Digital Phenotyping for Early Detection of Student Stress – Project Plan

Phase 1: Research and Requirements (Month 1)

- **Kickoff & Team Setup:** Meet to define project scope, roles (e.g. UI dev, ML dev, backend dev, PM), and milestones. Prepare project management tools (GitHub repo, Trello board).
    - Tools: Zoom/Slack for meetings, Trello/GitHub Projects for task tracking.
    - Roles: All interns (led by a designated PM).
- **Literature Review:** Search academic databases (Google Scholar, PubMed) for terms like "digital phenotyping," "smartphone stress," and "student mental health"frontiersin.org. Compile key findings on using smartphone sensors for stress detection (e.g. GPS, accelerometer, step count)frontiersin.org. Summarize relevant models and EMA (survey) methods from the literature.
    - Tools: Google Scholar, Zotero/EndNote, PubMed.
    - Roles: ML dev (lead), all interns assist.
- **Sensor & Data Source Identification:** Identify which phone sensors and data streams to use. Likely sources: GPS (location patterns), accelerometer/gyroscope (movement), step counter, screen usage logs, call/text logs, ambient sound, etc. Reference studies that used these (e.g. "smartphones have multiple passive sensors … GPS, accelerometer, gyroscope, step detector"frontiersin.org). Decide on initial sensors to integrate.
    - Tools: Smartphone technical documentation, pub articles, Flutter plugin docs.
    - Roles: UI dev (investigate Flutter sensor plugins), ML dev (research feature relevance).
- **EMA Survey Design:** Define the active survey (Ecological Momentary Assessment) protocol. Select validated stress/mood questionnaires (e.g. Perceived Stress Scale, PANAS), and determine schedule (e.g. 3–5 times/day). Plan push-notification strategy for surveyspmc.ncbi.nlm.nih.gov.
    - Tools: Existing EMA literaturepmc.ncbi.nlm.nih.gov, mental health assessment resources.
    - Roles: ML dev (select questions), UI dev (design survey UI), Mentor (psychology advisor input).
- **Privacy, Ethics & Consent Planning:** Research privacy regulations (GDPR, institutional IRB rules) and ethical guidelines for digital phenotypingpmc.ncbi.nlm.nih.gov. Outline informed consent text and opt-in/opt-out flows. Plan data anonymization and security measures (deidentification, encryption).
    - Tools: IRB guidelines, ethics literaturepmc.ncbi.nlm.nih.gov.
    - Roles: Backend dev (lead, for compliance research), all interns.
- **Functional Requirements Specification:** Document app requirements and user stories (e.g. "As a student, I want to log my mood via quick surveys and see stress feedback"). Specify

features: passive logging, EMA delivery, simple dashboard. Specify ML goals (stress prediction).

    - Tools: Google Docs or similar.
    - Roles: All interns collaborate.

- **Project Planning:** Break work into sprints/weeks, set milestones for design review, prototype demo, pilot test, paper draft. Schedule weekly mentor check-ins for feedback.
    - Tools: Trello/GitHub Milestones, calendar.
    - Roles: All interns (PM sets schedule).
- **Mentor Review 1:** Present initial findings (literature summary, proposed sensors/surveys, timeline) to mentor. Incorporate feedback to refine scope and plan.

Phase 2: Design and Architecture (Month 1–2)

Figure: Example digital phenotyping platform architecture with passive and active data collection, secure storage, and analysisjmir.org. Key design tasks:

- **System Architecture Design:** Define the overall system components: Flutter mobile app, Firebase backend (Firestore database, authentication), ML analysis module (cloud or on-device). Draw architecture and data flow diagrams showing how sensor data and EMA responses move from phone to cloud and back.
    - Tools: Draw.io or Figma for diagrams.
    - Roles: All interns.
- **Tech Stack Finalization:** Confirm libraries and frameworks: Flutter for cross-platform app, Firebase/Firestore for backend, TensorFlow/PyTorch for ML, Flutter sensor/notification plugins (e.g. sensors_plus, geolocator, flutter_local_notifications). Decide on any third-party tools (e.g. CARP Mobile Sensing Flutter library).
    - Tools: Documentation for Flutter/Firebase/ML frameworks.
    - Roles: UI dev and Backend dev (led by both), ML dev.
- **Data Flow and Database Schema:** Design how data is structured and stored. Plan Firestore collections (e.g. users, sensor_logs, survey_responses). Define data schema (fields for timestamp, sensor values, EMA scores). Establish how data syncing works (offline support, batching).
    - Tools: Firestore schema design (tables/views), documentation.
    - Roles: Backend dev (lead), ML dev.
- **Consent & Privacy Flow Design:** Mock up user flows for onboarding and consent screens in the app. Ensure screens explain data collection in plain language. Plan how users opt-in or withdraw. Incorporate compliance (e.g. allow account deletion).
    - Tools: Figma wireframes.
    - Roles: UI dev (lead), Backend dev.
- **UI/UX Mockups:** Create mockups for key app screens: onboarding, home/dashboard (stress level display), survey questions, settings. Define basic navigation flow.
    - Tools: Figma or Adobe XD.
    - Roles: UI dev (lead), all interns provide input.
- **Feature Prioritization (MVP):** List minimum viable features for prototype (e.g. passive logging of 2 sensors + at least one survey, basic ML output). Reserve advanced features (e.g. social features) for later.
    - Tools: Spreadsheet or backlog document.
    - Roles: All interns.

