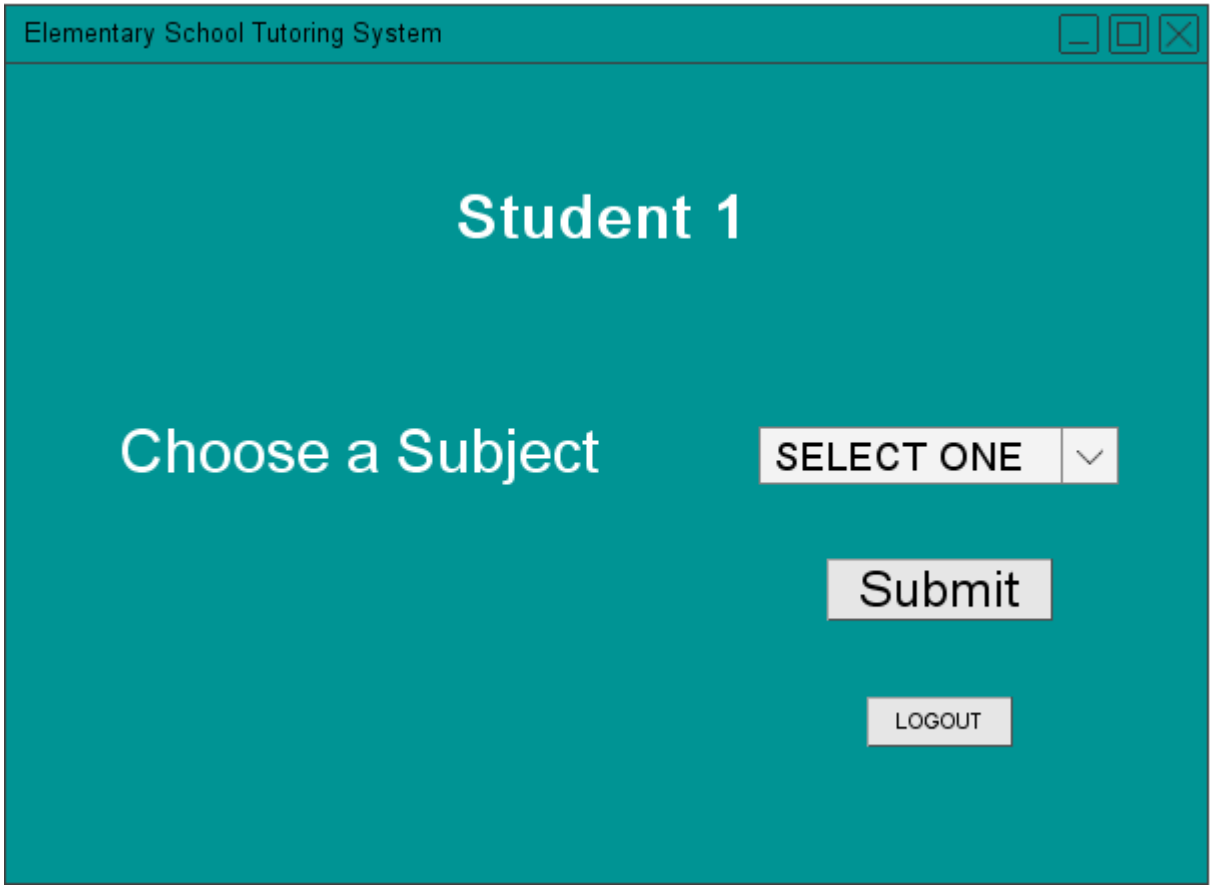
Elementary School Tutoring System> Teacher Login Page> Options to be selected by the teacher> Viewing Student Information page>



