Object Oriented Programming

(TFB1033)

Java Android Application

PROJECT PROPOSAL

ABSTRACT

Giving students varied tasks is one of the typical ways to see how information is affecting them at every educational level. As part of their duties, teachers must assign tasks to their students and evaluate their solutions. A growing number of mobile applications are being created and released every day as a result of the advancement of technology, with Android operating system applications dominating the market.

The major goal of this project was to create a mobile Android application that would let teachers quickly assign homework to students and monitor their participation in submitting accurate and comprehensive answers. As a mobile application, the program can be utilized anywhere.

The application gives the teacher the ability to register, add a course, add an assignment, edit an assignment, delete an assignment, mark an assignment for the student with the right assignment answer, and check the submission and completion rates of student assignments. While the application enables the student to sign up, enroll in the course, check the availability and submission status of assignments, and examine the pace of assignment submission. Possibly included in the intended category are UTP teachers and students.

<span style="text-decoration:underline">**CONTENTS**</span>

1 . **Abstract** ………………………………………………….……. 2

2 . **Introduction** ……………..…………………………………. 4

- 1.1 Our Team Members ……………………………… 4
- 1.2 Objective ……………………………………….... 4
- 1.3 Description ……………………………………… 5
- 1.4 What We Need …………………………………… 5
- 1.5 The Process ……………………………………… 5
- 1.6 Flow of the Application …………………………… 6

INTRODUCTION

The major goal of this project is to create a mobile application for tracking assignments that will enable teachers to assign homework to students and monitor how often they receive it back with the necessary and complete answers. The context, purpose, and goals of this study are discussed in this section.

1.1 Our Team Members

Muhammad Syahmi Nuaim bin Anam	21000530
Muhammad Farhan Asreen Bin Mohd Asri	21000222
Muhammad Amir Syauqi Bin Johari	21000173
Adil Haakim Bin Rosli	22007233
Jad Hazrein bin Harun	21002022
Megat Ammar Zuhair Bin Ismail	21001233
Nik Muhammad Arief Danial Bin Yusran	21001790
Syed Arif Daniel Bin Syed Amran	21001080
Misbahul Munir bin Che Omar	21002004
Muhammad Aliff Hazmi bin Mohd Hair	21001356
Muhammad Emir Faiq bin Abdullah	24607

1.2 Objective

The goal of this project was to give UTP students a chance to engage in a wide range of activities, such as registering to use the application's features and functionality, selecting a course from a list and enrolling in it, receiving available assignments and his or her submitted assignments on the courses the students have enrolled in, and having the opportunity to submit their own work.

1.3 Description

The creation of an Android mobile application for an assignment tracking system was the main goal of this project. This project is to be created primarily for the purpose of recording student solutions to coursework, and it should be created as a mobile application to improve the program's accessibility and facilitate tracking assignment solutions for its users, who are likely to be college students and teachers.

1.4 What We Need

1. A development team
2. Development tools
3. Database management system
4. Access to the UTP's authentication system

1.5 The Process

1. Creating the application's user interface and user experience.
2. Establishing the backend database to hold data on teachers, students, classes, and assignments.
3. Creating the tools for teachers to register, create courses, assignments, update them, and delete them.
4. Setting up the tool that allows teachers to grade assignments and monitor submission and completion rates.
5. Creating features for enrollment, assignment submission, assignment availability check, and student registration.
6. Linking the app to the UTP's teacher and student authentication system.
7. Ensuring compatibility and resolving bugs.

1.6 Flow of the Application

- The user launches the app and is presented with a login screen.
- If the user is a teacher, they will enter their UTP credentials to log in.
- If the user is a student, they will enter their UTP credentials to log in.
- Once logged in, the teacher will be taken to the teacher dashboard where they can view a list of their courses.
- The teacher can select a course to view the assignments for that course.
- The teacher can create a new assignment by filling out a form with assignment details such as title, description, due date, and possible answers.
- The teacher can edit or delete an existing assignment.
- The teacher can mark an assignment as complete or incomplete for each student.
- The teacher can view the submission and completion rates for each assignment.
- The student will be taken to the student dashboard where they can view a list of their enrolled courses.
- The student can select a course to view the assignments for that course.
- The student can view the details of an assignment, such as the title, description, and due date.
- The student can submit their answers for an assignment.
- The student can view the submission status of an assignment.

