

## **Flight Data APIs:**

- **Primary:** Amadeus Self-Service (2,000 free requests/month)
- **Secondary:** Skyscanner Partner API (free with affiliate approval)
- **Tertiary:** FlightStats Flex APIs (comprehensive flight tracking)
- **Fallback:** Manual flight entry with basic tracking

## **Weather Data APIs:**

- **Primary:** OpenWeatherMap One Call API 3.0 (1,000 free calls/day)
- **Secondary:** Open-Meteo API (completely free, no API key required)
- **Tertiary:** WeatherAPI.com (free tier available)
- **Fallback:** Static weather information without real-time updates

## **Currency Exchange APIs:**

- **Primary:** Fixer.io (100 free requests/month)
- **Secondary:** ExchangeRate-API (1,500 free requests/month)
- **Tertiary:** CurrencyAPI (300 free requests/month)
- **Fallback:** Cached exchange rates updated weekly

## **Hotel Search APIs:**

- **Primary:** Hotels.com API via RapidAPI
- **Secondary:** Booking.com API (requires partnership)
- **Tertiary:** Expedia Affiliate Network (EAN)
- **Fallback:** Manual hotel information entry

## **Location/Geocoding APIs:**

- **Primary:** OpenWeatherMap Geocoding API (free with weather API key)
- **Secondary:** MapBox Geocoding API (100,000 free requests/month)
- **Tertiary:** Google Maps Geocoding API (paid but highly accurate)
- **Fallback:** Core Location framework for basic location services

## **Enhanced Travel Data APIs:**

Based on the research, several additional APIs can provide premium features:

1. **TriplIt API:** Email parsing for automatic trip creation from booking confirmations
2. **Rome2Rio API:** Multi-modal transportation route planning
3. **Sygic Travel API:** Comprehensive destination guides and points of interest

4. **AirVisual API:** Air quality data for health-conscious travelers

5. **REST Countries API:** Country information, visa requirements, and travel advisories

### Smart Caching and Offline Strategies:

```
swift

// Intelligent API usage with caching
class APIManager {
    private let cache = NSCache<NSString, AnyObject>()

    func getCachedWeatherOrFetch(lat: Double, lon: Double) async throws -> WeatherData {
        let cacheKey = "weather_\u{0024}(lat)\u{0024}(lon)" as NSString

        // Check cache first (weather data valid for 10 minutes)
        if let cached = cache.object(forKey: cacheKey) as? CachedWeatherData,
            Date().timeIntervalSince(cached.timestamp) < 600 {
            return cached.data
        }

        // Try primary API first, fallback to secondary
        do {
            let weather = try await OpenWeatherService().getCurrentWeather(lat: lat, lon: lon)
            cache.setObject(CachedWeatherData(data: weather, timestamp: Date()), forKey: cacheKey)
            return weather
        } catch {
            // Fallback to Open-Meteo (free, no API key)
            let fallbackWeather = try await OpenMeteoService().getWeatherForecast(lat: lat, lon: lon)
            return fallbackWeather.convertToWeatherData()
        }
    }
}
```

### Framework Alternatives:

- **Database:** Core Data → SQLite.swift → Realm (if Core Data proves insufficient)
- **Networking:** URLSession → Alamofire (if advanced networking features needed)
- **Authentication:** Firebase Auth → AWS Cognito → Custom OAuth implementation

### Key Library Components:

- **Core Data Stack:** NSPersistentContainer, NSManagedObjectContext for data management
- **Networking Layer:** URLSession with Combine framework for reactive API calls
- **Image Processing:** Vision framework's VNDocumentCameraViewController for document scanning

- **Calendar Integration:** EventKit's EKEventStore for calendar access and .ics generation
- **QR Generation:** Core Image's CIFilter for QR code creation

This comprehensive technology stack provides multiple fallback options while leveraging proven iOS frameworks. The implementation prioritizes free-tier APIs and native iOS capabilities to ensure project feasibility within academic constraints.

---

## Scope and Estimated Project Timeline

### Minimum Viable Product (MVP) Definition

The MVP for TripSync focuses on core trip organization, document management, and sharing functionality that directly addresses user pain points identified in market research. The MVP will demonstrate all required technologies while providing genuine value to users through the features shown in Screens 1-11.

