Revolutionizing Maternal Care: Integrating Fetal Kick Counters with EMR Alerts

A response to "Connecting with fetus: The use of app-based fetal movement counting and experiences during pregnancy and birth" (2025)

Introduction

As physicians who code, we stand at a unique intersection of clinical practice and technological innovation. The recent study published in Midwifery (2025) on app-based fetal movement counting offers compelling insights into how digital tools can enhance maternal-fetal monitoring while improving the pregnancy experience. However, the study also reveals a critical gap: despite the benefits of regular fetal kick counting, compliance remains low, and there's minimal integration with clinical workflows.

This presents an opportunity to leverage our dual expertise as clinicians and developers to create more effective solutions. In this post, I'll outline how we can build upon existing fetal kick counter applications to create a comprehensive system that not only improves patient engagement but also integrates seamlessly with Electronic Medical Record (EMR) systems to alert physicians about concerning patterns.

The Evidence for App-Based Fetal Kick Counting

The 2025 study examined data from the Count the Kicks (CTK) app, analyzing both usage patterns and end-of-pregnancy surveys from 1,147 participants. The findings are particularly relevant to our work as physician-developers:

1. **Positive Outcomes**: Frequent use of fetal movement counting was associated with lower maternal anxiety (78.2% reported feeling less anxious), increased maternal-fetal bonding (72.3% reported improved bonding), and ultimately healthier birth outcomes.

2. **Compliance Challenges**: Despite these benefits, compliance with daily counting was surprisingly low. Most users counted fetal movements less than 21 days in each month of the third trimester, with usage particularly low in the first and third months.

3. **Clinical Disconnect**: While the app provided valuable data, there was no direct connection to clinical care systems, creating a siloed approach to monitoring.

These findings suggest that while app-based fetal kick counting has significant potential, current implementations are missing critical features that could improve both compliance and clinical utility.

The Missing Link: EMR Integration and Clinical Alerts

As physicians who code, we recognize that the true power of health applications comes from their ability to bridge the gap between patient self-monitoring and clinical decision-making. Here's how we can enhance fetal kick counter applications with EMR integration and intelligent alert systems:

1. Real-Time Data Synchronization with EMR

The foundation of an improved system would be bidirectional data flow between the patient's fetal kick counter app and the clinical EMR system:

```
// Example code for EMR integration using FHIR API
async function syncKickCountData(patientId, sessionData) {
  const fhirObservation = {
    resourceType: "Observation",
    status: "final",
    code: {
      coding: [{
        system: "http://loinc.org",
        code: "72052-5",
        display: "Fetal movement count"
      }]
    },
    subject: {
      reference: `Patient/${patientId}`
    },
    effectiveDateTime: sessionData.timestamp,
    valueQuantity: {
      value: sessionData.kickCount,
      unit: "count",
      system: "http://unitsofmeasure.org",
      code: "{count}"
    },
    component: [
      {
        code: {
          coding: [{
            system: "http://loinc.org",
            code: "8555-1",
            display: "Duration"
          }]
        },
        valueQuantity: {
          value: sessionData.duration,
          unit: "min",
          system: "http://unitsofmeasure.org",
          code: "min"
        }
      }
    ]
  };

  return await fhirClient.create(fhirObservation);
}
```

This integration would ensure that kick count data becomes part of the patient's official medical record, accessible to all care providers.

2. Intelligent Alert System for Clinicians

The core innovation would be an intelligent alert system that analyzes kick count patterns and notifies clinicians of potential concerns:

```
// Example algorithm for detecting concerning patterns
function analyzeKickCountPattern(patientData) {
  const recentSessions = patientData.sessions.slice(-10);

  // Check for significant increase in time to complete 10 movements
  const avgTimeLastThree = calculateAverage(recentSessions.slice(-3).map(s => s.duration));
  const avgTimePrevious = calculateAverage(recentSessions.slice(-10, -3).map(s => s.duration));

  if (avgTimeLastThree > avgTimePrevious * 1.5) {
    return {
      alertLevel: "warning",
      message: "Significant increase in time to complete kick count detected",
      recommendation: "Consider clinical evaluation"
    };
  }

  // Check for missed days
  const missedDays = calculateConsecutiveMissedDays(patientData.sessions);
  if (missedDays >= 3) {
    return {
      alertLevel: "info",
      message: `Patient has missed ${missedDays} consecutive days of kick counting`,
      recommendation: "Consider patient outreach"
    };
  }

  // Check for sudden decrease in movement
  // Additional pattern detection logic

  return { alertLevel: "normal" };
}
```

