# ASSIGNMENT No. 3

**Complex Computing Problem**

Course Title: Mobile Application Development Ashir Azeem: 02-134211-039

Course Code: CSC-341 Kabeer M Mansoor: 02-134211-090

Class: BS (CS)-6A Submission deadline: 25Dec23

Course Instructor: Fasiha Ikram Marks: 10

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***Instructions***

1. **This assignment is a complex computing problem which can be performed individually or in a group of 2 students.**
2. **Assignments should be done only on A4 size paper and will also be uploaded on LMS.**
3. **Deadline will not be extended for any reason.**
4. **Copied assignments would have zero marks.**
5. **Individual efforts would be appreciated.**
6. **Name, class, section, department and roll number on the sheets must be mentioned correctly.**
7. **Make a single PDF file as both soft copies and hard copies are mandatory.**

**Solutions must be designed by applying the following attributes/characteristics.**

|  |  |
| --- | --- |
| **Attribute** | **Problem Solving description** |
| **Depth of analysis required** | **Has no obvious solution, and requires conceptual thinking and innovative analysis to formulate suitable abstract models** |
| **Depth of knowledge required** | **A solution requires the use of in-depth computing or domain knowledge and an analytical approach that is based on well-founded principles** |
| **Level of problem** | **Is outside problems encompassed by standards and standard practice for professional computing** |
| **Consequences** | **Has significant consequences in a range of contexts** |

**[CLO-2] [PLO-3] [C3]**

1. **Scenario/Problem**

"Learn n' Play" is an engaging mobile application designed for kindergarten children to learn and practice essential skills such as alphabets, shapes, colors, and numbers. The app incorporates interactive games, puzzles, and activities to make learning enjoyable and effective. Following are the functionalities of the app.

1. User Registration and Profile Creation:
2. Alphabet Learning:
3. Shape and Colour Recognition:
4. Number Skills Development:
5. Progress Tracking and Rewards:
6. Parental Involvement and Feedback:

By combining fun and interactive elements with educational content, the "Learn n' Play" app creates an engaging learning environment for kindergarten children to acquire and strengthen their foundational skills in alphabets, shapes, colours, and numbers.

An educational mobile application for kindergarten children called "Learn n' Play." The app focuses on essential skills such as alphabets, shapes, colors, and numbers, incorporating interactive games, puzzles, and activities to make learning enjoyable. Additionally, the app features a background service that repeats tasks every 10 minutes, recognizing the need for repetition in a child's learning process. It also displays questions periodically, and detailed reports are sent to parents via email.

* Educational Content: Alphabets, Shapes, Colors, Numbers.
* Interactive Games, Puzzles, and Activities:Engaging elements to make learning fun for children.
* Background Service: Runs periodically (every 10 minutes) to repeat tasks. Recognizes the importance of repetition in a child's learning process.
* Memory Enhancement: Acknowledges the limited memory capacity of children. Suggests the repetition of content to enhance memory retention.
* Question Display: Displays questions either after every 10 minutes or while the mobile is in use.
* Parental Reporting: Detailed reports on the child's progress.Reports are sent to parents via email.
* To improve and expand on your concept, you might consider:
* Adaptive Learning: Tailor the difficulty of questions based on the child's performance.
* Multimedia Content: Incorporate audio, visuals, and interactive elements to enhance engagement.
* Parent-Teacher Interaction: Provide a platform for parents and teachers to communicate regarding the child's progress.
* Customization: Allow parents to customize the learning path based on the child's individual needs and preferences.
* Gamification: Introduce a reward system or badges to motivate children to complete tasks.
* Offline Mode: Consider including an offline mode for parents who may not have consistent internet access.
* Security and Privacy: Ensure robust security measures to protect children's data and privacy.
* Remember to test the app with the target audience to gather feedback for further improvements.

1. **Deliverables**
2. *Consider the given scenario and identify the development environment for the given problem, either native or hybrid. Justify your answer.*
3. *Storyboard for system and widget with database description and with maximum Nine (9) Activates.*
4. *Identify the broadcast receiver and background services that should be implanted in this project. Justify the need of selected broadcast receivers.*
5. *From the answer of part b. select any activity and implement database handling.*
6. *From the answer of part c, select any activity and implement broadcast receiver and background service.*
7. **Evaluation Criteria**

|  |  |  |
| --- | --- | --- |
| **1.** | *Storyboard for system and widget with database description* | 1 mark |
| **2.** | *Identify and implement the broadcast receiver.* | 2 marks |
| **3.** | *Identify and implement the background service.* | 2 marks |
| **4.** | Identify and implement the database handling. | 2 marks |
| **5.** | Viva + running application with error handling | 3 marks |
|  |  |  |