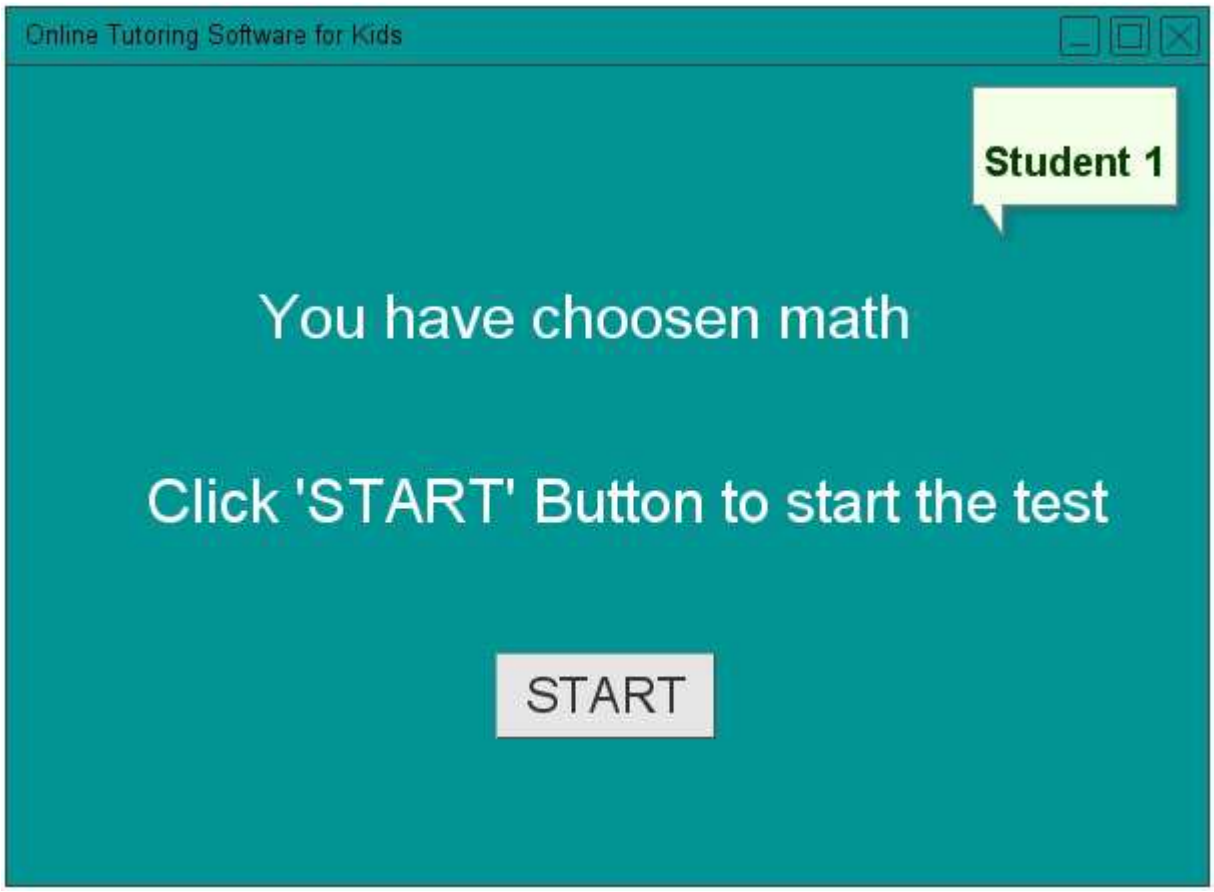
Elementary School Tutoring System> Student Login Page>