#### MVP Core Features:

1. User authentication and account management (Screen 1)
2. Trip creation and basic information management (Screens 2-3)
3. Flight integration with real-time status updates (Screen 9)
4. Document upload and organization (PDF/images) (Screens 5-6, 10)
5. QR code trip sharing with web interface (Screens 7-8)
6. Basic budget planning with currency conversion (Screen 13)
7. Destination discovery with cost estimates (Screen 8)
8. Offline access to essential trip information (All screens)

## Detailed Development Timeline

### Week 8 (Labs 8): Foundation and Authentication

*Milestone:* Basic app structure with user authentication

#### Specific Tasks:

- Initialize iOS project with required frameworks (UIKit, Firebase, Core Data)
- Configure Firebase project (Authentication, Firestore, Storage)
- Implement Screen 1: Authentication interface with email/social login
- Set up Core Data model for Trip, User, and Document entities
- Create basic tab bar navigation structure matching Screen 2 layout
- Configure app icons, launch screen, and basic styling

- Test authentication flow on device

*Learning Requirements:* Firebase iOS SDK integration, Core Data setup *Deliverable:* Users can register, sign in, and navigate between empty tab screens

## **Week 9 (Labs 9): Trip Management Core**

*Milestone:* Trip creation and display functionality

*Specific Tasks:*

- Implement Screen 2: My Trips hub with enhanced trip cards
- Create Screen 3: Trip creation modal with form validation
- Implement Core Data CRUD operations for Trip entity
- Set up Amadeus API integration for Screen 9: Flight search
- Build flight selection interface with search results
- Add trip editing capabilities and basic timeline display (Screen 4)
- Implement pull-to-refresh data synchronization

*Learning Requirements:* API integration patterns, form validation, table view management *Deliverable:* Users can create, view, and edit trips with basic flight information

## **Week 10 (Labs 10): Document Management System**

*Milestone:* Document upload and organization

*Specific Tasks:*

- Implement Screen 6: Document upload modal with camera integration
- Build Screen 5: Documents tab with category organization
- Create Screen 10: Document viewer with PDF/image display capabilities
- Set up Firebase Storage for document management
- Implement document categorization system (Flights, Hotels, Activities)
- Add document deletion and management features
- Configure offline document caching for airplane mode access

*Learning Requirements:* AVFoundation camera APIs, PDFKit integration, file management *Deliverable:* Users can upload, view, and organize travel documents within trips

## **Week 11 (Labs 11): Advanced Features Integration**

*Milestone:* Budget planning, currency conversion, and sharing

*Specific Tasks:*

- Implement Screen 13: Budget planning with Fixer.io API integration
- Build currency conversion service with real-time rates
- Create Screen 7: QR code generation modal using Core Image
- Develop Screen 8: Responsive web interface for trip sharing
- Implement deep linking back to iOS app
- Add budget estimation system with destination cost data
- Test sharing across different devices and platforms

*Learning Requirements:* Currency APIs, QR code generation, web development *Deliverable:* Users can plan budgets, convert currencies, and share trips via QR codes

## **Week 12 (Labs 12): Calendar Integration and Notifications**

*Milestone:* Calendar export and notification system

*Specific Tasks:*

- Implement Screen 14: Calendar export options with EventKit integration
- Build .ics file generation system for cross-platform compatibility
- Create Screen 15: Notification settings with granular controls
- Set up UserNotifications framework for trip reminders
- Implement one-week advance notification scheduling
- Add background notification processing
- Test calendar integration with iOS Calendar and Google Calendar

*Learning Requirements:* EventKit framework, notification scheduling, calendar file formats *Deliverable:* Users can export trips to calendar and receive customized notifications

## **Enhanced Features (Post-MVP)**

If development progresses faster than expected, these features from Screens 12 and 17 can be added:

### **Enhanced User Profile Management (Screens 12, 17):**

- Advanced settings and global preferences
- Travel statistics and achievements
- iCloud sync and offline mode toggles
- Account security and privacy controls
- Default currency and notification preferences

### **Advanced Destination Discovery (Screen 8):**

- Favorites management with heart icon interactions
- Cost estimation integration with budget planning
- User ratings and reviews system
- Location-based destination suggestions

## Project Scope Constraints

### Technical Limitations:

- Free-tier API limitations (Amadeus: 2,000 requests/month, Fixer.io: 100 requests/month)
- Firebase free tier constraints for storage and database operations
- iOS simulator testing limitations for camera and notification features
- Academic timeline constraints (5-week development window)

### Feature Prioritization:

- Core trip management takes priority over advanced social features
- Offline functionality essential for travel use case
- Budget planning more important than complex analytics
- QR sharing prioritized over complex collaboration features

