



System Design for the Solar Mobile Learning Application

Fahimul Hoque

P106501

Application Description:

(Note: The name of the application has not yet been confirmed thus in this report the application will be referred to as XYZ)

XYZ is a mobile application that enables its users to perform educational tasks like viewing educational contents (Slides, videos), engage in discussions, watch videos and take part in online quiz. The content of the application will be based on Solar Energy Concepts and Technologies mostly focusing in Solar Photovoltaic cells (PV).

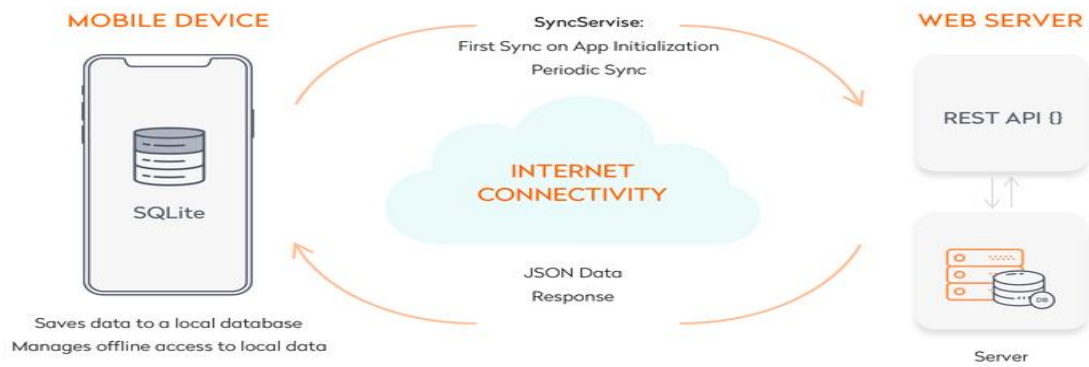
The primary users of the application are divided into two categories (actors): Students and Teachers.

- Students: Engage in content viewing tasks, watching videos, and performing quiz online.
- Teachers: Upload contents, videos, tasks and edit Quiz questions using the application/website

Before the users get access to the functionalities, both the users are required to register to the system by inserting the required credentials (username, email, password, etc.) and once registered the functionalities of the application are accessible to the user.

The application will also be accessible via website which will be developed and integrated with the application using APIs. API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. When we use an application on our mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to our phone. The application then interprets that data and presents us with the information we wanted in a readable way. All of this happens via API.

An API will be developed that allows the mobile application to communicate with a developed website (created by us) and show the students the required information, slides, quizzes they need. This will also allow the students who did not install the application to get all the functionalities by visiting the website directly using the mobile browser or laptop.



In this image shows how a mobile app communicates with the website's server using an API, where the API collects the required data from the server and sends it back to the mobile app for users to see/save

Network Architecture Diagram

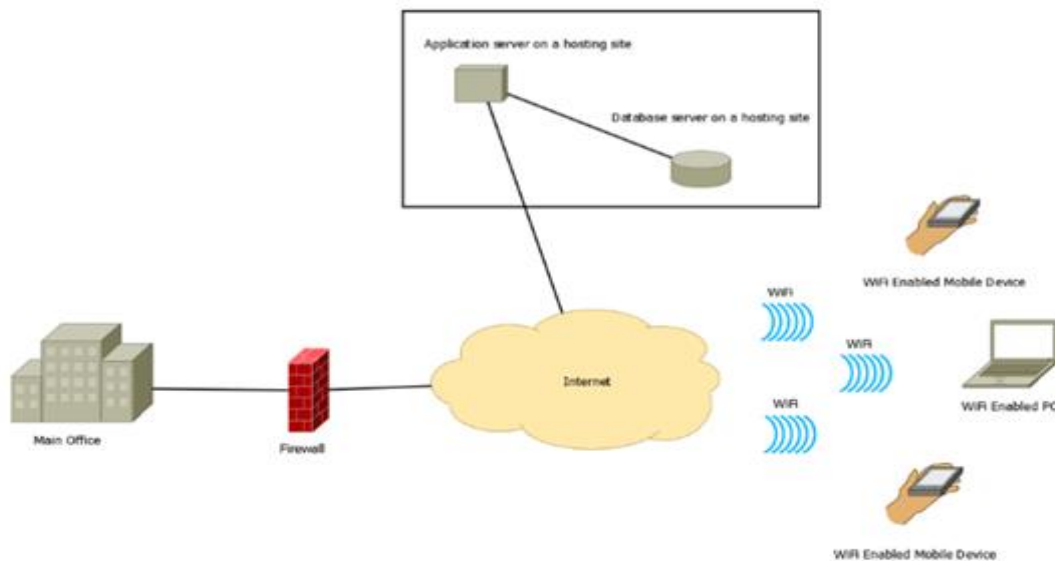


Figure 1: Network Architecture Diagram

The network architecture diagram showcases the components responsible for running the application. The main office will hold the personnel responsible for developing, maintaining and analysing the data from the application. The firewall protects the main office's network

infrastructure from malicious attacks (hacking), the data for the application will be stored in online hosting servers from which the users will be able to access the functionalities of the application from their smart phones and laptops and tablets.