3. Advanced Alert Prioritization Based on Clinical Risk Factors

To minimize alert fatigue and ensure the most critical cases receive prompt attention, we can implement a sophisticated prioritization system that factors in clinical risk data from the EMR:

```
// Advanced alert prioritization based on clinical risk factors
function calculateAlertPriority(patientData, kickCountAlert) {
  let priorityScore = kickCountAlert.baseScore;

  // Factor in clinical risk factors from EMR
  if (patientData.clinicalRiskFactors.includes("hypertension")) {
    priorityScore += 2;
  }

  if (patientData.clinicalRiskFactors.includes("gestational_diabetes")) {
    priorityScore += 2;
  }

  if (patientData.clinicalRiskFactors.includes("previous_stillbirth")) {
    priorityScore += 3;
  }

  // Factor in gestational age
  if (patientData.gestationalAge > 40) {
    priorityScore += 2;
  } else if (patientData.gestationalAge > 37) {
    priorityScore += 1;
  }

  // Determine final alert level
  if (priorityScore >= 7) {
    return "critical";
  } else if (priorityScore >= 4) {
    return "warning";
  } else {
    return "info";
  }
}
```

4. Customizable Alert Thresholds for Different Practice Settings

Different clinical settings may require different alert thresholds. A high-risk maternal-fetal medicine practice might want earlier notifications than a general OB/GYN office:

```
// Practice-level configuration for alert thresholds
const practiceAlertSettings = {
  highRiskSpecialistPractice: {
    timeIncreaseThreshold: 1.3, // 30% increase triggers alert
    missedDaysThreshold: 2,     // 2 consecutive missed days
    movementDecreaseThreshold: 0.7 // 30% decrease in movement rate
  },
  generalObGynPractice: {
    timeIncreaseThreshold: 1.5, // 50% increase triggers alert
    missedDaysThreshold: 3,     // 3 consecutive missed days
    movementDecreaseThreshold: 0.6 // 40% decrease in movement rate
  },
  midwiferyPractice: {
    timeIncreaseThreshold: 1.4, // 40% increase triggers alert
    missedDaysThreshold: 2,     // 2 consecutive missed days
    movementDecreaseThreshold: 0.65 // 35% decrease in movement rate
  }
};
```

5. EMR-Integrated Dashboard for OB/GYNs

A specialized dashboard within the EMR would provide clinicians with an at-a-glance view of their patients' fetal movement patterns:

- Color-coded patient list highlighting concerning patterns
- Trend visualization showing kick count patterns over time
- One-click access to detailed kick count history
- Integration with other relevant clinical data (ultrasound results, maternal vitals, etc.)
- Ability to document clinical decisions based on kick count data

6. Automated Clinical Decision Support

The system can go beyond simple alerts to provide evidence-based clinical recommendations:

```
// Clinical decision support recommendations based on alert patterns
function generateClinicalRecommendations(patientData, alertHistory) {
  const recentAlerts = alertHistory.filter(a =>
    (new Date() - new Date(a.timestamp)) < 7 * 24 * 60 * 60 * 1000 // Last 7 days
  );

  if (recentAlerts.some(a => a.level === "critical")) {
    return {
      recommendationType: "urgent_evaluation",
      suggestedActions: [
        "Schedule same-day assessment",
        "Perform NST (Non-Stress Test)",
        "Consider BPP (Biophysical Profile)",
        "Evaluate for delivery if ≥ 39 weeks gestation"
      ],
      emrOrderSets: ["NST_ORDER_SET", "BPP_ORDER_SET"]
    };
  } else if (recentAlerts.filter(a => a.level === "warning").length >= 2) {
    return {
      recommendationType: "prompt_evaluation",
      suggestedActions: [
        "Schedule evaluation within 24-48 hours",
        "Perform NST",
        "Review recent prenatal history"
      ],
      emrOrderSets: ["NST_ORDER_SET"]
    };
  }

  // Additional recommendation logic...

  return {
    recommendationType: "routine_care",
    suggestedActions: [
      "Continue routine prenatal care",
      "Reinforce importance of daily kick counting"
    ],
    emrOrderSets: []
  };
}
```

Improving Patient Compliance Through Better Design

The study highlighted poor compliance as a major limitation. As physician-developers, we can address this through thoughtful application design:

1. Personalized Reminder System

```
 // Adaptive reminder system based on user behavior
function generateOptimalReminderTime(userProfile) {
  // Analyze historical usage patterns
  const preferredTimeRanges =
analyzePreferredCountingTimes(userProfile.sessions);

  // Consider user-specified preferences
  const userPreferences = userProfile.reminderPreferences || {};

  // Factor in compliance history to adjust frequency
  const complianceRate = calculateComplianceRate(userProfile.sessions, 14); // Last
14 days

  // Generate personalized reminder schedule
  return createReminderSchedule(preferredTimeRanges, userPreferences,
complianceRate);
}
```