**Deliverable # 1**

*Consider the given scenario and identify the development environment for the given problem, either native or hybrid. Justify your answer*

**Detailed Comparison of Native vs. Hybrid Development for "Learn n' Play" App**

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Native Development | Hybrid Development | Justification for "Learn n' Play" |
| **Performance** | - Excellent responsiveness and low latency. - Utilizes platform-specific APIs and components for optimal performance. | - Good performance for basic functionalities. - Might experience performance issues with complex elements or resource-intensive tasks due to abstraction layer. | - **Critical for interactive learning experience and smooth operation (e.g., animations, games).** - Performance limitations could negatively impact user engagement and learning outcomes. |
| **User Experience** | - Seamless integration with native platform elements and design guidelines. - Feels "natural" and familiar to users on each platform. | - Good overall user experience. - Might lack platform-specific polish and responsiveness due to reliance on webview. | - **Prioritize a well-integrated and engaging user experience to keep children motivated and focused.** - Hybrid limitations could make the app feel less engaging. |
| **Cross-Platform Compatibility** | - Requires separate development for each platform (iOS, Android) with platform-specific languages and tools. | - Single codebase runs on multiple platforms, reducing development time and cost. | - **Not a primary concern as the app targets specific platforms (likely iOS and Android).** - Native approach offers full control over each platform's experience. |
| **Development Time & Cost** | - Longer development time due to separate codebases. - Higher overall cost due to platform-specific expertise required. | - Faster development and lower cost due to single codebase. | - **Time and budget may be tight, but prioritizing native ensures optimal performance and user experience.** - Consider hybrid if resources are extremely limited. |
| **Maintenance & Updates** | - Separate maintenance and updates required for each platform version. | - Unified updates applied across all platforms, simplifying maintenance. | - **Frequent updates might be needed for bug fixes and new features.** - Native requires more maintenance effort, but hybrid updates might not fully address platform-specific issues. |
| **Access to Device Features** | - Full access to native device features like camera, GPS, sensors, etc. | - Limited access to certain features, requiring workarounds or plugins. | - **Background services, detailed reporting, and potential use of device features necessitate full access.** - Limited access in hybrid could hinder app functionality and reporting accuracy. |

**B) STORY BOARD:**

**A white board with writing on it

Description automatically generated**

**SCREENS:**

|  |  |  |
| --- | --- | --- |
| **A blue and yellow cover with yellow letters  Description automatically generated** | **A screenshot of a login screen  Description automatically generated** | **A screen shot of a sign up form  Description automatically generated** |
| **A screenshot of a game  Description automatically generated** | **A screenshot of a game  Description automatically generated** | **A screenshot of a phone  Description automatically generated** |
| **A screenshot of a game  Description automatically generated** | **A screenshot of a game  Description automatically generated** | **A screenshot of a video game  Description automatically generated** |
|  |  |  |
| **A screenshot of a login screen  Description automatically generated** | **A screenshot of a phone  Description automatically generated** |  |

***c) Identify the broadcast receiver and background services that should be implanted in this project. Justify the need of selected broadcast receivers.***

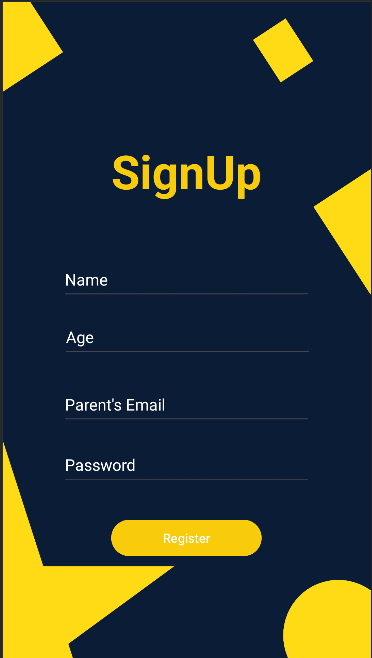
***BROADCAST RECEIVER:***

package com.example.learnnplay;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
  
public class MainActivity extends AppCompatActivity {  
 TextView alphabetTextView;  
 TextView numbersTextView;  
 TextView colorsTextView;  
 TextView shapesTextView;  
  
 TextView profileSettingTextView,quizTextView;  
 ImageView settingImageView;  
  