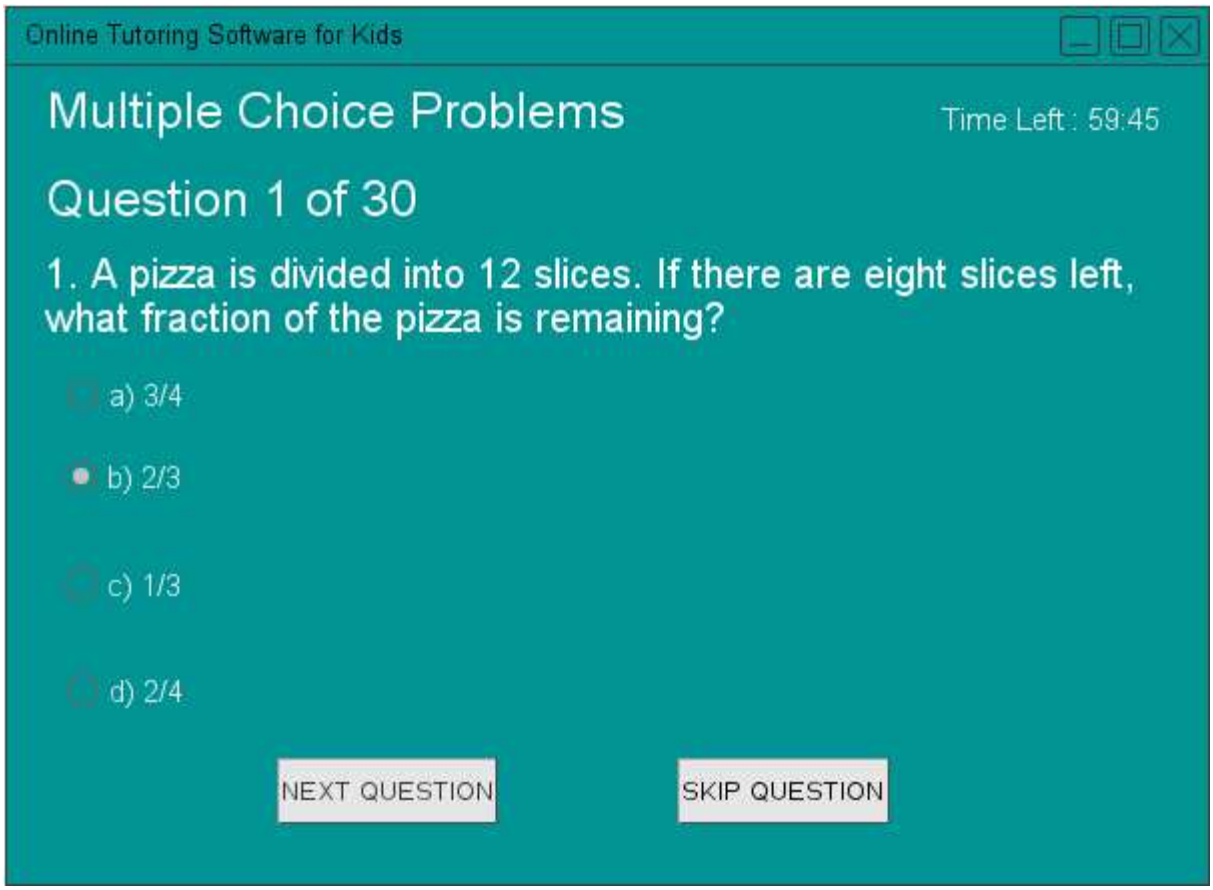
Elementary School Tutoring System> Student Login Page> Student 1 subject selection page>