Development Methodology

For the application we will be implementing Prototyping Methodology as the system development methodology (SDM). In this development method a prototype is built at a very early stage for the better understanding of the requirements of the system. Then the prototype is used, tested and further modified until the requirements of the users are matched. This development method is used when the user requirements are uncertain or unsure. So, a prototype is built, and developers work on it to keep improving the system. It's an iterative process in which user interaction and evaluation are included. When using this methodology, the amount of risk of the whole project is greatly reduced. The steps that will be taking in prototyping methodology are:



Figure 2: Steps of prototyping methodology.

1. Initial requirements: The first step of this development process are the initial gathering and analysis of requirements. The key objectives and functionalities will be listed out by the clients/users. The initial gathering of functional, technical and systemic requirements from various stakeholders is called requirement gathering.
2. Design: After all the requirements are collected from the stakeholders, a rough design of the system will be created. This early design will not contain the details of the system but only the important functionalities of it. As a draft, it will give the users an idea of how the system will look and function (Rouse M. , 2016). All the class diagram and use cases will be created here in this step.
3. Prototyping: After a rough design is made a prototype is then built based on that design. A prototype of the proposed system is built in this stage. This system will have some but not all the functionalities of the proposed system. The prototypes will be created after each iterations.
4. Customer Evaluation: When the prototype is built, is it then presented to users who'll use the system and give proper feedback of its success and flaws. All of the suggestions and feedbacks are then gathered and collected as customer evaluation and passed on to the developers.
5. Deployment: When the developers get all the suggestions and feedbacks from the users of the prototype they will start making changes to the system. the system will be modified and improved upon users request and a new prototype will be created at the end. The prototype will again be evaluated and tested. This step will continue until all the user requirements are fulfilled. When all the user requirements match the system only then the final system will be created and deployed to the masses (Rouse M. , 2016).

Functional Requirements:

