Great White Shark Sighting Tracker

Project Timeline & Milestones

Project Overview

- **Project Name**: Great White Shark Sighting Tracker
- **Location Focus**: Wallace, California coast
- **Platforms**:
  - iOS App (Swift)
  - Desktop App (Java)

Timeline Overview

Phase	Description	Duration	Cumulative Time
1	Planning & Design	2 weeks	2 weeks
2	iOS & API MVP	4 weeks	6 weeks
3	Desktop MVP	6 weeks	12 weeks
4	LLM Summaries	8 weeks	20 weeks
5	Geofence Alerts	10 weeks	30 weeks
6	QA & Launch	12 weeks	42 weeks

Note: Timeline recalibrated on April 16, 2025 to ensure precise phase boundaries and milestone synchronization.

Detailed Timeline & Key Milestones

Phase 1: Planning & Design (Weeks 1-2)

- **Week 1**
  - [ ] Complete UI/UX wireframes for iOS app
  - [ ] Complete UI/UX wireframes for desktop app
  - [ ] Define data schema for PostgreSQL with PostGIS extension
  - [ ] Draft API specification document
- **Week 2**
  - [ ] Finalize architecture diagram
  - [ ] Set up development environments

- [ ] Configure Magic by Modular for project management
- [ ] Establish GitHub repository and CI/CD pipelines

MILESTONE: Architecture & Design Documentation Approved

Phase 2: iOS & API MVP (Weeks 3-6)

- **Week 3**
  - [ ] Set up Mojo development environment
  - [ ] Implement basic API endpoints for shark sighting submission
  - [ ] Create iOS project structure in Xcode
- **Week 4**
  - [ ] Implement MapKit integration in iOS app
  - [ ] Develop sighting submission form UI
  - [ ] Set up PostgreSQL database with PostGIS extension
- **Week 5**
  - [ ] Implement camera/photo library integration in iOS
  - [ ] Connect iOS app to API endpoints
  - [ ] Develop basic map visualization of sightings
- **Week 6**
  - [ ] Implement basic filtering functionality in iOS app
  - [ ] Complete API MVP endpoints for fetching sightings
  - [ ] Internal testing of iOS app MVP

  MILESTONE: iOS & API MVP Functional

Phase 3: Desktop MVP (Weeks 7-12)

- **Week 7**
  - [ ] Set up Java environment with JavaFX
  - [ ] Create basic application structure
  - [ ] Design dashboard layout
- **Week 8-9**
  - [ ] Implement map integration via Java webview (Mapbox/Leaflet)
  - [ ] Create sighting submission dialog with drag-and-drop
  - [ ] Connect to API endpoints
- **Week 10**
  - [ ] Develop TableView with search, sort, and filter functionality
  - [ ] Implement user authentication and profile management

- **Week 11-12**
  - [ ] Create basic reporting dashboard
  - [ ] Integrate desktop app with API
  - [ ] Internal testing of desktop app MVP
  - [ ] Refine UI/UX based on feedback

  MILESTONE: Desktop Application MVP Complete (July 11, 2025)

Phase 4: LLM Summaries (Weeks 13-20)

- **Week 13-14**
  - [ ] Research and select appropriate LLM provider
  - [ ] Set up Java SDK credentials for LLM access
  - [ ] Design prompts for generating sighting summaries
- **Week 15-16**
  - [ ] Implement LLM service integration in backend API
  - [ ] Develop daily summary generation functionality
  - [ ] Create UI components for displaying summaries in iOS app
- **Week 17-18**
  - [ ] Implement summary display in desktop application
  - [ ] Develop scheduled summary generation system
  - [ ] Add user preferences for summary customization
- **Week 19-20**
  - [ ] Refine prompts based on test results
  - [ ] Implement caching and optimization for LLM requests
  - [ ] Conduct internal testing of summary features

  MILESTONE: LLM Summary Features Implemented (September 6, 2025)

Phase 5: Geofence Alerts (Weeks 21-30)

- **Week 21-22**
  - [ ] Design geofence configuration UI for iOS
  - [ ] Implement Core Location geofencing on iOS
  - [ ] Develop push notification system for iOS
- **Week 23-24**
  - [ ] Create geofence configuration UI for desktop
  - [ ] Implement background location monitoring in desktop app
  - [ ] Develop system tray notification system

- **Week 25-27**
  - [ ] Create geofence management endpoints in API
  - [ ] Implement user preference storage for alert settings
  - [ ] Develop alerting logic and rules engine
- **Week 28-30**
  - [ ] Integrate alert system with sighting submission workflow
  - [ ] Test geofence accuracy and notification delivery
  - [ ] Optimize for battery usage on mobile

  MILESTONE: Geofence Alert System Functional (November 16, 2025)

Phase 6: QA & Launch (Weeks 31-42)

- **Week 31-32**
  - [ ] Conduct comprehensive internal testing
  - [ ] Fix identified bugs and issues
  - [ ] Perform security audit and penetration testing
- **Week 33-35**
  - [ ] Recruit beta testers in Wallace, CA
  - [ ] Distribute beta versions of iOS and desktop apps
  - [ ] Collect and analyze feedback
- **Week 36-38**
  - [ ] Address beta feedback
  - [ ] Implement final refinements
  - [ ] Prepare App Store submission materials
- **Week 39-42**
  - [ ] Submit iOS app to App Store for review
  - [ ] Prepare desktop app distribution channels
  - [ ] Finalize documentation and support materials

  MILESTONE: Public Launch (February 9, 2026)

Key Performance Indicators

- **User Adoption**: Number of active users in Wallace, CA
- **Sighting Reports**: Total number of submitted sightings
- **Alert Effectiveness**: Time between sighting submission and alert delivery
- **Summary Quality**: User satisfaction with LLM-generated insights

- **Cross-Platform Engagement**: Percentage of users utilizing both iOS and desktop apps

---

Risk Assessment & Mitigation

Risk	Impact	Likelihood	Mitigation Strategy
LLM provider cost increases	High	Medium	Implement caching strategy, set usage limits
API performance issues with high load	High	Medium	Implement rate limiting, load balancing
iOS App Store rejection	High	Low	Follow Apple guidelines strictly, prepare for multiple review cycles
Inaccurate location data	Medium	Medium	Implement verification system, user rating of report accuracy
Low user adoption	High	Medium	Engage local community early, partner with Wallace conservation groups

---

Resource Allocation

Resource	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6
iOS Developer	50%	100%	20%	50%	70%	50%
Java Developer	50%	20%	100%	50%	70%	50%
Backend Engineer	70%	80%	80%	80%	60%	40%
UX/UI Designer	100%	50%	50%	30%	50%	20%
QA Tester	20%	40%	40%	40%	50%	100%

Next Steps

1. Schedule kickoff meeting
2. Assign Phase 1 tasks to team members
3. Set up project tracking in your preferred tool
4. Provision development environments
5. Begin wireframing and design work