**Task 2: Implement Student Class Registration**

**Subtasks:**

1. **Create Student Input Interface for Class ID**
   * Develop a UI where students can input the class ID they received from their teacher.
   * Add fields for class selection or time if multiple options exist.
2. **Implement Class Search Based on ID and Time**
   * Write the backend logic to search Firebase for the corresponding class ID and validate the time.
   * Handle scenarios where students enter incorrect or expired IDs.
3. **Handle Registration for Classes with Multiple Times**
   * Decide whether students can choose a specific class time if available (e.g., Mr. Bob’s science at 12 PM or 2 PM).
   * Display available class times for students if multiple sessions exist.

To implement **Task 2: Student Class Registration**, you will need to work with both **Firebase** for the backend and **SwiftUI** for the user interface in Xcode. Here’s how you can approach the subtasks:

**1. Create Student Input Interface for Class ID (SwiftUI)**

* **Develop the UI:** Use SwiftUI to create a simple form where students can input the class ID. You can use a TextField for class ID input and possibly a Picker or List for class selection if there are multiple classes or time options.
* **SwiftUI Example**

struct ClassRegistrationView: View {

@State private var classID: String = ""

@State private var selectedClassTime: String = ""

let availableTimes = ["12 PM", "2 PM"] // Example times

var body: some View {

Form {

TextField("Enter Class ID", text: $classID)

Picker("Select Class Time", selection: $selectedClassTime) {

ForEach(availableTimes, id: \.self) {

Text($0)

}

}

}

.navigationTitle("Register for Class")

}

}

* This interface allows students to enter the class ID and select the time from the available options.

**2. Implement Class Search Based on ID and Time (Firebase Backend)**

* **Search Firebase for Class ID:** Use Firebase to store the class data. You can store class information in a Firestore collection (e.g., Classes). Each class can have an ID, a time, and other details.
* **Validate Class ID and Time:** When students submit their class ID and time, query Firebase to check if the class exists and if the time is valid. If the ID or time is incorrect, you can display an error message.
* **Firebase Example (in Swift):**

import Firebase

func validateClass(classID: String, classTime: String, completion: @escaping (Bool) -> Void) {

let db = Firestore.firestore()

db.collection("Classes").whereField("classID", isEqualTo: classID).whereField("time", isEqualTo: classTime).getDocuments { snapshot, error in

if let error = error {

print("Error getting documents: \(error)")

completion(false)

} else if snapshot?.isEmpty == true {

print("Class not found or time invalid")

completion(false)

} else {

print("Class found and valid time")

completion(true)

}

}

}

* This code queries Firestore to find a class by its ID and time.

**3. Handle Registration for Classes with Multiple Times**

* **Allow Students to Choose Class Time:** If there are multiple available times, present the user with the option to select from them. This can be done with a Picker or List in SwiftUI.
* **Display Available Times:** After retrieving class information from Firebase, display the available times if there are multiple options. The example UI in **Step 1** already shows how to create a picker for class times.
* If you want to dynamically load the available times from Firebase, you can fetch them first, then display the options in the UI.

**Summary:**

* **Firebase** will handle the backend, storing class data and checking if a class ID is valid.
* **SwiftUI** will create the interface for students to input class IDs and select times.

4o