2. Gamification and Positive Reinforcement

- Achievement badges for consistent counting
- Visual progress tracking through the pregnancy journey
- Optional social sharing features for milestone celebrations
- Personalized encouragement messages based on compliance patterns

3. Educational Content Integration

- Just-in-time education about the importance of kick counting
- Visualization of how kick counting correlates with fetal well-being
- Testimonials from other mothers about how kick counting impacted their pregnancy

Population Health Management Integration

Beyond individual patient care, the aggregated kick count data can provide valuable population health insights:

```
 // Aggregate kick count data for population health insights
function generatePopulationHealthMetrics(practiceId, timeframe = "last30days") {
  return db.transaction(async (tx) => {
    // Get all patients with kick count data in the specified timeframe
    const patients = await tx.patients.getWithKickCountData(practiceId, timeframe);
    // Calculate compliance metrics
    const complianceRates = patients.map(p => calculateComplianceRate(p.kickCountSessions));
    const avgComplianceRate = average(complianceRates);

    // Identify high-risk patients with concerning patterns
    const highRiskPatients = patients.filter(p =>
      hasRecentAlert(p.alerts, "critical") ||
      hasMultipleAlerts(p.alerts, "warning", 2)
    );

    // Generate provider-specific metrics and recommendations
    // ...

    return {
      overallMetrics: {
        totalPatientsMonitored: patients.length,
        averageComplianceRate: avgComplianceRate,
        highRiskPatientCount: highRiskPatients.length,
        interventionRate: interventionRate
      },
      highRiskPatients: highRiskPatients.map(p => ({
        id: p.id,
        name: p.name,
        gestationalAge: p.gestationalAge,
        lastAlertDate: getLastAlertDate(p.alerts),
        lastAlertLevel: getLastAlertLevel(p.alerts)
      })),
      // Additional metrics and recommendations...
    };
  });
}
```

Implementation Strategy for Clinical Practice

For physician-developers looking to implement such a system, I recommend the following approach:

1. **Start with a Pilot Program**: Begin with a small cohort of patients to test the integrated system, focusing on high-risk pregnancies where the benefits may be most immediate.

2. **Measure Key Metrics**:

3. Patient compliance rates
4. Clinician response to alerts
5. False positive/negative rates for concerning patterns
6. Patient and provider satisfaction

7. Clinical outcomes and interventions triggered by the system

8. **Iterative Refinement**: Use feedback from both patients and clinicians to continuously improve the system, particularly the alert algorithms to minimize alert fatigue.

9. **Address Privacy and Security**: Ensure robust data protection measures that comply with HIPAA and other relevant regulations.

10. **Quality Improvement Tracking**:

```
// Track alert outcomes for quality improvement
function trackAlertOutcome(alertId, outcome) {
  return db.transaction(async (tx) => {
    const alert = await tx.alerts.getById(alertId);

    // Record the clinical outcome
    await tx.alertOutcomes.create({
      alertId: alertId,
      patientId: alert.patientId,
      providerId: alert.providerId,
      outcome: outcome.result, // e.g., "true_positive", "false_positive"
      clinicalAction: outcome.action, // e.g., "nst_performed", "additional_monitoring"
      timeToAction: outcome.timeToAction, // minutes from alert to clinical action
      notes: outcome.notes
    });

    // Update alert quality metrics for algorithm improvement
    // ...

    return { success: true };
  });
}
```

Conclusion

The 2025 study on app-based fetal kick counting provides valuable evidence that digital tools can positively impact the pregnancy experience. As physicians who code, we have the opportunity to take these findings further by creating integrated systems that not only improve the patient experience but also enhance clinical care through EMR integration and intelligent alerts.

By bridging the gap between patient self-monitoring and clinical workflows, we can create a more connected maternal care ecosystem that improves outcomes while reducing anxiety. The technology to accomplish this exists today—we simply need to apply our unique perspective as clinicians who understand both the medical necessity and the technical implementation.

I invite fellow members of the Doctors Who Code community to share their experiences with similar integration projects or to contribute ideas for enhancing the proposed system. Together, we can leverage our dual expertise to create solutions that truly advance maternal-fetal care.

Dr. [Your Name]
Maternal-Fetal Medicine Specialist & Healthcare Developer
Doctors Who Code Blog Contributor