SXM GO - Development Team Assignment

Version: 1.0

Last Updated: January 2026

Purpose: Divide development workload across 4 developers for parallel execution

---

Developer Assignment Structure

Developer 1: Core Authentication & User Management

Primary Responsibilities:

- User authentication system (email, social logins)
- User profile management & cloud storage
- Account settings & preferences
- Privacy controls & data management
- Session management & security

Key Requirements:

- FR-001 to FR-011 (Authentication & Profiles)
- TR-022 to TR-026 (Security)
- UX-004 (Permission handling)
- QA-011 to QA-013 (Security testing)

Deliverables:

- Authentication API endpoints
- User profile database schema
- Social OAuth integration (Google, Apple, Facebook)
- Data encryption implementation
- Account deletion functionality
- GDPR/CCPA compliance features

Dependencies:

- Backend infrastructure setup (shared)
- API framework selection (shared decision)

Estimated Effort: 3-4 weeks

Developer 2: Map, Location & Check-In System

Primary Responsibilities:

- Interactive map implementation
- Location database & pin management
- GPS-based check-in verification
- Anti-fraud measures
- Offline map functionality
- Location categories & filtering

Key Requirements:

- FR-012 to FR-026 (Interactive Map)
- FR-027 to FR-035 (Check-In System)
- TR-014 (Maps API integration)
- TR-027 to TR-029 (Battery/GPS optimization)
- QA-004 (GPS accuracy testing)

Deliverables:

- Map UI component with Google Maps/Mapbox integration
- Location data model & API
- Check-in verification logic (50-100m radius)
- GPS spoofing detection
- Location detail views (name, description, hours, photos)
- Offline caching system
- Category filtering system
- QR code verification (optional)

Dependencies:

- User authentication (Dev 1)
- Points system API (Dev 3)

Estimated Effort: 4-5 weeks

Developer 3: Gamification Engine (Points, Challenges, Badges)

Primary Responsibilities:

- Points calculation & allocation system
- Challenge creation & tracking
- Badge/achievement system
- Activity tracking (Strava-style)
- Route recording & visualization
- Progress tracking logic

Key Requirements:

- FR-036 to FR-043 (Points System)
- FR-059 to FR-071 (Challenges & Badges)
- FR-084 to FR-096 (Activity Tracking)
- DA-003 (Analytics for popular activities)

Deliverables:

- Points calculation engine (10-50 per check-in, bonuses, multipliers)
- Challenge tracking system (Foodie, High Roller, Island Nomad, Sun Chaser)
- Badge unlock logic (Bronze/Silver/Gold tiers)
- Activity recording service (GPS route tracking)
- Route visualization on map
- Activity statistics (distance, duration, pace, elevation)
- Achievement notification system
- Social media sharing for activities

Dependencies:

- Check-in events (Dev 2)
- User profiles (Dev 1)
- Leaderboard data sync (Dev 4)

Estimated Effort: 4-5 weeks

Developer 4: Social Features & Leaderboards

Primary Responsibilities:

- Leaderboard system (global, weekly, monthly, group)
- Itinerary creation & sharing
- Social features (friends, groups)
- Real-time rankings
- Notification system
- Social media sharing

Key Requirements:

- FR-044 to FR-058 (Itinerary System)
- FR-072 to FR-083 (Leaderboard System)
- FR-097 to FR-103 (Social Features)
- UX-008 to UX-010 (Notifications)
- TR-010 (Real-time database)
- TR-016 (Push notifications)

Deliverables:

- Leaderboard API & UI (global, weekly, monthly, group)
- Real-time ranking updates
- Itinerary builder (3-20 locations)
- Itinerary sharing (unique links, QR codes)
- Pre-made featured itineraries
- Group management system (create, invite, join)
- Friend system (search, requests, profiles)
- Push notification service
- Social media share functionality

Dependencies:

- User profiles (Dev 1)
- Points data (Dev 3)
- Location data (Dev 2)

Estimated Effort: 4-5 weeks

Shared Responsibilities (All Developers)

Infrastructure Setup (Week 1-2)