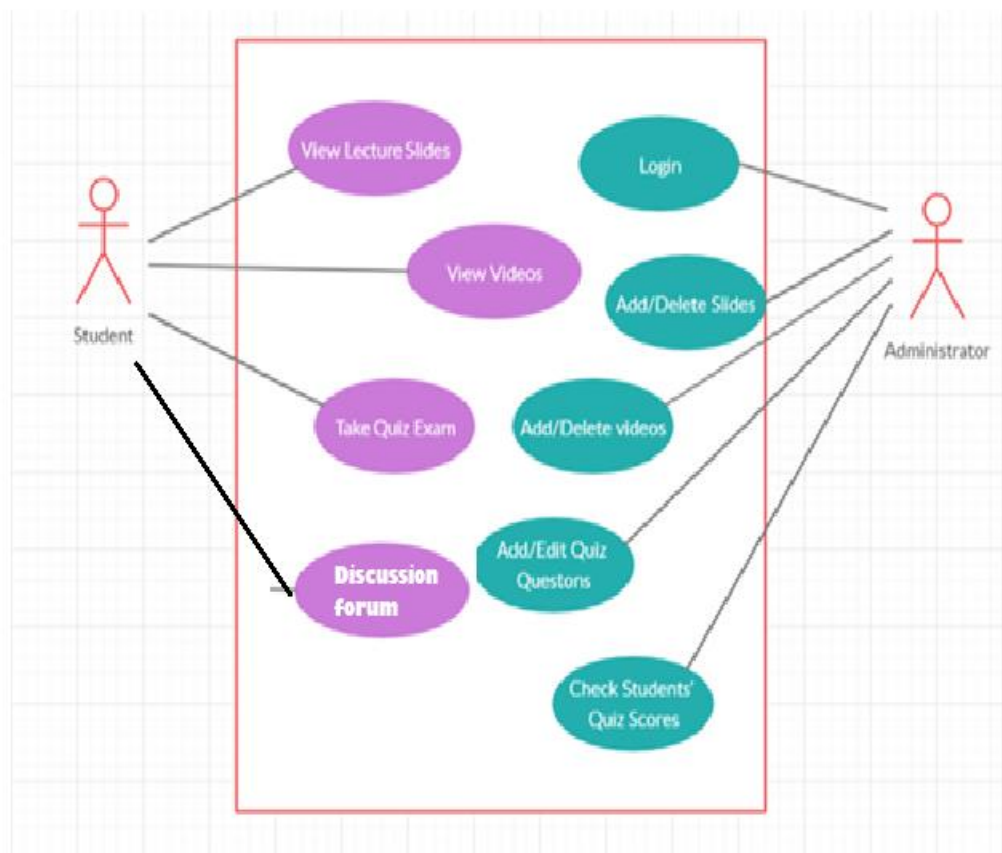


Figure 3: Use case diagram for the application

The use case diagram demonstrates the functionalities of the application that have been decided till now. The Student are able to perform three functions: View the content of the courses, view videos related to the topics and take part in quiz. The Teacher/Administrator is able to add or remove contents, add or remove videos, add and edit quiz questions, and check the overall student scores for all the students. The score of the quiz for the students will be shown to each student individually once they complete the quiz and the score will be illustrated instantaneously.

Analysis of Use Cases

Use Case:	Signup
Actor(s):	Student/Administrator
Description:	This use case begins when an individual student or teacher wants to register to the application.

Use Case	Signup	
Goal in Context	To register a new student or teacher into the system	
Primary Actor	Student/Teacher	
Trigger	An individual wants to register to the system	
Typical Course of Events	System Responses	
Actor Action		
1. This use case begins when student or teacher wants to register into the system.		
2. The individual will input the Username, password, email, first name, last name,Student ID.	System checks all require fields are given or not and then stores the data in the user database.	
Alternative Course of Events		
Line 2a: If the user is a teacher then he or she inputs username, password, email, first name, last name and teacher ID.		

Use Case:	Login
Actor(s):	Student/Administrator
Description:	This use case begins when an individual student or teacher wants to log in to the application.

Use Case	Login
Goal in Context	To allow a new student or teacher to log into the system

Primary Actor	Student/Teacher	
Trigger	An individual wants to log in to the system	
Typical Course of Events	System Responses	
Actor Action		
1. This use case begins when student or teacher wants to log into the system.		
2. The individual will input the Username, password,.	System checks all require fields are given and matches the data stored into the system. If accurate then the user is allowed to enter the application.	
Alternative Course of Events		
Line 2a: If the data doesn't match then an error message is displayed.		

Use Case:	Add/Delete Slides
Actor(s):	Administrator
Description:	This use case begins when an administrator wants to make changes to the content.

Use Case	Add/Delete Slides	
Goal in Context	To allow teachers to add or remove contents	
Primary Actor	Teacher	
Trigger	A teacher wants to add or remove content in the system	
Typical Course of Events	System Responses	
Actor Action		
1. This use case begins when a teacher wants to add or remove contents.		
2. The teacher add/removes new content by selecting the content from their device and uploading them.	System checks if the content is uploaded and then saves it to the database for other users to access.	

Alternative Course of Events
Line 2a: If an error occurs while uploading then an error message is displayed highlighting the error.

The other functionalities of the teacher : Add/Remove videos, Add/Remove quiz all follow the same steps as the use case mentioned above.

Use Case:	View Students Quiz score
Actor(s):	Administrator
Description:	This use case begins when an administrator wants to check the scores of all the students that have taken the online quiz

Use Case	Check Quiz Scores	
Goal in Context	To allow teachers to check the quiz scores of all the students	
Primary Actor	Teacher	
Trigger	A teacher wants to check the quiz scores	
Typical Course of Events	System Responses	
Actor Action		
1. This use case begins when a teacher wants to check the quiz scores of all the students.		
2. The teacher clicks on the check score option and input criteria (select class, date,topic) and the resultant scores are displayed.	System checks the selected criteria and then searches the database to find the scores related to the criteria. Once found the data and result scores are displayed.	
Alternative Course of Events		
Line 2a: If no data can be found based on the criteria then an error message is displayed.		

Use Case:	Discussion forum
Actor(s):	Student
Description:	This use case begins when an individual student wants to engage in discussion with peers regarding a topic.

Use Case	Discussion forum	
Goal in Context	To allow students to engage in discussion with peers	
Primary Actor	Students	
Trigger	A students wants to discuss a topic with peers	
Typical Course of Events	System Responses	
Actor Action		
1. This use case begins when a student wants to engage in discussion with peers regarding a topic		
2. The students click on the “discussion” forum option.	System checks the selected criteria and then searches the database to find all the topics posted by peers and displays them in order of date posted.	
3. Students Create a new topic with a title and posts a comment	The system saves the new topic to the database and allows peers to make comments on the post.	
Alternative Course of Events		
Line 3a: If students wants to make comment on a previously posted topic, the system retrieves all the previous comments from the database and allows student to make new comments and saves them to the database.		

For the students activities, the student don’t have access to make changes to the contents,videos and quiz. They are only permitted to view the contents and take part in the quiz by selecting the required options from the application’s menu.

