FACULTY OF INFORMATION TECHNOLOGY

LIST OF FINAL PROJECT TOPICS

SEMESTER 1A SCHOOL YEAR 2025 - 2026

Subject Code : ITE1221E
Subject : Mobile Device Programming
Class/Group :
Lecturer :

General requirements:

- Students must submit the following to the Learning Management System (LMS):
  - A report (Word, PDF format),
  - A presentation (in PPT or PPTX format),
  - The source code of the project.
- If you are found to have committed academic dishonesty on the project, such as using someone else's source code or report, you will fail the course and receive a score of 0.

Topic 1. Coloring shapes application

Problem description:

Develop an android application to color the shapes. The shapes, color, duration of maximum time of coloring task are stored in database. The application supports two types of users: participants (who play the task) and admins (who manage the setting). Key functions include login, doing tasks, managing user information, and advanced features such as sending notification when time up, uploading images, displaying history, scoring mark, and supporting multiple languages. The application must have a user-friendly, responsive interface, ensure basic security, and be scalable.

Objectives:

- **Academic:** Reinforce knowledge of mobile programming, including the Model-View-ViewModel (MVVM) model, the MVP approach, Service, Foreground Service, Notification, database management, and basic security.
- **Practical skills:** Develop skills in UI/UX design, architecture design, mobile programming, database integration, and teamwork using Git.
- **Real-world application:** Build an application simulating real-world education application, familiarizing students with commercial application development requirements.

Requirements:

1. Technologies used

- **Android**: view layout, background service, database, notification

- **Database**:SQLite
- **Project building tool**: Gradle.
- **Version control**: Use Git (submit source code via GitHub/GitLab).
- **Additional libraries**: Color, Palette

2. Main Functions

a. Users

❖ **Basic functions (45%)**

- Registration/Login/Logout:
  - Register an account with email, password, full name, year of birth, gender.
  - Log in using email and password.
- Task doing:
  - View a list of upcoming or past tasks.
  - View task details (name, description, maximum duration, color).
- Scoring:
  - Update score of each task.
  - View history of done task.

❖ **Advanced functions (10%)**

- Support multilingual interface (English and Vietnamese)
- Log in using email (UEF).
- Task
  - Flexible in choosing color, shape of brushes
- Notification:
  - Use visual effects for a "time's up" notification
  - Allow check-in by scanning face
  - Allow celebration effects when user in top 3, 10, 100

b. Administrators

❖ **Basic Functions (35%)**

- Task management:
  - Add, edit, or delete task information (name, description, maximun duration, color)
  - Set timer count up or down
- Shape management:
  - Add, edit, or delete categories (e.g., triangle, rectangle).
- Participant management:
  - View the list of registered participants
- Reporting:

- View the list of top 3, 10, 100 of hight score participants.

❖ **Advanced Functions (10%)**

- Participant management:
  - View product of paticipants
- Reporting:
  - View reports base on age, gender, average score

Topic 2. Blood Bank Management System (Android Application)

Problem Statement

Blood is a critical need in emergency medical situations such as surgeries, accidents, and certain diseases. However, patients often face challenges in finding required blood groups on time due to lack of proper information, communication gaps, and unorganized donor records. Traditional systems rely on manual processes and outdated methods, leading to delays and inefficiency. Therefore, there is a need for a mobile-based system that connects donors, patients, and blood banks in real time, ensuring faster accessibility and transparency.

Objectives

- To provide a mobile application for easy donor registration and management.
- To facilitate real-time blood requests by patients or hospitals.
- To maintain up-to-date inventory of blood units in blood banks.
- To implement a search and match system for required blood groups.
- To notify donors about urgent requests and donation eligibility through push notifications.
- To improve overall efficiency, accessibility, and transparency in blood management

Requirements:

1. Scope

- The app will be available on Android devices.
- Registered donors, recipients, and blood banks can use the system.
- The app will allow real-time stock updates of blood availability.
- Notifications will be sent to donors in case of urgent need.
- Can be extended to multi-blood bank networks and geolocation services.

2. Technologies Used

Java / Kotlin	Android app development
XML (Material Design)	UI Design for Android layouts
Android Studio	Development environment
SQLite	Database to store donor, patient, and blood bank data
Firebase Cloud Messaging	Push notifications

3. Expected Outcomes

- A **fully functional Android app** that allows donor registration, patient requests, and blood bank inventory management.
- **Faster and more reliable** access to blood units in emergencies.
- **Reduced manual errors** and improved transparency.
- A **centralized database** accessible from anywhere.

4. Modules / Features

- **User Authentication Module**: Login/Sign-up (Patients, Donors, Blood Bank Staff).
- **Donor Module**: Registration, profile management, donation history.
- **Recipient Module**: Request for blood, check status, search for availability.
- **Blood Bank Module**: Manage inventory, approve/reject requests, update stock.
- **Admin Module**: Manage users, monitor requests, generate reports.
- **Notification Module**: Push notifications for urgent requests, donation reminders.

