# Republic of the Philippines

# SULTAN KUDARAT STATE UNIVERSITY

# GRADUATE SCHOOL

# MASTER IN INFORMATION TECHNOLOGY

# ACCESS, EJC Montilla, 9800 City of Tacurong

# PROPOSED CAPSTONE PROJECT IN MOBILE COMPUTING

# InteracTeach+: A Data-Driven Interactive Learning

# Management and Analytics System for Teachers and

# Students

# Submitted by:

# YUSEP M. OMAR

# Submitted to:

# Dr. ELBREN ANTONIO

## Chapter 1: Introduction

### Background of the Study

The digital transformation of education has accelerated the demand for intelligent learning systems. Teachers face challenges in manual lesson planning, tracking student progress, and creating personalized learning materials. Students often experience limited engagement and generic feedback. InteracTeach+ addresses these challenges by providing a mobile-ready platform (Flutter) integrated with analytics, AI-assisted content generation, and gamification.

### Problem Statement

General Problem: How can a smart learning platform improve student learning outcomes and reduce teacher workload using real-time data and intelligent automation?

**Specific Problems:**

1. How can lesson delivery and resource management be simplified?

2. How can students receive personalized feedback and learning paths?

3. What tools can help teachers identify learners needing intervention?

4. How can assessments be automatically generated based on lesson content?

5. How can user data privacy and ethical AI usage be ensured?

### Objectives

General Objective: Develop a smart interactive learning system that enhances teaching efficiency and student performance through automation and analytics.

Specific Objectives:

- Design a user-friendly mobile app for teacher content uploads.

- Develop a gamified learning interface for students.

- Implement an AI engine for assessment generation and remediation.

- Provide analytics dashboards and reporting tools.

- Establish protocols for ethical AI and data protection.

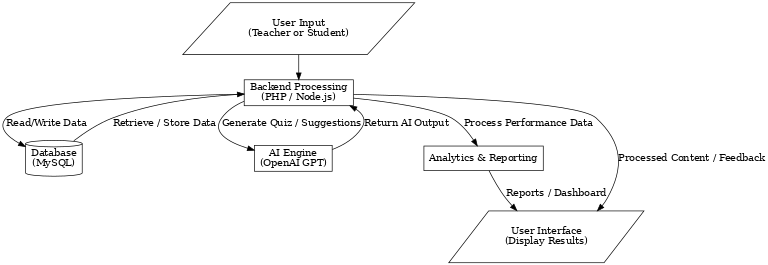
### Scope and Limitations

Scope: Includes content upload, quiz generation, analytics, AI-assisted suggestions, gamified student interaction, and basic access control.

Limitations: Web/API based; initial Flutter app focuses on core features. No biometric attendance or financial features. AI open-ended processing limited in initial phase. Accessibility features may be limited to English.

### Significance of the Project

**Teachers:** Automate tasks and gain AI insights.  
**Students:** Access engaging adaptive content.  
**Administrators:** Monitor school-wide performance.  
**Researchers:** Prototype for future LMS tools.  
**Data Privacy Stakeholders:** Benefit from ethical standards.



**Chapter 2: Review of Related Systems / Literature**

Existing systems like Google Classroom and Moodle focus on content delivery but lack advanced automation and analytics. InteracTeach+ integrates AI-generated assessments, gamification, and real-time analytics.

What makes InteracTeach+ simpler for beginners: Simplified content upload, guided quiz generation, and an intuitive mobile UI built with Flutter.

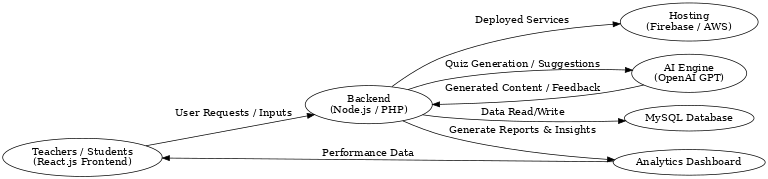
## Chapter 3: Methodology

### Tools and Technologies Used

**- Frontend:** Flutter (Dart) - Mobile app  
**- Backend:** PHP (REST API)  
**- Database:** MySQL  
**- Hosting:** Firebase / AWS for storage and hosting  
**- AI Engine:** OpenAI GPT (for content generation suggestions)  
**- Design:** Figma  
**- Testing:** Postman, Selenium / Flutter integration tests

### System Architecture

See diagram below:



System Flowchart:

### System Flow (simplified)

User Input (Flutter app) → REST API (PHP) → MySQL Database → AI Engine (for quiz generation when needed) → API returns response → Flutter displays result.

### Development Process

SDLC approach: Planning → Analysis → Design → Development → Testing → Deployment. Agile sprints recommended for iterative feature delivery.

**Chapter 4: System Design and Implementation**

### Screenshots of App UI

Insert screenshots of your Flutter app here when available.

### Code Snippets

Below are ready-to-use sample code files for a minimal mobile attendance and quiz submission system. Copy the code into your project files.