DATABASE

3.1 Relational Schema

TEACHER(TEACHER_ID, NAME, EMAIL, UTP_CREDENTIALS)

TEACHER_ID is the primary key

NAME, EMAIL, and UTP_CREDENTIALS are the attributes of the teacher

COURSE(COURSE_ID, COURSE_NAME, COURSE_CODE, TEACHER_ID)

COURSE_ID is the primary key

COURSE_NAME, COURSE_CODE are the attributes of the course

TEACHER_ID is a foreign key referencing TEACHER_ID in Teacher table

ASSIGNMENT(ASSIGNMENT_ID, TITLE, DESCRIPTION, DUE_DATE, POSSIBLE_ANSWERS, COURSE_ID)

ASSIGNMENT_ID is the primary key

TITLE, DESCRIPTION, DUE_DATE, POSSIBLE_ANSWERS are the attributes of the assignment

COURSE_ID is a foreign key referencing the COURSE_ID in Course table

STUDENT(STUDENT_ID, NAME, EMAIL, UTP_CREDENTIALS)

STUDENT_ID is the primary key

name, email, and UTP_CREDENTIALS are the attributes of the student

ENROLLED(STUDENT_ID, COURSE_ID)

(STUDENT_ID, COURSE_ID) is a composite primary key

STUDENT_ID is a foreign key referencing the STUDENT_ID in the Student table

COURSE_ID is a foreign key referencing the COURSE_ID in the Course table

SUBMISSION(STUDENT_ID, ASSIGNMENT_ID, ANSWERS, SUBMISSION_STATUS)

(STUDENT_ID, ASSIGNMENT_ID) is a composite primary key

STUDENT_ID is a foreign key referencing the STUDENT_ID in the Student table

ASSIGNMENT_ID is a foreign key referencing the ASSIGNMENT_ID in the Assignment table

SUBMISSION_STATUS is an attribute which indicates the status of the submission

MARKING(STUDENT_ID, ASSIGNMENT_ID, STATUS, GRADE)

(STUDENT_ID, ASSIGNMENT_ID) is a composite primary key

STUDENT_ID is a foreign key referencing the STUDENT_ID in the Student TABLE

ASSIGNMENT_ID is a foreign key referencing the ASSIGNMENT_ID in the Assignment table

STATUS is an attribute which indicates the status of the assignment (Complete or Incomplete)

GRADE is an attribute which indicates the grade obtained by the student

3.2 SQL Code

```
CREATE TABLE TEACHER (
TEACHER_ID INT PRIMARY KEY,
NAME VARCHAR(255),
EMAIL VARCHAR(255),
UTP_CREDENTIALS VARCHAR(255)
);

CREATE TABLE COURSE (
COURSE_ID INT PRIMARY KEY,
COURSE_NAME VARCHAR(255),
COURSE_CODE VARCHAR(255),
TEACHER_ID INT,
FOREIGN KEY (TEACHER_ID) REFERENCES TEACHER(TEACHER_ID)
);

CREATE TABLE ASSIGNMENT (
ASSIGNMENT_ID INT PRIMARY KEY,
TITLE VARCHAR(255),
DESCRIPTION VARCHAR(255),
DUE_DATE DATE,
POSSIBLE_ANSWERS VARCHAR(255),
COURSE_ID INT,
FOREIGN KEY (COURSE_ID) REFERENCES
COURSE(COURSE_ID)
);

CREATE TABLE STUDENT (
STUDENT_ID INT PRIMARY KEY,
NAME VARCHAR(255),
EMAIL VARCHAR(255),
UTP_CREDENTIALS VARCHAR(255)
);

CREATE TABLE ENROLLED (
STUDENT_ID INT,
COURSE_ID INT,
PRIMARY KEY (STUDENT_ID, COURSE_ID),
FOREIGN KEY (STUDENT_ID) REFERENCES STUDENT(STUDENT_ID),
FOREIGN KEY (COURSE_ID) REFERENCES COURSE(COURSE_ID)
);
```

```
CREATE TABLE SUBMISSION (
STUDENT_ID INT,
ASSIGNMENT_ID INT,
ANSWERS VARCHAR(255),
SUBMISSION_STATUS VARCHAR(255),
PRIMARY KEY (STUDENT_ID, ASSIGNMENT_ID),
FOREIGN KEY (STUDENT_ID) REFERENCES STUDENT(STUDENT_ID),
FOREIGN KEY (ASSIGNMENT_ID) REFERENCES ASSIGNMENT(ASSIGNMENT_ID)
);

CREATE TABLE MARKING (
STUDENT_ID INT,
ASSIGNMENT_ID INT,
STATUS VARCHAR(255),
GRADE INT,
PRIMARY KEY (STUDENT_ID, ASSIGNMENT_ID),
FOREIGN KEY (STUDENT_ID) REFERENCES STUDENT(STUDENT_ID),
FOREIGN KEY (ASSIGNMENT_ID) REFERENCES ASSIGNMENT(ASSIGNMENT_ID)
);
```