Neighborhood Watch Application – Complete Document

# 1) Users and Roles

1.1 Three user types:  
- Administrator  
- Security Officer  
- Neighborhood Watch Member

1.2 Primary platforms:  
- Administrator: Web (desktop)  
- Security Officer: Mobile app (Android/iOS)  
- Member: Mobile app (primary), optional web portal

1.3 Unique privileges:  
- Administrator: User management, route setup, payments, reports, system configuration  
- Security Officer: Patrol management, checkpoint scans, incidents, SOS responses  
- Member: SOS alerts, tips, community feed, subscriptions

# 2) Authentication & Registration

2.1 Admin account creation & recovery:  
- Created via invite, recovery with email + TOTP

2.2 Security officer registration & approval:  
- Apply/invite, ID verification, admin approval

2.3 Member registration & approval:  
- Self-signup with OTP, join neighborhood zone, payment verification

2.4 Admin 2FA:  
- TOTP required for login and sensitive actions

# 3) Features by Role

3.1 Administrator actions:  
- Manage users, patrol routes, checkpoints, shifts, subscriptions, reports, notifications

3.2 Security officer (mobile):  
- Start patrol, GPS logging, scan checkpoints, file incidents, SOS response

3.3 Member features:  
- SOS alerts, tips with media, community feed, polls, payments, household profile

3.4 Non-payment for 2 months:  
- Day 1 reminder, Day 7 restrictions, Day 30 warning, Day 60 suspension

# 4) Patrol & Monitoring Rules

4.1 Logging patrols:  
- Start shift, GPS breadcrumbs, checkpoint scans, end shift summary

4.2 Details captured at each scan:  
- Officer ID, checkpoint ID, GPS, timestamp, notes, photos, battery, network state

4.3 Low checkpoint coverage:  
- Prompt officer, supervisor notification, flagged as non-compliant

4.4 Automatic anomaly notifications:  
- Missed scans, route deviation, SOS SLA breaches, tampering alerts

# 5) Alerts & Community Features

5.1 Raising emergency alerts:  
- SOS button, lock-screen widget, shake-to-SOS, wearables, USSD fallback

5.2 When an emergency alert is raised:  
- Officer dispatch notification, live incident room, contacts alerted

5.3 Community interactions:  
- Announcements, polls, moderated tips, badges, missing pet board

# 6) Reporting

6.1 Report types:  
- Patrol compliance, incident response, SOS volumes, heatmaps, revenue, churn

6.2 Access:  
- Admin: all  
- Supervisors: zones  
- Officers: personal summary  
- Members: personal & community stats

# 7) Security, Scalability & Performance

7.1 Why secure:  
- Sensitive data, lives at risk

7.2 Payments security:  
- PCI-DSS provider, tokenization, encrypted storage, audit logs

7.3 Scalability plan:  
- Modular architecture, containerization, PostgreSQL with replicas, Redis, CDN

7.4 Disaster recovery:  
- Automated backups, multi-AZ, RPO 15min, RTO 60min

7.5 Performance requirements:  
- API <300ms, incident create <2s, push dispatch <5s

# 8) Mobile App Offline Functionality

8.1 Offline capabilities:  
- Patrol logging, checkpoint scans, incidents queued, cached feed, SOS fallback

8.2 Sync strategy:  
- Local SQLite, background sync, conflict resolution, resumable uploads

# 9) Suspension & Deletion Rules

9.1 Member suspension:  
- Non-payment 60 days, abuse, false reports, TOS breach

9.2 Officer suspension:  
- Non-compliance, tampering, misconduct

9.3 Reinstating officer:  
- Supervisor review, remedial training, probation, reinstatement by admin

9.4 Permanent deletion (officer):  
- Account disabled, tokens revoked, data anonymized, history retained

# 10) Open Questions & Technology

Mobile app tech:  
- Flutter recommended for cross-platform and offline-first

Backend stack:  
- Node.js/NestJS or Django/FastAPI, PostgreSQL, Redis, message queues, Keycloak for auth, push notification services

Notifications:  
- Push (FCM/APNs), SMS, email, webhook retries, template engine

Payments (Botswana):  
- Mobile money + card payments through PCI-compliant aggregator