 TextView rateTextView;  
  
 private MyBroadCastReciever receiver;  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 shapesTextView = findViewById(R.id.*shapesTextView*);  
 colorsTextView = findViewById(R.id.*colorsTextView*);  
 alphabetTextView = findViewById(R.id.*alphabetsTextView*);  
 numbersTextView = findViewById(R.id.*numbersTextView*);  
 settingImageView = findViewById(R.id.*settingImageView*);  
 rateTextView = findViewById(R.id.*rateTextView*);  
 quizTextView = findViewById(R.id.*quizTextView*);  
  
 shapesTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,ShapeLearning.class));  
 finish();  
 }  
 });  
 colorsTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,ColorLearning.class));  
 finish();  
  
 }  
 });  
 alphabetTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,AlphabetLearning.class));  
 finish();  
 }  
 });  
 numbersTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,NumbersLearning.class));  
 finish();  
 }  
 });  
 settingImageView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,ProfileSettings.class));  
 finish();  
 }  
 });  
 rateTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,RateMyApp.class));  
 finish();  
 }  
 });  
 quizTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this, Quiz.class));  
 finish();  
 }  
 });  
 receiver = new MyBroadCastReciever();  
 IntentFilter filter = new IntentFilter();  
 filter.addAction(Intent.*ACTION\_AIRPLANE\_MODE\_CHANGED*);  
 registerReceiver(receiver,filter);  
  
  
  
  
  
 }  
 private void broadcastIntent() {  
 sendBroadcast(new  
 Intent("com.example.broadcastreceivers.MY\_CUSTOM\_ACTION"));  
 }  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 unregisterReceiver(receiver);  
 }  
  
}

**JUSTIFICATION:**

The purpose of this broadcast receiver seems to be to respond to changes in the airplane mode status on the device. When the airplane mode is toggled, the receiver's ‘**onReceive**’ method is likely triggered to perform some actions.

**d) *From the answer of part b. select any activity and implement database handling.***

**

***DATABASE:***

package com.example.learnnplay;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.util.Log;

public class UserProfileDBHelper extends SQLiteOpenHelper {

private static final String DATABASE\_NAME = "UserProfilesDB";

private static final int DATABASE\_VERSION = 1;

private static final String TABLE\_NAME = "UserProfiles";

private static final String COLUMN\_ID = "id";

private static final String COLUMN\_NAME = "name";

private static final String COLUMN\_AGE = "age";

private static final String COLUMN\_EMAIL = "email";

private static final String COLUMN\_PASSWORD = "password";

public UserProfileDBHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

String createTableQuery = "CREATE TABLE IF NOT EXISTS " + TABLE\_NAME + " (" +

COLUMN\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +

COLUMN\_NAME + " TEXT, " +

COLUMN\_AGE + " TEXT, " +

COLUMN\_EMAIL + " TEXT, " +

COLUMN\_PASSWORD + " TEXT)"; // Include password column in table creation

db.execSQL(createTableQuery);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

onCreate(db);

}

public boolean insertUserProfile(String name, String age, String email, String password) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues contentValues = new ContentValues();

contentValues.put(COLUMN\_NAME, name);

contentValues.put(COLUMN\_AGE, age);

contentValues.put(COLUMN\_EMAIL, email);

contentValues.put(COLUMN\_PASSWORD, password);

long result = db.insert(TABLE\_NAME, null, contentValues);

if (result == -1) {

Log.e("DB\_INSERT\_ERROR", "Error inserting user profile");

}

return result != -1;

}

public boolean checkUserCredentials(String email, String password) {

SQLiteDatabase db = this.getReadableDatabase();

String[] columns = {COLUMN\_EMAIL};

String selection = COLUMN\_EMAIL + " = ?" + " AND " + COLUMN\_PASSWORD + " = ?";

String[] selectionArgs = {email, password};

Cursor cursor = db.query(TABLE\_NAME, columns, selection, selectionArgs,

null, null, null);

int count = cursor.getCount();

cursor.close();

return count > 0;