Topic 3. Women Safety Application (Android Platform)

Problem Statement

Women's safety is a serious concern in today's society. In emergency situations such as harassment, assault, or accidents, women often find it difficult to reach out for immediate help. Traditional methods like calling or texting may take time and may not work under stress. Hence, there is a need for a mobile-based Women Safety App that allows users to quickly alert trusted contacts and authorities, share live location, and provide immediate assistance at the press of a button.

Objectives

- To provide a **panic button (SOS feature)** that instantly sends alerts.
- To enable **real-time location sharing** with emergency contacts.
- To allow users to **call police, emergency helplines, or family** directly through the app.
- To ensure **quick communication** via SMS/notifications during emergencies.
- To provide an **easy-to-use interface** accessible in high-stress conditions.
- To increase the sense of **safety and confidence** among women.

Requirements:

1. Scope

- Designed for **Android smartphones**.
- Accessible by **women of all age groups** with basic smartphone knowledge.
- Provides **emergency SOS alerts, live tracking, and direct calls**.
- Can be extended to **integration with government helpline numbers**.

- Useful not only for women but also for **elderly citizens and children**.

2. Technologies Used

Java / Kotlin	Android app development
XML (Material Design)	UI/UX design
Android Studio	Development environment
Firebase / SQLite	Storing user profiles and emergency contacts
Google Maps API	Live location tracking
SMS Manager / Twilio API	Sending SOS alerts via SMS
Firebase Cloud Messaging	Push notifications

3. Modules / Features

- **User Registration & Login Module**: Profile creation with name, emergency contacts, medical info.
- **SOS / Panic Button Module**: One-tap SOS that sends SMS with live location to emergency contacts.
- **Location Tracking Module**: Continuous GPS tracking until help arrives.
- **Emergency Call Module**: Quick dial to police (100/112), ambulance, or relatives.
- **Shake/Voice Activation Module** *(optional advanced feature)*: Trigger SOS by shaking the phone or speaking a keyword.
- **Admin / Helpline Module** *(optional)*: Connection with official women helplines or NGOs.

Topic 4. Personalized News Feed Application

Problem Statement

With the rapid growth of online media, users are overwhelmed by the massive amount of news available on the internet. Traditional news applications provide generic feeds without considering the user's personal interests, which leads to information overload and lack of engagement. There is a need for a mobile application that provides users with a personalized news feed, curated according to their preferences, interests, and reading habits

Objectives

- To develop an Android app that fetches live news articles from online sources via APIs.
- To allow users to select preferred categories (e.g., Sports, Technology, Politics, Health, Entertainment).
- To implement a recommendation system that tailors news feeds based on user preferences and reading history.
- To provide search and filter options for quick news discovery.
- To enable **offline reading** by saving articles.
- To ensure a **user-friendly interface** with real-time updates

Requirements:

1. **Scope**
   - Android-based application for smartphones.
   - News fetched via **News APIs (e.g., NewsAPI.org, Google News API, or custom RSS feeds)**.
   - Personalization based on user-selected categories and engagement.
   - Optional integration with **AI/ML models** for smarter recommendations.
   - Can be extended to **multi-language news** and **regional sources**.
2. **Technologies Used**

Java / Kotlin	Android app development
XML (Material Design)	UI/UX design
Android Studio	Development environment
Firebase / SQLite	Storing user profiles and emergency contacts
Google Maps API	Live location tracking
SMS Manager / Twilio API	Sending SOS alerts via SMS
Firebase Cloud Messaging	Push notifications

3. **Modules / Features**
   - **User Authentication Module**: Login/Signup via email, Google, or Firebase Auth.
   - **Category Selection Module**: Users select preferred news categories during onboarding.
   - **News Feed Module**: Fetch and display news articles using APIs; Infinite scroll / refresh feature.
   - **Recommendation Module** *(advanced)*: Learn user reading patterns and recommend relevant articles.
   - **Search & Filter Module**: Search news by keywords, category, or source.
   - **Save & Share Module**: Save articles for offline reading, share via social media.
   - **Notification Module**: Push alerts for breaking news or trending topics.

Topic 5. Personal Expense Tracker

Detailed Requirements

- Screen Management: Use Activities to manage different screens such as Add Expense, Expense List, and Expense Report.
- Layout Design: Implement layouts (e.g., LinearLayout, RelativeLayout, ConstraintLayout) for data entry and expense list display.