Non-Functional Requirements

- **Technical Requirements**

The operational environment of the application will be on the Internet and can be accessed through the mobile application. The system is operational and accessible in any device with any operating system (Android or iOS) and Internet connection. The system is also designed to be accessible via tablet devices using Android or IOS or Windows OS.

The application system will be developed using Flutter/ReactNative a mobile application development frameworks system and will be written in Python, Java, Javascript, CSS, PHP, XML and other mobile application related languages. The development process will be done on devices using Windows 10 Operating system.

The database for the application system will be created using MySQL and PHPMyAdmin and stored in a hosting site to make the system go online.

- **Usability Requirements**

Usability is defined to be the measurement of a product's capacity to achieve the goals of the users (Rouse M. , 2016). For the application, the usability requirements were implemented by following the standard mobile application for education systems. Since the age group for users were estimated to be of 14 years and above, the aim of the interface design was to keep the design minimal and avoid clustering of information that may divert the user from the main goal of the system. Donald Norman's Design Principles of consistency, visibility and constraint for websites were also implemented in the interface of the Charité website to increase the usability, effectiveness and efficiency of the system. Consistency was implemented by keeping the main functionality available in every page to reduce user's work and also by keeping the color pattern same in every page. Visibility principle is implemented by making the main functionalities like signup, login and view contents stand out from the rest of the contents in the pages. Constraint principle was implemented by putting proper validations and restricting access to pages based on the user's level, for example students will not be able to access teachers' dashboard and functionalities. All of these were implemented to improve the usability of the system and increase the performance of the system.

- Reliability Requirements

The availability of the mobile application will be 24/7 since the domain and hosting site for the application will be online based on the network architecture diagram. While the application is available any time from any place in the world, a lot of data transaction will be occurring within the system specially user's sensitive data. Thus, to make sure the integrity of the data is maintained, the database of the system will be backed up several times within a month in case of data being lost or server crashing down. Also integrity of data is maintained by implementing proper validations while user's are inputting their personal data so avoid data redundancy and duplication. Validations like similar username with another previous user, password length, email will be checked.

- Security Requirements

The application's functionalities are divided between the two types of users (Student and Teacher). Thus, to restrict one type of user from accessing the other's functionalities username and passwords are implemented that will enable the system to identify the access each user has to the functionalities. For every user to access the functionalities of the system, username/email and passwords have to be entered and matched with the values stored in the system's database.

In addition, the data that will be stored in the database will be automatically encrypted using MD5 HASH, so that user's personal and sensitive data are protected and reduce leak of sensitive information.

Sequence Diagrams for the use cases:

The sequence diagram demonstrates how each components of the system and the user will interact with the application to perform each use case and tasks.

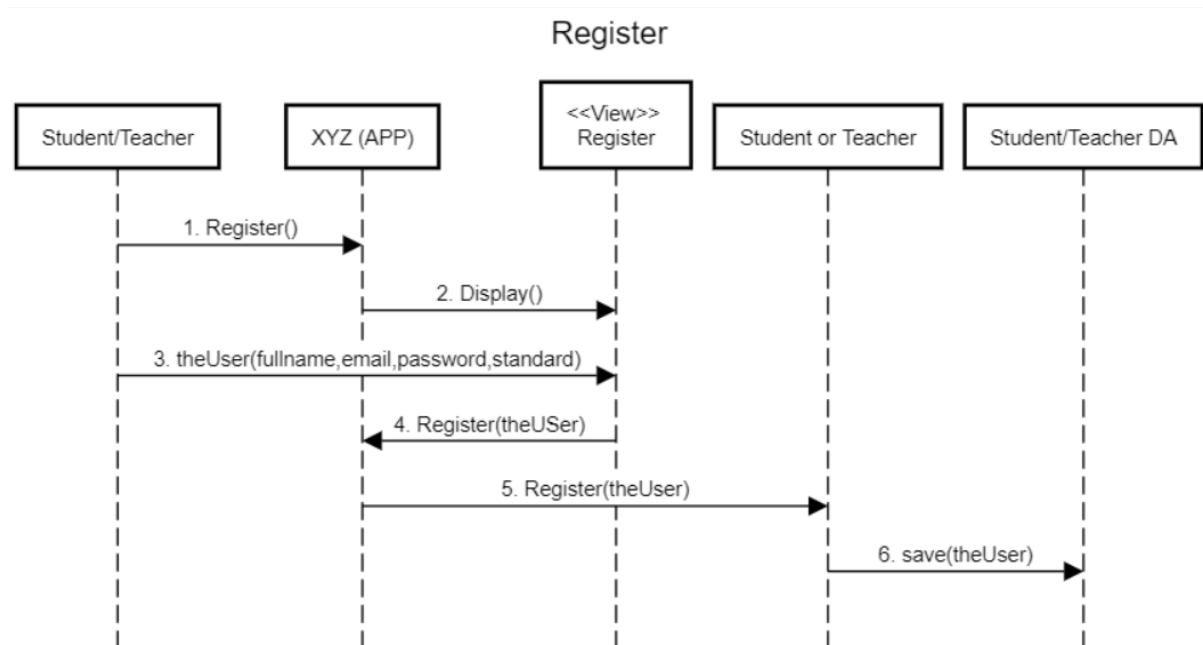


Figure 3: Sequence diagram for use case “Register”