- Backend platform selection (Supabase/Firebase/AWS)
- API architecture design (REST/GraphQL)
- Database schema design
- Development environment setup
- CI/CD pipeline configuration
- Analytics integration (Mixpanel/Amplitude/Firebase)
- CDN setup for static assets

Testing Phase (Weeks 11-12)

- Integration testing across modules
- End-to-end testing of critical flows
- Performance testing (1,000+ concurrent users)
- Beta deployment and feedback collection
- Bug fixes and optimization

Documentation

- API documentation (endpoints, payloads, responses)
- Code comments and inline documentation
- Integration guides for handoffs
- Deployment procedures

Integration Points & Communication Protocol

Critical Handoffs

1. Dev 1 → All Developers:

- User authentication tokens (format, expiration)

- User ID structure and format
- Profile data structure (JSON schema)
- Authentication middleware/libraries

2. Dev 2 → Dev 3:

- Check-in event payload structure
- Location visit data format
- GPS coordinate standards

3. Dev 3 → Dev 4:

- Points earned event notifications
- Achievement unlocked event format
- Activity completion data

4. Dev 2 → Dev 4:

- Location metadata for itineraries
- Location search/filter API
- Location images and descriptions

API Contract Management

- Use shared API specification document (OpenAPI/Swagger)
- Version all API endpoints
- Document breaking changes with 1-week notice
- Mock endpoints for parallel development

Weekly Sync Meeting Agenda

- API contract reviews
- Dependency blockers
- Integration testing results
- Performance benchmarks
- Next week's priorities

Communication Channels

- **Slack/Discord:** Daily quick updates

- **GitHub:** Code reviews, issues, PRs
- **Jira/Linear:** Task tracking
- **Confluence/Notion:** Technical documentation

MVP Development Timeline

Phase 0: Planning & Setup (Week 1-2)

All Developers:

- Finalize tech stack decisions
- Set up development environments
- Create database schemas
- Define API contracts
- Initialize repositories

Phase 1: Foundation (Week 3-6)

Dev 1:

- ✅ Basic email/password authentication
- ✅ User profile CRUD operations
- ✅ Session management

Dev 2:

- ✅ Map rendering with 50+ locations
- ✅ Location pin display
- ✅ GPS location tracking

Dev 3:

- ✅ Points calculation logic
- ✅ Basic challenge definitions
- ✅ Badge data models

Dev 4:

- ✅ Global leaderboard (top 100)

- ✅ Basic itinerary data model
- ✅ Database for rankings

Phase 2: Integration (Week 7-9)

All Developers:

- Check-in → Points → Leaderboard pipeline
- Profile displays points & rank
- 4 basic challenges operational (Foodie, High Roller, Island Nomad, Sun Chaser)
- Activity tracking working
- Itinerary creation functional

Integration Focus:

- End-to-end check-in flow
- Real-time leaderboard updates
- Challenge progress tracking
- Social OAuth working

Phase 3: Polish & Testing (Week 10)

All Developers:

- UI/UX refinement
- Performance optimization
- Bug fixes from integration testing
- Edge case handling
- Error messaging improvement

Phase 4: Beta Testing (Week 11-12)

All Developers:

- Deploy to TestFlight/Play Store Beta
- Monitor crash reports
- Gather user feedback
- Performance monitoring
- Final bug fixes

Phase 5: Launch Preparation (Week 13-14)

All Developers:

- App store assets (screenshots, descriptions)
- Final QA pass
- Production deployment
- Marketing materials support
- Launch day monitoring

Feature Priority Matrix

Must-Have for MVP Launch

Feature	Owner	Status
User authentication	Dev 1	Required
Interactive map	Dev 2	Required
GPS check-in	Dev 2	Required
Points system	Dev 3	Required
Global leaderboard	Dev 4	Required
4 basic badges	Dev 3	Required
iOS app	All	Required

Should-Have for MVP

Feature	Owner	Priority
Social OAuth	Dev 1	High
Offline maps	Dev 2	High
Activity tracking	Dev 3	Medium
Weekly leaderboard	Dev 4	Medium