- **Data Processing Plan:** Outline steps for preprocessing and feature engineering: e.g. aggregating sensor data into features (activity levels, mobility patterns), handling missing data, labeling stress via surveys. Sketch ML pipeline stages.
    - Tools: Jupyter notebooks (planning), Python libraries list.
    - Roles: ML dev (lead).
- **Preliminary UI Dashboard Concept:** If time permits, consider a web or in-app dashboard to visualize data. Plan charts (e.g. stress vs time graph).
    - Tools: Mockup tools (Figma).
    - Roles: UI dev.
- **Documentation:** Write up the design decisions (architecture diagram, DB schema, UI mockups) in a design document.
    - Tools: Google Docs/Confluence.
    - Roles: All interns.
- **Mentor Review 2:** Present architecture and design deliverables to mentor. Update designs per feedback before coding.

Phase 3: Development and Integration (Month 2–3)

- **Firebase & Backend Setup:** Create Firebase project and Firestore database. Enable Firebase Authentication (e.g. anonymous or email). Set up Firestore security rules (e.g. users see their own data).
    - Steps: Use Firebase console/CLI to initialize project. Define collections and indexes.
    - Tools: Firebase Console, Firebase CLI.
    - Roles: Backend dev (lead).
- **Initialize Flutter Project:** Scaffold a new Flutter app. Set up project structure (separate folders for screens, models, services). Add necessary dependencies (e.g. cloud_firestore, firebase_auth, firebase_messaging, sensors_plus, etc.).
    - Tools: Flutter SDK, Android Studio/VS Code.
    - Roles: UI dev.
- **Sensor Integration (Passive Data):** Integrate plugins for phone sensors: e.g. use sensors_plus for accelerometer/gyroscope, geolocator or location for GPS, and health or other APIs if needed. Request runtime permissions (location, activity recognition).
    - Steps: Implement code to listen to sensor streams and collect data periodically (e.g. every minute for location, accelerometer at intervals). Format data (timestamp, values).
    - Tools: Flutter plugins, testing on Android/iOS devices.
    - Roles: UI dev (lead), Backend dev (assist with Firebase saving).
- **EMA Survey Implementation (Active Data):** Implement survey feature: design survey questions in code, create Flutter forms for responses. Schedule and send survey notifications using Firebase Cloud Messaging (FCM) and local notifications (e.g. flutter_local_notifications)pmc.ncbi.nlm.nih.gov.
    - Steps: Use FCM to send push notifications at scheduled times (e.g. 4 times/day). On tap, open the survey screen. Save responses (with timestamp) to Firestore.
    - Tools: firebase_messaging, flutter_local_notifications.
    - Roles: UI dev (lead), ML dev (define questions, logic).
- **Data Upload and Sync:** Implement data syncing: write sensor and survey data to Firestore. Handle offline cases (cache in local storage or using Firestore offline persistence). Ensure minimal data is lost if offline.
    - Tools: Firebase SDK for Flutter.
    - Roles: Backend dev (lead), UI dev.
- **User Authentication & Profiles:** Integrate Firebase Auth. On first open, assign user a unique ID. Allow optional login if needed. Link all data to this user ID.
    - Tools: Firebase Auth plugin.

    - Roles: Backend dev.