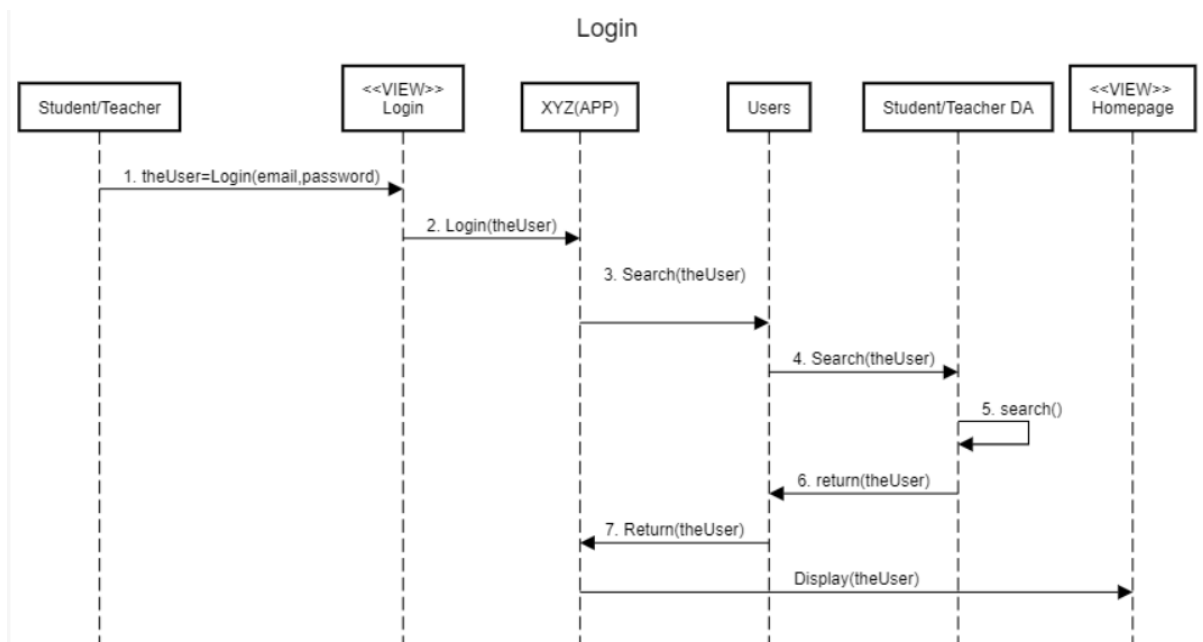


Figure 4: Sequence Diagram for use case “Login”