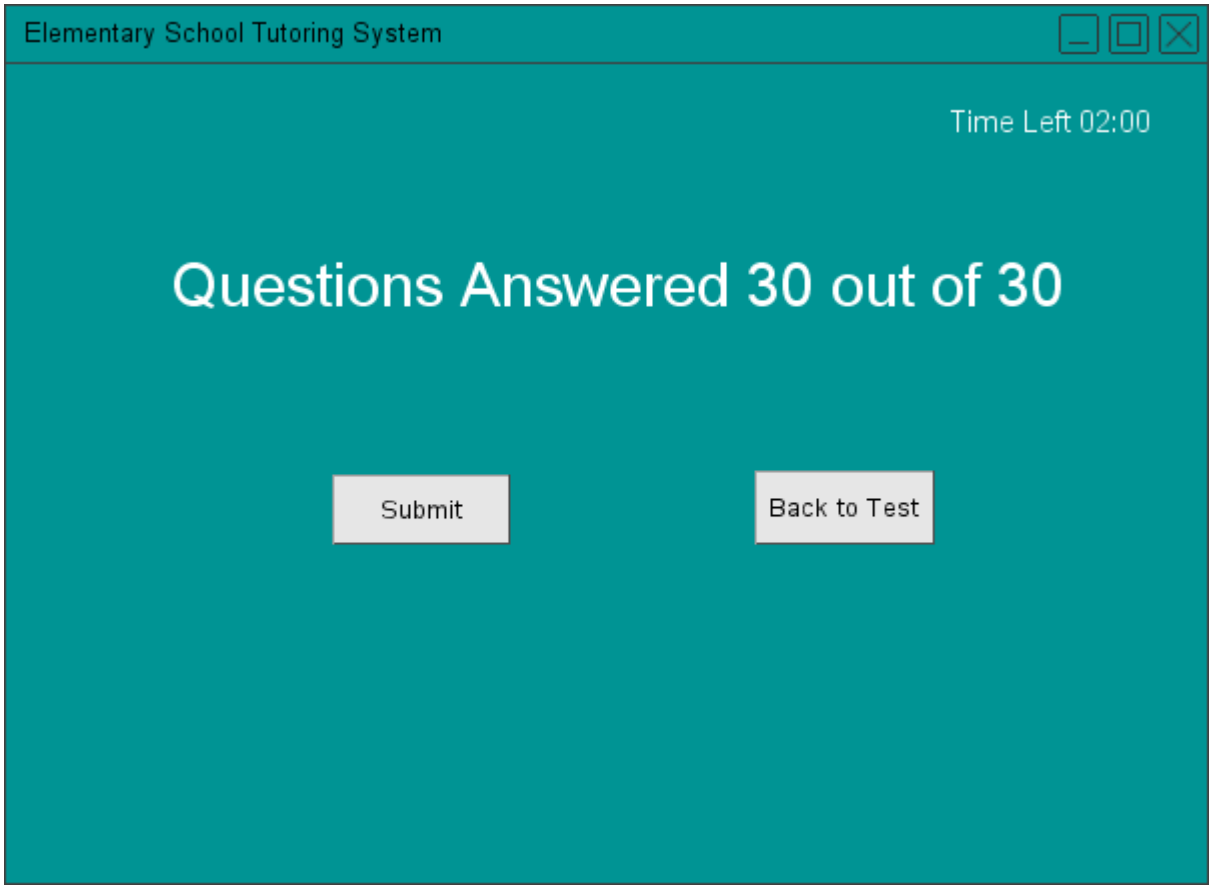
Elementary School Tutoring System> Student Login Page> Student 1 subject selection page> Student 1 chooses Subject Math and start test>



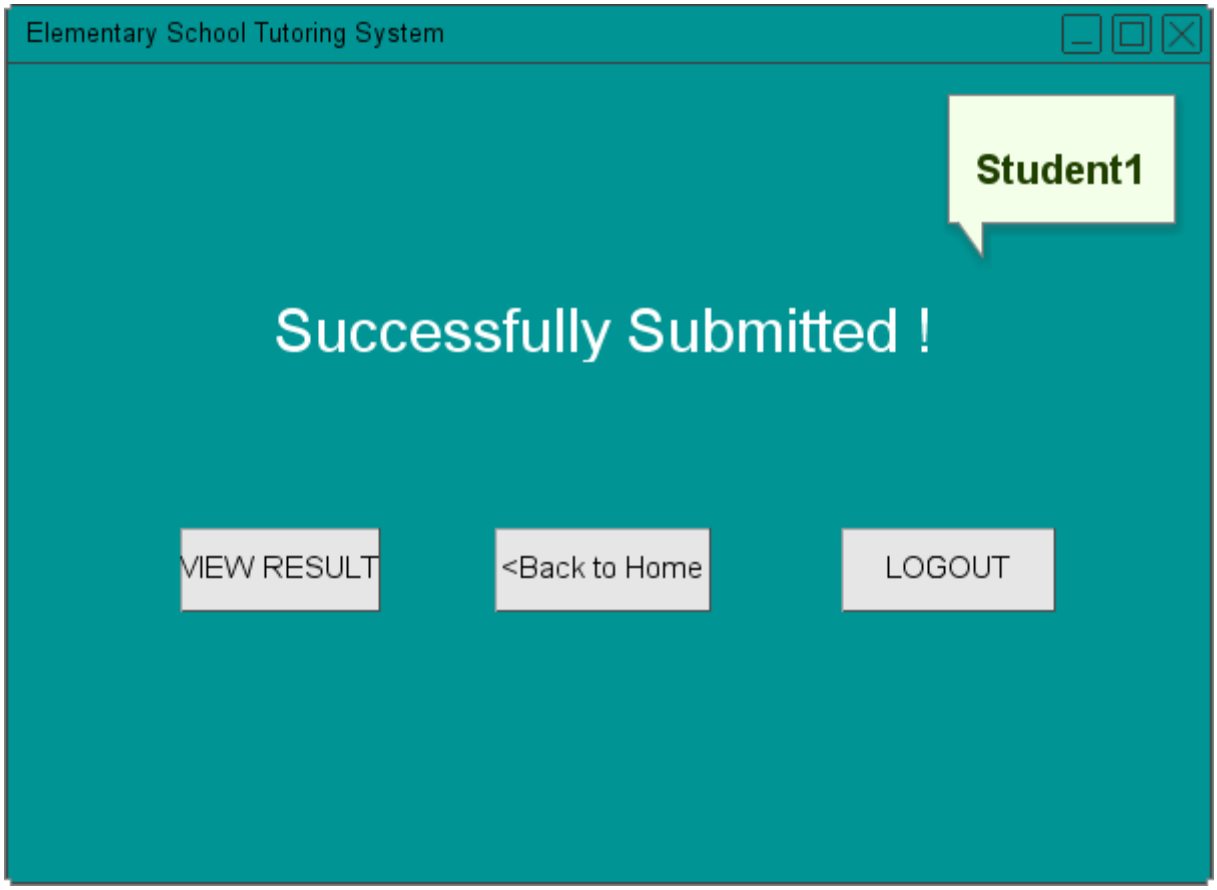
Elementary School Tutoring System> Student Login Page> Student 1 subject selection page> Student 1 chooses Subject Math and start test page> Student 1 testing page>