- **Basic UI Screens:** Build initial UI: onboarding screen (with consent checkbox), home/dashboard (show basic stats or last survey), survey screens (questions and submit button), settings screen (consent again, about).
    - Tools: Flutter widgets, theming.
    - Roles: UI dev.
- **Initial ML Module Integration:** Develop a simple ML workflow: use collected (or simulated) data to train a model offline (e.g. in Python) for stress prediction. Decide deployment:
    - If cloud-based: write a Firebase Cloud Function or HTTP endpoint to run inference on new data.
    - If on-device: convert model to TensorFlow Lite and integrate with Flutter (using tflite_flutter or mlkit).
    - Tools: Python (scikit-learn/TensorFlow), Firebase Cloud Functions (Node.js/Python), TensorFlow Lite.
    - Roles: ML dev (lead), Backend dev (assist).
- **Consent & Privacy Implementation:** Implement the consent flow in-app: at onboarding, present privacy policy and require opt-in before any data is sent. Ensure all data writes respect user consent (e.g. flag not to collect if revoked). Use HTTPS (built-in).
    - Tools: Firebase rules, secure storage.
    - Roles: Backend dev.
- **Continuous Integration (Optional):** Set up basic CI for building the Flutter app (e.g. GitHub Actions) to catch build errors.
    - Tools: GitHub Actions, Flutter test suite.
    - Roles: UI dev.
- **Integration Testing:** Test each component: verify sensor data is captured on device, survey notifications appear on schedule, and data correctly appears in Firestore.
    - Tools: Firebase Emulator Suite (for offline testing), physical devices.
    - Roles: All interns.
- **Internal Demo:** Present the working prototype (app collecting data and sending to backend) to mentor, demonstrating sensor logging and EMA flow. Incorporate early feedback.

Phase 4: Testing, Evaluation, and Optimization (Month 3)

- **Functional Testing:** Develop and run tests: unit tests for individual modules, integration tests for data flow. Verify accuracy of sensor readings (e.g. move device and check accelerometer output) and survey reliability.
    - Tools: Flutter flutter_test, emulators, unit test frameworks.
    - Roles: UI dev, Backend dev.
- **Pilot Data Collection:** Recruit a small group of test users (classmates or volunteers). Deploy the app (internal beta) and collect real data for 1–2 weeks. Monitor compliance rates (survey completion).
    - Tools: Feedback forms, monitoring dashboard.
    - Roles: All interns.
- **Data Preprocessing:** Gather the pilot data from Firestore. Clean the data: remove duplicates, fill or remove missing values, anonymize IDs. Organize sensor time-series and corresponding EMA labels.
    - Tools: Python (Pandas, NumPy).
    - Roles: ML dev (lead).
- **Feature Engineering:** Extract meaningful features from raw data: e.g. daily step count, average acceleration variance, sleep duration proxy (e.g. phone inactivity at night), screen time, survey scores, etc. Document feature set.
    - Tools: Python libraries, possibly Jupyter notebooks.
    - Roles: ML dev.
- **Model Training & Validation:** Train ML models to predict stress levels (binary or continuous) using the engineered features and EMA labels. Try algorithms like Random Forest, SVM, or a small neural net. Use cross-validation to assess performance.
    - Tools: scikit-learn, TensorFlow.
    - Roles: ML dev (lead).
- **Model Evaluation:** Compute metrics (accuracy, F1-score, ROC AUC). Analyze feature importance to interpret which behaviors correlate with stress. Document results and any data limitations.
    - Tools: Python (sklearn.metrics, visualization).
    - Roles: ML dev.
- **Model Optimization:** Based on evaluation, refine features or model parameters. Iterate training (e.g. remove noisy features, balance data). Finalize the chosen model.
    - Tools: Python.
    - Roles: ML dev.
- **Deployment of ML Model:** Deploy the final model:

    - If cloud: update the Cloud Function with trained model; test the endpoint on new inputs.
    - If on-device: convert to TFLite and integrate into the Flutter app; test inference accuracy on device.
    - Tools: Firebase Functions, TFLite converter, Flutter plugin.
    - Roles: ML dev, Backend dev.
- **App & UI Refinement:** Incorporate user feedback from pilot: fix usability issues, add requested features (e.g. a progress chart showing stress trend). Improve visuals (better charts, themes).
    - Tools: Flutter, design resources.
    - Roles: UI dev.
- **Performance & Battery Optimization:** Ensure background data collection is efficient: adjust sensor sampling rates, use batches, limit wakeups. Test battery impact and optimize (e.g. using work manager or appropriate lifecycle hooks).
    - Tools: Android Profiler, iOS Instruments.
    - Roles: UI dev.
- **Privacy/Consent Audit:** Re-verify all data flows against privacy plan. Check that no personal data is stored inadvertently. Ensure data deletion option works.
    - Tools: Manual review, Firebase logs.
    - Roles: All interns.
- **Mentor Review 4:** Present test results, improved app demo, and final ML performance metrics to mentor. Prepare for final adjustments.