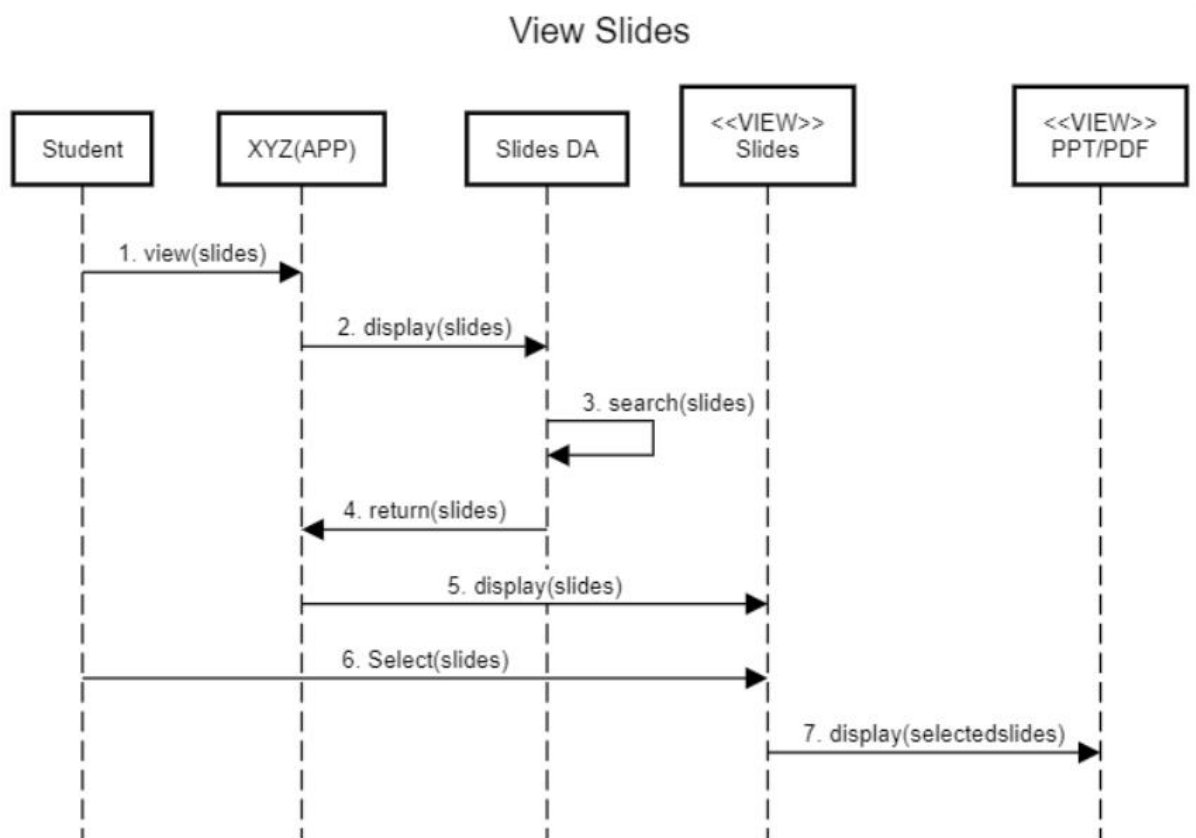


Figure 5: Sequence Diagram for use case “View Slides”

The use cases “View videos” “View quiz” all follow the same sequence as above.

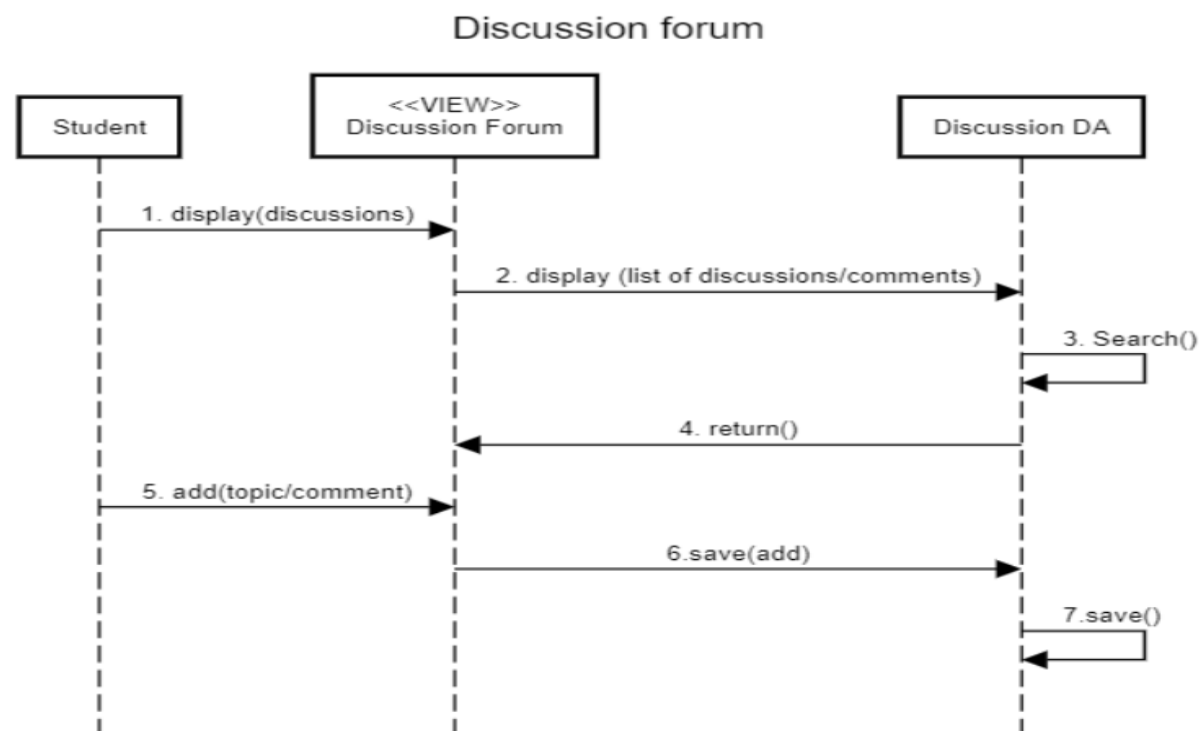


Figure 6: Sequence diagram for use case “Discussion forum”