#### Flutter - main.dart (Sample)

import 'package:flutter/material.dart';

import 'package:http/http.dart' as http;

import 'dart:convert';

void main() => runApp(InteracTeachApp());

class InteracTeachApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'InteracTeach+',

home: LoginPage(),

);

}

}

class LoginPage extends StatefulWidget {

@override

\_LoginPageState createState() => \_LoginPageState();

}

class \_LoginPageState extends State<LoginPage> {

final TextEditingController emailController = TextEditingController();

final TextEditingController passwordController = TextEditingController();

Future<void> login() async {

final response = await http.post(

Uri.parse('https://your-server.com/api/login.php'),

body: {

'email': emailController.text,

'password': passwordController.text,

},

);

final data = json.decode(response.body);

if (data['success']) {

Navigator.push(context, MaterialPageRoute(builder: (\_) => HomePage(user: data['user'])));

} else {

ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text('Login failed')));

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('InteracTeach+ Login')),

body: Padding(

padding: EdgeInsets.all(16.0),

child: Column(

children: [

TextField(controller: emailController, decoration: InputDecoration(labelText: 'Email')),

TextField(controller: passwordController, decoration: InputDecoration(labelText: 'Password', obscureText: true)),

SizedBox(height: 20),

ElevatedButton(onPressed: login, child: Text('Login')),

],

),

),

);

}

}

class HomePage extends StatelessWidget {

final Map user;

HomePage({required this.user});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('InteracTeach+')),

body: Center(child: Text('Welcome, ${user['name']}')),

floatingActionButton: FloatingActionButton(

child: Icon(Icons.add),

onPressed: () => Navigator.push(context, MaterialPageRoute(builder: (\_) => SubmitAttendancePage(userId: user['id']))),

),

);

}

}

class SubmitAttendancePage extends StatefulWidget {

final int userId;

SubmitAttendancePage({required this.userId});

@override

\_SubmitAttendancePageState createState() => \_SubmitAttendancePageState();

}

class \_SubmitAttendancePageState extends State<SubmitAttendancePage> {

Future<void> submitAttendance(String status) async {

final response = await http.post(Uri.parse('https://your-server.com/api/attendance.php'),

body: {'user\_id': widget.userId.toString(), 'status': status});

final data = json.decode(response.body);

ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(data['message'])));

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Submit Attendance')),

body: Padding(

padding: EdgeInsets.all(16.0),

child: Column(

children: [

ElevatedButton(child: Text('Present'), onPressed: () => submitAttendance('present')),

ElevatedButton(child: Text('Absent'), onPressed: () => submitAttendance('absent')),

],

),

),

);

}

}

#### Backend - PHP (Sample API)

<?php

// db.php - database connection

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "interacteach\_db";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die(json\_encode(['success'=>false, 'message'=>'DB connection failed']));

}

?>

attendance.php

<?php

// attendance.php - handle attendance submission

header('Content-Type: application/json');

require 'db.php';

$user\_id = $\_POST['user\_id'] ?? null;

$status = $\_POST['status'] ?? null;

if (!$user\_id || !$status) {

echo json\_encode(['success' => false, 'message' => 'Missing parameters']);

exit;

}

$stmt = $conn->prepare("INSERT INTO attendance (user\_id, status, timestamp) VALUES (?, ?, NOW())");

$stmt->bind\_param("is", $user\_id, $status);

if ($stmt->execute()) {

echo json\_encode(['success' => true, 'message' => 'Attendance submitted']);

} else {

echo json\_encode(['success' => false, 'message' => 'DB insert failed']);

}

?>

#### Database Schema / ERD

SQL to create required tables (MySQL):

-- SQL Dump: interacteach\_db

CREATE DATABASE IF NOT EXISTS interacteach\_db;

USE interacteach\_db;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

password VARCHAR(255),

role ENUM('student','teacher','admin') DEFAULT 'student',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE attendance (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

status ENUM('present','absent','late') DEFAULT 'present',

timestamp DATETIME,

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

);

CREATE TABLE quizzes (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255),

content TEXT,

created\_by INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE analytics\_log (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

event VARCHAR(255),

details TEXT,

event\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Chapter 5: Results and Testing**

Test Cases (examples):

1. Login with valid credentials => Expect: success and user data returned.

2. Submit attendance (present) => Expect: success message and DB record created.

3. View analytics dashboard => Expect: aggregated counts of attendance by date.

### Sample Test Data

INSERT INTO users (name, email, password, role) VALUES   
('Juan Dela Cruz','juan@example.com','[hashed\_password]','student'),  
('Maria Santos','maria@example.com','[hashed\_password]','teacher');

### Common Bugs and Fixes

- Issue: AI returns irrelevant quiz items. Fix: tune prompt and filter by keywords.

- Issue: Slow dashboard queries. Fix: add indexes and optimize SQL queries.

- Issue: CORS errors from Flutter app. Fix: enable CORS headers on API server.

**Chapter 6: Conclusion and Recommendations**

InteracTeach+ offers a mobile-capable solution to reduce teacher workload and increase student engagement through automation, AI, and gamification.

Recommendations: Pilot testing, expand AI for open-ended responses, add accessibility features, integrate GPS auto-detect for attendance, and add export to Excel/CSV functionality.

**References**

- Al-Fraihat, D., Joy, M., & Sinclair, J. (2020). Evaluating E-learning systems success.

- Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does Gamification Work?

- Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2016). Intelligence Unleashed.

- Flutter.dev / developer.android.com

- W3Schools (for PHP examples)

## Appendices

Appendix A: Full source code is included in repo or below sections.

Appendix B: SQL Dump file content included above.

Appendix C: Test data and screenshots.