- Basic Views/Widgets: Utilize standard components including EditText (for entering names/amounts), Button (for add/save actions), Spinner (for selecting categories), and ListView or RecyclerView (for displaying expense lists).
- Local Data Storage: Employ an SQLite Database to ensure persistent storage of expense data.
- User Notifications: Use Toast messages for success/failure alerts and Dialog boxes for expense deletion confirmations.
- Data Transfer: Apply Intents to transfer data between Activities (e.g., from the Expense List screen to the Expense Detail/Edit screen).
- Data Visualization: Investigate and integrate charting libraries (e.g., MPAndroidChart) to visualize spending trends on a weekly, monthly, and yearly basis.
- Cloud Synchronization (Advanced): Explore Firebase Realtime Database or Firestore for data backup, enabling users to access their expenses across multiple devices.
- Architectural Patterns: Adopt architectural models such as MVVM (Model–View–ViewModel) or MVP (Model–View–Presenter) to improve code structure and maintainability.

Core Functions

- Record, categorize, and monitor personal expenses.
- Manage application screens, including adding new expenses, viewing expense lists, and generating reports.
- Design user interfaces for data entry and expense list display.
- Store expense data securely.
- Perform CRUD (Create, Read, Update, Delete) operations for expenses and categories.
- Provide success/failure notifications and confirmation prompts for expense deletion.
- Transfer data between screens (Activities).

Advanced Functions

- Integrate charting libraries to visualize spending trends on a weekly, monthly, and yearly basis.
- Enable cloud synchronization for data backup and multi-device accessibility.

- Apply architectural patterns such as MVVM (Model-View-ViewModel) or MVP (Model-View-Presenter) to improve code organization and maintainability.

Topic 6. Organ Donor Connect

Description

An application that connects organ donors, hospitals, and coordinators to handle donor registrations, transplant requests, and emergency coordination in real time, reducing waiting times and improving transparency.

Requirements

1. **Technologies**
   - **Kotlin/Java**
   - **Database: Room/SQLite (offline-first), Firestore/Realtime DB (synchronization)**
   - **Firebase Auth, Firebase Cloud Messaging (FCM)**
   - **Google Maps/Places (location, nearby hospitals)**
   - **WorkManager, Retrofit/OkHttp (if APIs are used)**
2. **Basic Functions**
   - Authentication (Donor/Hospital/Coordinator)
   - Donor profile: blood type, organ donation consent, status
   - Hospital: create and track transplant requests
   - Search & filter donors by minimum criteria (blood type, location)
   - Push notifications for emergencies and status updates
   - …
3. **Advanced Functions**
   - Matching engine based on compatibility (ABO/Rh, age, distance)
   - Geofencing for nearby donor prioritization
   - Audit trail & role-based access control (RBAC)
   - Reporting dashboard: number of cases, response time, match rates
   - Data encryption and privacy compliance
   - …

Topic 7. Vaccination & Health Camp Scheduler

Description

A scheduling solution for vaccination or health camps: manage campaigns, assign slots across multiple centers, allow citizens to book appointments, and send reminders.

Requirements

4. Technologies

- **Kotlin/Java + Material 3 UI**
- **Room/SQLite (offline-first) + Firestore (real-time sync)**
- **FCM (reminders), WorkManager (recurring tasks)**
- **QR Code library (ZXing/ML Kit), Cloud Storage**

5. Basic Functions

- Authentication (Citizen/Clinic/Admin)
- Campaign creation: define centers and available slots
- Appointment booking/modification, QR check-in at centers
- Vaccine inventory by batch (stock, usage, wastage)
- Notifications for next doses and post-vaccination surveys
- …

6. Advanced Functions

- Smart slot allocation: prioritize high-risk groups, location-based optimization
- Overload prevention: waiting lists, no-show handling
- Analytics: vaccination rate, no-show rate, stock per center
- Digital vaccination certificates (PDF) & personal health wallet
- …

EVALUATION CRITERIA AND SCALES

- **D1 (3 points):** Evaluation of the results of the project: (The results are application products, software, application programs, application models, application scenarios, demo scenarios, etc.). Based on small criteria:
  - The level of results meets the set requirements and functions.
  - Correctness and quality of results.
  - Practicality and convenience.
  - Apply new techniques and new technologies when building applications.
- **D2 (3 points):** Evaluation of the content of the report, presentation, layout, knowledge of research, research, etc. It can be based on the following criteria:
  - Content of research, research on the theoretical part, basic knowledge
  - The application of knowledge to the topic: system design analysis, database, etc.
  - Presentation, layout, formatting, spelling, references.
- **D3 (2 points):** Evaluate the direct report, answer questions with the Lecturer.
- **D4 (1 point)**: Evaluation of the spirit, attitude, working style, compliance with the progress and requirements of the teacher, **Individual contributions in the group**.
- **D5 (1 point):** Use AI tools (to develop functions in the application or utilize AI tools in the course). Students present to demonstrate this.

Ho Chi Minh City, on Sep 8, 2025
Head of Department/Department

Ho Chi Minh City, on Sep 8, 2025
Questioner