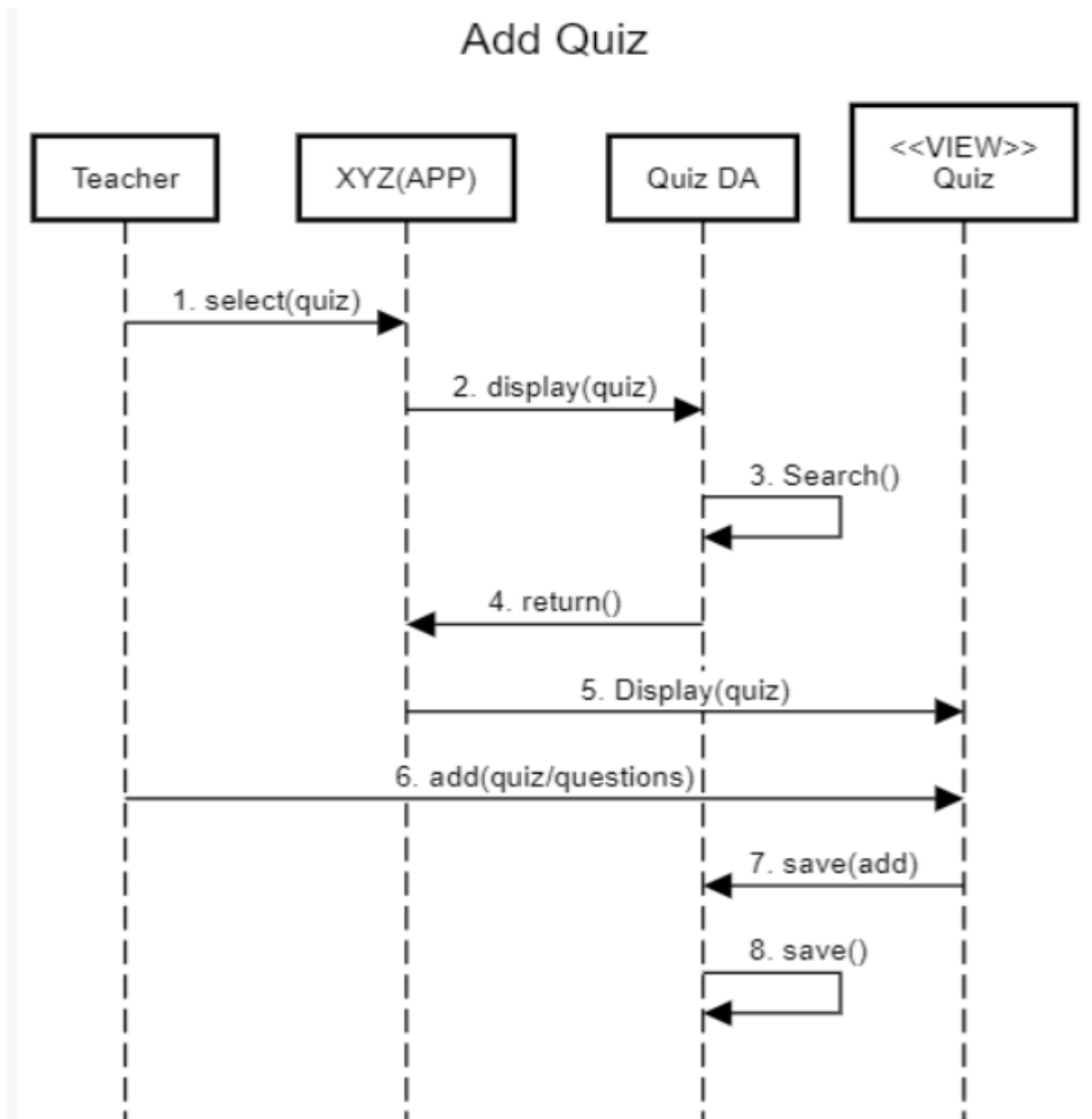


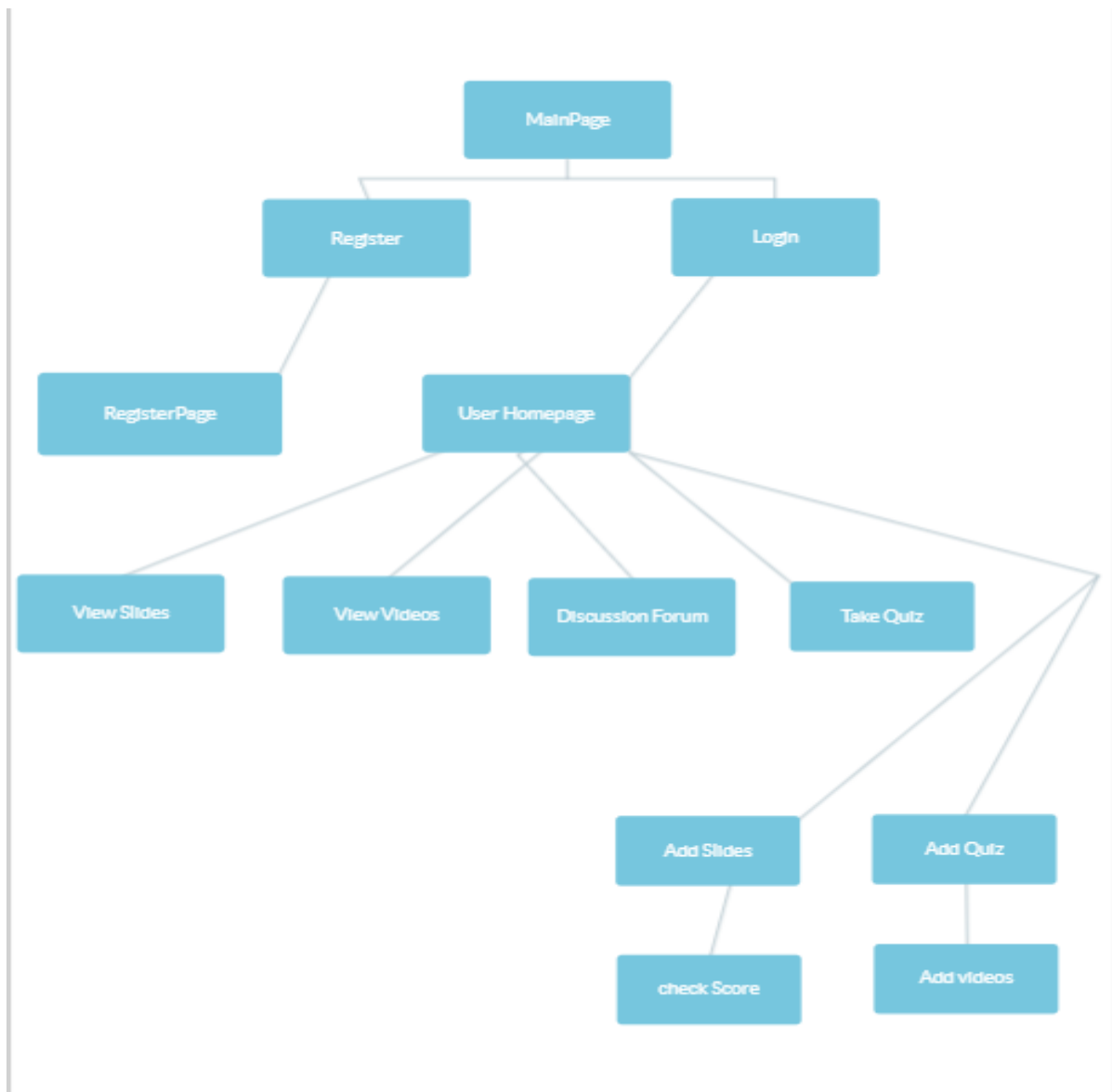
Figure 7: Sequence diagram for use case “Add Quiz”

The other use cases for the teacher “Add slides”/ “Add videos” all follow the above sequence.

Risk Management:

Risk	Potential Impact on Project Success L/M/H	Likelihood of Occurrence L/M/H	Mitigation Plan
Presence of bugs and errors in the system.	M	L	Perform various tests on each use cases to debug and remove the errors.
Hardware Failure	L	L	Have backup devices to continue the development process and keep backups of progress in various different external drives and cloud storage.
Unavailability of hosting sites	H	L	Secure more than one hosting site in a different region to use as a backup in the event of one of the hosting site becoming unavailable.
Loss of sensitive user data	H	L	Implement and test data encryption thoroughly while storing data in the database.
Project deliverables not meeting the objectives	M	M	Stay true to the original objectives and aims without any deviations by regularly revising the objectives with supervisor.

Web Hierarchy:



Iterations and Testing:

The project will be divided into three iterations:

- The first iteration will consist of “Register” “Login” and all the “Students” use cases
- The second iteration will consist of all the “Teachers” use cases
- The third iteration will consist of the development of the web site

Then the iterations will be joined together and an API will be attached with them to make the whole system (Application and Website) work together.

Each iterations will be tested separately for identifying bugs and errors. Each use cases within the iterations will also be tested separately to reduce bugs and errors to a minimum. The testing method will be conducted using:

- Unit Testing: where each use case will be tested by putting in all the various possibilities of scenarios and tested to make sure they run smoothly.
- Integration Testing: where each use will be tested based on how well they integrate with the other use cases
- System testing: where the overall system will be tested by performing several possible run tests and scenarios.