Zakat Calculator Implementation Guide

Islamic Jurisprudence & Technical Specifications

Table of Contents

Islamic Foundation

Quranic Basis

Primary Verses:

- *"And establish prayer and give Zakat, and whatever good you put forward for yourselves - you will find it with Allah."* (Quran 2:110)
- *"Take from their wealth a charity to purify them and sanctify them with it."* (Quran 9:103)

Hadith References

Authentic Sources:

- *"Islam is built on five pillars... giving Zakat"* (Sahih Bukhari 8)
- *"No Zakat is due on property mounting to less than five Awsuq"* (Sahih Bukhari 1459)

Zakat Categories

1. Cash & Bank Savings (Zakat on Wealth)

Islamic Ruling: Obligatory on cash, savings, and liquid assets **Nisab:** 85 grams of gold OR 595 grams of silver (whichever is lower) **Rate:** 2.5% annually **Conditions:**

- Must be owned for complete lunar year (Hawl)
- Must exceed Nisab threshold
- Must be surplus to basic needs

Calculation:

```
Zakatable Amount = Total Cash + Bank Savings + Investments - Debts
If Zakatable Amount ≥ Nisab:
    Zakat Due = Zakatable Amount × 0.025
```

2. Gold & Silver

Islamic Ruling: Based on authentic Hadith regarding precious metals **Gold Nisab:** 85 grams (approximately 20 Mithqal) **Silver Nisab:** 595 grams (approximately 200 Dirhams) **Rate:** 2.5% of total weight value

Calculation:

```
Gold Value = Weight (grams) × Current Gold Price per gram
Silver Value = Weight (grams) × Current Silver Price per gram
Total Precious Metals = Gold Value + Silver Value
If Total ≥ Nisab: Zakat = Total × 0.025
```

3. Business Inventory & Trade Goods

Islamic Ruling: Zakat applicable on trade merchandise **Nisab:** Same as cash (85g gold or 595g silver equivalent) **Rate:** 2.5% **Includes:**

- Raw materials
- Work in progress
- Finished goods for sale
- Business investments

4. Livestock (Zakat al-An'am)

Camels:

- 5-9 camels: 1 sheep
- 10-14 camels: 2 sheep
- 15-19 camels: 3 sheep
- 20-24 camels: 4 sheep
- 25-35 camels: 1 young female camel

Cattle:

- 30-39 cattle: 1 young calf
- 40-59 cattle: 1 young female calf

- 60-69 cattle: 2 young calves

Sheep & Goats:

- 40-120: 1 sheep
- 121-200: 2 sheep
- 201-399: 3 sheep
- 400+: 1 sheep per 100

5. Agricultural Produce (Zakat al-Zuru')

Nisab: 5 Awsuq (approximately 653 kg or 1,440 lbs) **Rate:**

- Rain-fed crops: 10% (Ushr)
- Irrigated crops: 5% (Nisf al-Ushr) **Applicable to:** Grains, fruits, vegetables sold commercially

---

Technical Implementation

Core Architecture

```
ZakatCalculator/
├── models/
│   ├── ZakatCategory.kt
│   ├── NisabThreshold.kt
│   └── ZakatCalculation.kt
├── calculators/
│   ├── CashZakatCalculator.kt
│   ├── GoldSilverCalculator.kt
│   ├── BusinessCalculator.kt
│   ├── LivestockCalculator.kt
│   └── AgricultureCalculator.kt
├── services/
│   ├── NisabService.kt
│   ├── CurrencyService.kt
│   └── HijriDateService.kt
└── validators/
    └── IslamicValidationService.kt
```

Data Models

ZakatCategory

kotlin

```kotlin
enum class ZakatCategory {
    CASH_SAVINGS,
    GOLD_SILVER,
    BUSINESS_TRADE,
    LIVESTOCK_CAMELS,
    LIVESTOCK_CATTLE,
    LIVESTOCK_SHEEP_GOATS,
    AGRICULTURE_GRAINS,
    AGRICULTURE_FRUITS
}
```

NisabThreshold

```kotlin
data class NisabThreshold(
    val goldGrams: Double = 85.0,
    val silverGrams: Double = 595.0,
    val livestockMinimum: Map<String, Int> = mapOf(
        "camels" to 5,
        "cattle" to 30,
        "sheep_goats" to 40
    ),
    val agricultureKg: Double = 653.0
)
```

Currency & Market Data Integration

```kotlin
interface MarketDataService {
    suspend fun getGoldPricePerGram(currency: String): Double
    suspend fun getSilverPricePerGram(currency: String): Double
    suspend fun getExchangeRate(fromCurrency: String, toCurrency: String): Double
}
```

Calculation Logic

Primary Calculation Engine

```kotlin
```

```
class ZakatCalculationEngine {

    fun calculateCashZakat(
        cashAmount: Double,
        currency: String,
        nisabReference: NisabReference // GOLD or SILVER
    ): ZakatResult {
        val nisabValue = when(nisabReference) {
            GOLD -> getNisabValueInGold(currency)
            SILVER -> getNisabValueInSilver(currency)
        }

        return if (cashAmount >= nisabValue) {
            ZakatResult(
                zakatable = true,
                zakatDue = cashAmount * 0.025,
                nisabExceeded = cashAmount - nisabValue
            )
        } else {
            ZakatResult(zakatable = false, zakatDue = 0.0)
        }
    }

    fun calculateGoldSilverZakat(
        goldWeight: Double,
        silverWeight: Double,
        goldPrice: Double,
        silverPrice: Double
    ): ZakatResult {
        val goldValue = goldWeight * goldPrice
        val silverValue = silverWeight * silverPrice
        val totalValue = goldValue + silverValue

        val meetsNisab = goldWeight >= 85.0 || silverWeight >= 595.0 ||
                totalValue >= minOf(85.0 * goldPrice, 595.0 * silverPrice)

        return if (meetsNisab) {
            ZakatResult(
                zakatable = true,
                zakatDue = totalValue * 0.025
            )
        } else {
            ZakatResult(zakatable = false, zakatDue = 0.0)
        }
```

```
    }
}
```