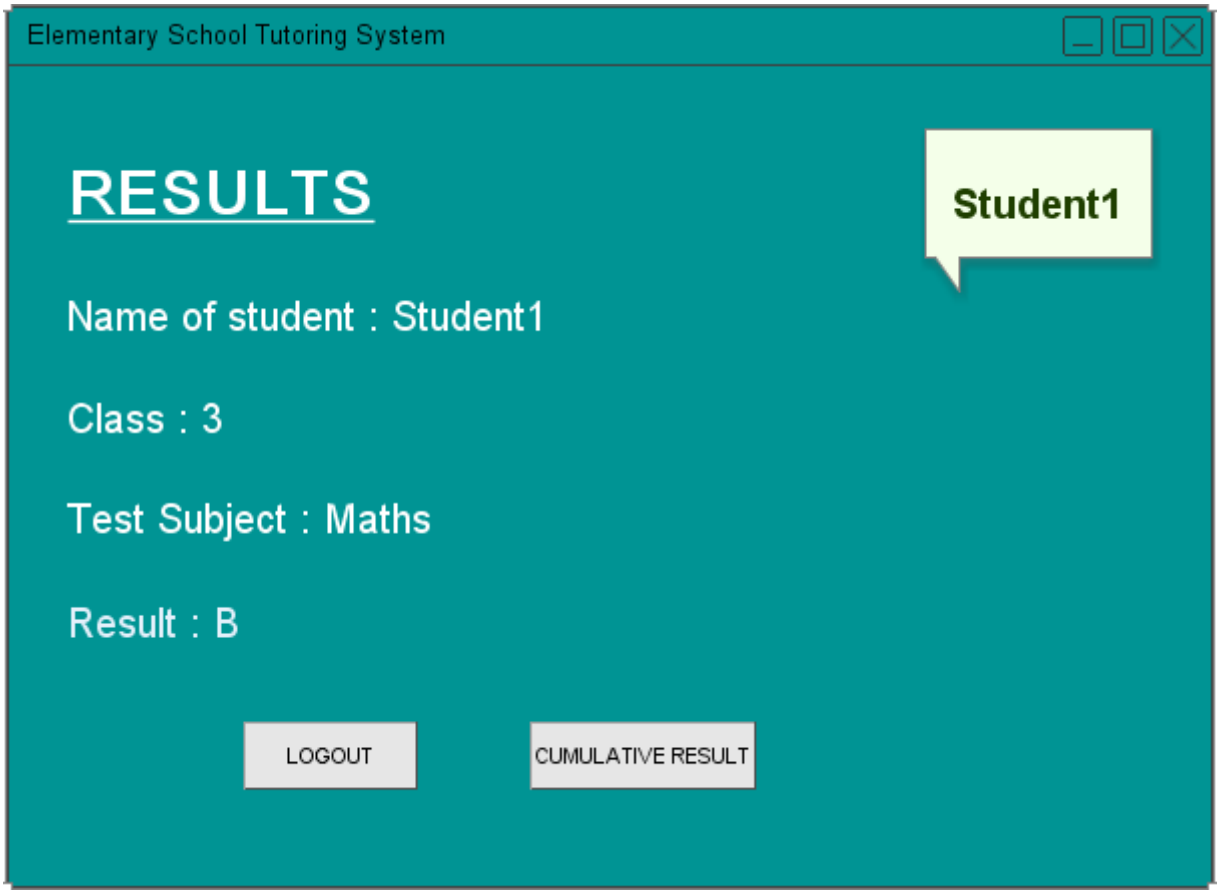
Elementary School Tutoring System> Student Login Page> Student 1 subject selection page> Student 1 chooses Subject Math and start test page> Student 1 testing page> Submit test page>