Feature	Owner	Priority
Push notifications	Dev 4	Medium
Android app	All	High

Nice-to-Have (Post-MVP)

Feature	Owner	Timeline
Custom itineraries	Dev 4	Phase 2
Group competitions	Dev 4	Phase 2
Advanced anti-fraud	Dev 2	Phase 2
Route visualization	Dev 3	Phase 2
Friend system	Dev 4	Phase 3
QR verification	Dev 2	Phase 3

Technology Stack Recommendations

Mobile Framework

Recommended: React Native

- Single codebase for iOS & Android
- Excellent map library ecosystem
- Faster development time
- Large developer community

Alternative: Flutter (if team has Dart experience)

Backend

Option 1: Supabase (Recommended for MVP)

- Built-in authentication
- Real-time database
- PostgreSQL with REST API

- File storage included
- Generous free tier

Option 2: Firebase

- Easy real-time updates
- Strong mobile SDK
- Good for MVP speed

Option 3: AWS

- Most scalable
- Higher complexity
- Better for post-MVP scaling

Mapping

Google Maps Platform

- Most comprehensive
- Familiar to users
- Higher cost

Mapbox

- More affordable
- Customizable styling
- Good offline support

Real-time Rankings

- Supabase Realtime (if using Supabase)
- Firebase Realtime Database
- Socket.io with Node.js

Risk Mitigation by Developer

Dev 1 Risks

Risk	Mitigation
OAuth integration delays	Start with email/password, add OAuth in Week 2
Data privacy compliance	Use established libraries, legal review early

Dev 2 Risks

Risk	Mitigation
GPS accuracy in urban areas	Implement 50-100m configurable radius
Battery drain	Use adaptive GPS precision, geofencing
API costs (Google Maps)	Implement aggressive caching, consider Mapbox

Dev 3 Risks

Risk	Mitigation
Complex points calculation	Create comprehensive test suite early
GPS spoofing	Pattern detection algorithms, manual review queue

Dev 4 Risks

Risk	Mitigation
Real-time performance	Implement efficient database indexing, caching layer
Notification spam	User preference controls, rate limiting

Code Repository Structure

```
sxm-go/
```

```
├── mobile/          # React Native app
│   ├── src/
│   │   ├── auth/        # Dev 1: Authentication
│   │   ├── map/         # Dev 2: Map & Locations
│   │   ├── gamification/  # Dev 3: Points & Challenges
│   │   ├── social/      # Dev 4: Leaderboards & Social
│   │   └── shared/      # Shared components
│   └── tests/
├── backend/         # API & Database
│   ├── auth/        # Dev 1
│   ├── locations/   # Dev 2
│   ├── gamification/  # Dev 3
│   └── social/      # Dev 4
└── docs/            # Documentation
```

Success Metrics by Developer

Dev 1: Authentication

- < 3 second login time
- 0 authentication-related crashes
- 100% GDPR compliance

Dev 2: Map & Check-ins

- < 2 second map load time
- < 1 second check-in verification
- < 1% false positive check-ins
- GPS accuracy within 25 meters

Dev 3: Gamification

- Points calculated within 500ms of check-in
- 0 duplicate point awards
- Activity tracking battery impact < 10%/hour

Dev 4: Social

- Leaderboard updates within 2 seconds

- < 100ms response for top 100 rankings
- Notification delivery > 95% success rate

Post-Launch Support Division

Week 1-2 Post-Launch (All hands on deck)

- Monitor crash reports
- Fix critical bugs
- Performance optimization
- User feedback triage

Ongoing Maintenance (Rotating weekly)

- **Week 1:** Dev 1 on-call
- **Week 2:** Dev 2 on-call
- **Week 3:** Dev 3 on-call
- **Week 4:** Dev 4 on-call

Contact & Escalation

Project Lead: [Name]
Technical Decisions: Requires 3/4 developer consensus
Blocker Escalation: Slack #sxm-go-blockers channel
Daily Standup: 10 AM (15 minutes max)

Document Control

Version	Date	Author	Changes
1.0	January 2026	Development Team	Initial team assignment

Next Review: End of Week 2 (after infrastructure setup)