}

}

*package com.example.learnnplay;*

*import android.content.Intent;*

*import android.os.Bundle;*

*import android.widget.Button;*

*import android.widget.EditText;*

*import android.widget.Toast;*

*import androidx.appcompat.app.AppCompatActivity;*

*public class RegistrationActivity extends AppCompatActivity {*

*EditText editTextName, editTextAge, editTextEmail,editTextPass;*

*UserProfileDBHelper dbHelper;*

*@Override*

*protected void onCreate(Bundle savedInstanceState) {*

*super.onCreate(savedInstanceState);*

*setContentView(R.layout.activity\_registration);*

*// Initialize EditText fields and Database Helper*

*editTextName = findViewById(R.id.editTextName);*

*editTextAge = findViewById(R.id.editTextAge);*

*editTextEmail = findViewById(R.id.editTextEmail);*

*editTextPass = findViewById(R.id.editTextPass);*

*dbHelper = new UserProfileDBHelper(this);*

*Button btnRegister = findViewById(R.id.btnRegister);*

*btnRegister.setOnClickListener(v -> {*

*// Retrieve input data from EditText fields*

*String name = editTextName.getText().toString().trim();*

*String age = editTextAge.getText().toString().trim();*

*String email = editTextEmail.getText().toString().trim();*

*String password = editTextPass.getText().toString().trim();*

*// Validate input fields*

*if (name.isEmpty() || age.isEmpty() || email.isEmpty() || password.isEmpty()) {*

*Toast.makeText(RegistrationActivity.this, "Please fill all fields", Toast.LENGTH\_SHORT).show();*

*return;*

*}*

*// Validate age as a number*

*if (!age.matches("\\d+")) {*

*Toast.makeText(RegistrationActivity.this, "Enter a valid age", Toast.LENGTH\_SHORT).show();*

*return;*

*}*

*// Insert user profile using Database Helper*

*boolean success = dbHelper.insertUserProfile(name, age, email,password);*

*if (success) {*

*Toast.makeText(RegistrationActivity.this, "Registration Successful", Toast.LENGTH\_SHORT).show();*

*startActivity(new Intent(RegistrationActivity.this, MainActivity.class));*

*finish();*

*} else {*

*Toast.makeText(RegistrationActivity.this, "Error saving profile", Toast.LENGTH\_SHORT).show();*

*}*

*});*

*}*

*}*

1. **From the answer of part c, select any activity and implement broadcast receiver and background service.**



package com.example.learnnplay;  
  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.provider.Settings;  
import android.widget.Toast;  
public class MyBroadCastReciever extends BroadcastReceiver {  
 @Override  
 public void onReceive(Conte package com.example.learnnplay;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
  
public class MainActivity extends AppCompatActivity {  
 TextView alphabetTextView;  
 TextView numbersTextView;  
 TextView colorsTextView;  
 TextView shapesTextView;  
  
 TextView profileSettingTextView,quizTextView;  
 ImageView settingImageView;  
  
 TextView rateTextView;  
  
 private MyBroadCastReciever receiver;  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 shapesTextView = findViewById(R.id.*shapesTextView*);  
 colorsTextView = findViewById(R.id.*colorsTextView*);  
 alphabetTextView = findViewById(R.id.*alphabetsTextView*);  
 numbersTextView = findViewById(R.id.*numbersTextView*);  
 settingImageView = findViewById(R.id.*settingImageView*);  
 rateTextView = findViewById(R.id.*rateTextView*);  
 quizTextView = findViewById(R.id.*quizTextView*);  
  
 shapesTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,ShapeLearning.class));  
 finish();  
 }  
 });  
 colorsTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,ColorLearning.class));  
 finish();  
  
 }  
 });  
 alphabetTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,AlphabetLearning.class));  
 finish();  
 }  
 });  
 numbersTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,NumbersLearning.class));  
 finish();  
 }  
 });  
 settingImageView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,ProfileSettings.class));  
 finish();  
 }  
 });  
 rateTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,RateMyApp.class));  
 finish();  
 }  
 });  
 quizTextView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this, Quiz.class));  
 finish();  
 }  
 });  
 receiver = new MyBroadCastReciever();  
 IntentFilter filter = new IntentFilter();  
 filter.addAction(Intent.*ACTION\_AIRPLANE\_MODE\_CHANGED*);  
 registerReceiver(receiver,filter);  
  
  
  
  
  
 }  
 private void broadcastIntent() {  
 sendBroadcast(new  
 Intent("com.example.broadcastreceivers.MY\_CUSTOM\_ACTION"));  
 }  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 unregisterReceiver(receiver);  
 }  
  
}

xt context, Intent intent) {  
 String action = intent.getAction();  
 if (action != null) {  
 switch (action) {  
 case Intent.*ACTION\_AIRPLANE\_MODE\_CHANGED*:  
 boolean isAirplaneModeOn =  
 intent.getBooleanExtra("state", false);  
 String message = isAirplaneModeOn ? "Airplane mode turned on" : "Airplane mode turned off";  
 showToast(context, message);  
 break;  
 }  
 }  
 }  
  
 private void showToast(Context context, String message) {  
 Toast.*makeText*(context, message, Toast.*LENGTH\_LONG*).show();  
}  
}