Phase 5: Deployment, Final Documentation, and Publication (Month 3–4)

- **Finalize App for Release:** Polish remaining bugs, finalize app version. Prepare release build certificates (Android keystore, iOS provisioning).
    - Tools: Flutter build tools, platform-specific app store consoles.
    - Roles: UI dev.
- **App Deployment:** Publish the app prototype to Google Play Store (testing track) and Apple TestFlight (if iOS). Provide access instructions to pilot users.
    - Tools: Google Play Console, App Store Connect.
    - Roles: UI dev.
- **System Documentation:** Write comprehensive documentation:
    - Code documentation (README, code comments).
    - Architecture/design docs.

    - User manual (how app works, privacy policy).
    - Tools: Markdown, GitHub Wiki.
    - Roles: All interns contribute.
- **Research Paper Preparation:** Draft paper for conference submission. Include:
    - **Introduction/Lit Review:** Summarize relevant workfrontiersin.orgfrontiersin.org.
    - **Methods:** Describe app design, sensors/EMA integration, data collection procedure.
    - **Results:** Report data analysis and model performance (tables, figures).
    - **Discussion:** Interpret findings, limitations, future work.
    - Tools: Overleaf/LaTeX or MS Word, reference manager.
    - Roles: ML dev (lead writing), all interns (contribute sections and proofreading).
- **Figures and Visuals:** Create high-quality figures: app screenshots, data flow diagrams, graphs of results (e.g. feature correlations, ROC curves).
    - Tools: matplotlib/Excel, Canva.
    - Roles: ML dev (data figures), UI dev (screenshots).
- **Conference Submission:** Identify a suitable venue (e.g. CHI, IEEE EMBC, mHealth workshop). Format paper to guidelines and submit by deadline.
    - Tools: Conference websites, submission portal.
    - Roles: All interns.
- **Presentation/Poster:** Prepare a presentation or poster summarizing the project (motivation, methods, results).
    - Tools: PowerPoint, LaTeX Beamer, or poster templates.
    - Roles: UI dev (design), ML dev (content).
- **Final Privacy & Compliance Check:** Confirm final project meets ethical standards. Ensure all consent documentation is recorded. Delete any leftover test data.
    - Tools: Checklists, review of collected data.
    - Roles: All interns.
- **Final Mentor Review & Handoff:** Demonstrate final app and present paper outline to mentor. Ensure all deliverables (working prototype and draft paper) meet requirements. Organize code repository and share project documentation.

Sources: Project tasks and design are informed by research on digital phenotyping and mobile sensingfrontiersin.orgfrontiersin.org. Ethical considerations are guided by frameworks for mental health appspmc.ncbi.nlm.nih.gov. System architecture follows best practices for sensor data platformsjmir.org. EMA implementation is based on smartphone survey methodologiespmc.ncbi.nlm.nih.govpmc.ncbi.nlm.nih.gov.

Citations

Frontiers | Passive sensing data predicts stress in university students: a supervised machine learning method for digital phenotyping

https://www.frontiersin.org/journals/psychiatry/articles/10.3389/fpsyt.2024.1422027/full

Frontiers | Passive sensing data predicts stress in university students: a supervised machine learning method for digital phenotyping

https://www.frontiersin.org/journals/psychiatry/articles/10.3389/fpsyt.2024.1422027/full

Ecological Momentary Assessment Using Smartphone-Based Mobile Application for Affect and Stress Assessment - PMC

https://pmc.ncbi.nlm.nih.gov/articles/PMC6230530/

Ecological Momentary Assessment Using Smartphone-Based Mobile Application for Affect and Stress Assessment - PMC

https://pmc.ncbi.nlm.nih.gov/articles/PMC6230530/

Ethical Development of Digital Phenotyping Tools for Mental Health Applications: Delphi Study - PMC

https://pmc.ncbi.nlm.nih.gov/articles/PMC8367187/

Untitled

https://www.jmir.org/2019/11/e16399/PDF

Ecological Momentary Assessment Using Smartphone-Based Mobile Application for Affect and Stress Assessment - PMC

https://pmc.ncbi.nlm.nih.gov/articles/PMC6230530/

All Sources

 frontiersin

pmc.ncbi.nlm.nih

 jmir