Elementary School Tutoring System> Student Login Page> Student 1 subject selection page> Student 1 chooses Subject Math and start test page> Student 1 testing page> Submit test page> Successful Submission page>



Elementary School Tutoring System> Student Login Page> Student 1 subject selection page> Student 1 chooses Subject Math and start test page> Student 1 testing page> Submit test page> Successful Submission page> Result Page>



4. Glossary/Definitions

Scenarios: To accomplish particular task scenarios helps the user how to interact with the system. It usually represents an illustrative example of user system interactions. It represents the sequence of interactions between the system and one or more actors

Use Case: It represents the sequence of interactions between one or more actors and the system.

Use Case Diagram: Representing the functionality of the system is analyzed by use case diagram. It also describes the interactions between actor and the system. Entry conditions and exit conditions are also included in use case diagram.

Entry Condition: Before the use case is used it should satisfy this condition.

Exit Condition: Before the use case is used it should satisfy this condition.

Boundary Condition: It is a special condition used by the system which includes start up, shut down and exceptions.

Non Functional Requirements: It describes the user visible aspects of the system that are not directly related with the functionality of the system.

Functional Requirements: Describes the interactions between a user and system independent of the system design or implementation.

Actor: It either acts upon the system or acted upon by the system. It can be a user, another system, physical environment etc. It must be outside of the boundary system.

Client: A role representing the person or company paying for the development of system.

Design Goal: A quality that the system should optimize. Design goals are often inferred from nonfunctional requirements and are used to guide design decisions.

End user: A role of the people who will use the delivered system.

Entity Object: An object that represents persistent o long-lived information tracked by the system.

Extend Relationship: A type of relationship in a use case diagram denoting that one use case extends the flow of events of another one.

Functional Model: Describes the functionality of the system from the users point of view.

Include Relationship: A type of relationship in a use case diagram denoting that a use case invokes another use case.

Rationale: The justification of decisions.

Customer: Current or potential buyer or user of the products or service of an individual or organization, called the supplier, seller, or vendor.

Defect: A flaw in a component or system that can cause the component or system to fail to perform its required function.

Failure: Deviation of the component or system from its expected delivery, service or result

Model: A system of assumptions, concepts and relationships between them allowing describing in an approximate way a specific aspect of reality.

Modeling language: Any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules.

Modeling tool: A tool that supports the creation, amendment and verification of models of the software or system.

Performance: The degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate.

Process: A set of interrelated activities, which transform inputs into outputs.

Process requirement: A requirement related to the development process.

Product: An output of a process.

Product requirement: A requirement related to the product of the development process. They affect quality of the product.

Quality: The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations

Risk analysis: The process of assessing identified risks to estimate their impact and probability of occurrence.

Security: Attributes of software products that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.

User: A person who uses a software product.