Hijri Calendar Integration

```kotlin
class HijriDateService {
    fun hasCompletedHawl(acquisitionDate: HijriDate): Boolean {
        val currentHijriDate = getCurrentHijriDate()
        return currentHijriDate.year > acquisitionDate.year ||
               (currentHijriDate.year == acquisitionDate.year + 1 &&
                currentHijriDate.dayOfYear >= acquisitionDate.dayOfYear)
    }
}
```

User Interface Guidelines

Input Validation

- **Currency Formatting:** Support local currency formats
- **Weight Units:** Allow kg/grams for gold/silver, appropriate units for livestock
- **Date Picker:** Hijri and Gregorian calendar support
- **Real-time Calculation:** Update Zakat amounts as user inputs data

Educational Components

```
For each category, include:
- Brief Islamic ruling explanation
- Relevant Quranic verse or Hadith reference
- Practical examples
- Nisab threshold explanation in local currency
```

Accessibility Features

- **Multi-language Support:** Arabic, English, Urdu, Bahasa, etc.
- **Right-to-Left (RTL)** support for Arabic languages
- **Voice Input:** For easier data entry
- **Large Text Options:** For elderly users

Verification & Testing

Islamic Compliance Testing

1. **Mufti Review Panel:** Engage certified Islamic scholars
2. **Test Cases:** Create scenarios covering edge cases
3. **Comparative Analysis:** Validate against established Zakat institutions
4. **Regional Variations:** Account for different madhab interpretations

Technical Testing

kotlin

```kotlin
@Test
fun testCashZakatCalculation() {
    val calculator = ZakatCalculationEngine()
    val result = calculator.calculateCashZakat(
        cashAmount = 10000.0,
        currency = "USD",
        nisabReference = GOLD
    )

    assertTrue(result.zakatable)
    assertEquals(250.0, result.zakatDue, 0.01)
}

@Test
fun testNisabThreshold() {
    val goldNisab = 85.0 * getCurrentGoldPrice()
    val silverNisab = 595.0 * getCurrentSilverPrice()
    val lowerNisab = minOf(goldNisab, silverNisab)

    val result = calculator.calculateCashZakat(lowerNisab - 1.0, "USD", GOLD)
    assertFalse(result.zakatable)

    val result2 = calculator.calculateCashZakat(lowerNisab + 1.0, "USD", GOLD)
    assertTrue(result2.zakatable)
}
```

Compliance Requirements

Scholarly Verification

- **Fatwa Committee Approval:** Obtain formal approval from recognized Islamic authorities
- **Documentation:** Maintain detailed documentation of all Islamic rulings applied
- **Updates:** Regular review process for any changes in Islamic interpretations

Data Security & Privacy

- **Encryption:** All financial data must be encrypted at rest and in transit
- **Privacy Policy:** Clear statement about data usage in compliance with Islamic ethics
- **Local Storage:** Option to calculate without cloud storage for privacy concerns

Regional Considerations

- **Local Madhab:** Allow users to select their school of thought
- **Currency Support:** Support for all major currencies and regional variations
- **Cultural Sensitivity:** Appropriate terminology and cultural considerations

Regular Updates

- **Market Data:** Real-time precious metal prices and exchange rates
- **Nisab Values:** Updated nisab thresholds based on current market conditions
- **Islamic Calendar:** Accurate Hijri calendar integration

Implementation Checklist

Phase 1: Core Development

- [ ] Basic Zakat categories implementation
- [ ] Nisab calculation engine
- [ ] Currency and market data integration
- [ ] Basic UI with input forms

Phase 2: Advanced Features

- [ ] Hijri calendar integration
- [ ] Multi-language support
- [ ] Educational content integration
- [ ] Historical calculation tracking

Phase 3: Verification & Launch

- [ ] Islamic scholar review and approval
- [ ] Comprehensive testing suite
- [ ] Beta testing with Muslim community
- [ ] Final compliance verification

Phase 4: Maintenance

- [ ] Regular market data updates
- [ ] Ongoing scholarly consultation

- [ ] User feedback integration
- [ ] Performance monitoring

Recommended Islamic Authorities for Verification

1. **International Islamic Fiqh Academy (OIC)**
2. **European Council for Fatwa and Research**
3. **Islamic Society of North America (ISNA) Fiqh Committee**
4. **Dar al-Ifta (Egypt, Saudi Arabia, etc.)**
5. **Local respected Islamic institutions and scholars**

Disclaimer

This implementation guide should be reviewed and approved by qualified Islamic scholars before deployment. The calculations must align with authentic Islamic teachings and may require adjustments based on different schools of Islamic jurisprudence (madhab).

"And Allah knows best" - والله أعلم