### Risk Mitigation:

- API rate limiting managed through intelligent caching strategies
- Offline-first design ensures core functionality without network dependency
- Multiple fallback options for each critical technology component
- MVP scope designed to be achievable within timeline constraints

## Success Metrics

### MVP Completion Criteria:

- User can create account and authenticate successfully
- Trip creation and editing workflow fully functional
- Document upload and viewing works offline
- Flight integration displays real-time status
- QR sharing generates functional web links
- Budget calculations display accurate currency conversion
- Calendar export produces valid .ics files
- All 17 wireframe screens implemented with core functionality

## **Technical Requirements Met:**

- Persistent storage via Core Data with CloudKit sync
- Web services integration (Amadeus + Fixer.io + Firebase)
- Maps integration for destination visualization
- Local notifications for trip reminders
- Camera integration for document scanning
- Calendar integration via EventKit

This timeline ensures TripSync delivers a comprehensive travel companion app that addresses real user needs while demonstrating mastery of iOS development frameworks and best practices. The phased approach allows for iterative testing and refinement while maintaining realistic expectations for academic project completion.

---

## **References**

- Amadeus Developers. (2024). *Flight Offers Search API Documentation*. Retrieved from <https://developers.amadeus.com/self-service/category/air/api-doc/flight-offers-search>
- Apple Inc. (2024). *Human Interface Guidelines*. Retrieved from <https://developer.apple.com/design/human-interface-guidelines/>
- Apple Inc. (2024). *iOS App Programming Guide*. Retrieved from <https://developer.apple.com/library/archive/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html>
- Apple Inc. (2024). *Core Data Programming Guide*. Retrieved from <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CoreData/index.html>
- Apple Inc. (2024). *EventKit Framework Reference*. Retrieved from <https://developer.apple.com/documentation/eventkit>
- Applause. (2024). *Travel App User Experience Report: Crisis Management in Mobile Applications*. App Quality Report, 15(3), 24-31.
- Cleveroad. (2025). *Best Travel APIs: Types, Providers, and Tips for Selection*. Retrieved from <https://www.cleveroad.com/blog/travel-apis/>
- Datarade. (2023). *Best 6 Travel APIs to use in 2023*. Retrieved from <https://datarade.ai/top-lists/best-travel-apis>
- Firebase. (2024). *iOS Firebase SDK Documentation*. Retrieved from <https://firebase.google.com/docs/ios/setup>

Fixer.io. (2024). *Currency Conversion API Documentation*. Retrieved from  
<https://fixer.io/documentation>

FlightsLogic. (2019). *Free Travel API / Travel Booking API - Flight API*. Retrieved from  
<https://www.flightslogic.com/free-travel-api.php>

HackerNoon. (2022). *My Best Travel APIs*. Retrieved from <https://hackernoon.com/my-best-travel-apis>

Hotels.com. (2024). *Hotels.com API Documentation*. Retrieved from  
<https://rapidapi.com/apidojo/api/hotels-com-provider>

Open-Meteo. (2024). *Free Open-Source Weather API*. Retrieved from <https://open-meteo.com/>

OpenWeatherMap. (2024). *Weather API Documentation*. Retrieved from  
<https://openweathermap.org/api>

OpenWeatherMap. (2024). *One Call API 3.0 Documentation*. Retrieved from  
<https://openweathermap.org/api/one-call-3>

OpenWeatherMap. (2024). *Geocoding API Documentation*. Retrieved from  
<https://openweathermap.org/api/geocoding-api>

Rome2Rio. (2024). *Multi-modal Transportation API*. Retrieved from <https://www.rome2rio.com/api>

Skyscanner. (2024). *Travel APIs for Partners*. Retrieved from <https://partners.skyscanner.net/>

Sygic Travel. (2024). *Travel API Documentation*. Retrieved from <https://travel.sygic.com/api>

Travelopro. (2019). *Free Travel API, Travel Booking API*. Retrieved from  
<https://www.travelopro.com/free-travel-api.php>

Tripl. (2024). *App Store Listing and User Reviews*. Retrieved from <https://apps.apple.com/app/tripit-travel-planner/id311035142>

Tripl. (2024). *Tripl API Documentation*. Retrieved from <https://tripit.github.io/api/doc/v1/>

Wanderlog. (2024). *Travel Planning App Features and Trustpilot Reviews*. Retrieved from  
<https://wanderlog.com> and <https://www.trustpilot.com/review/wanderlog.com>

WeatherAPI.com. (2024). *Free Weather API Documentation*. Retrieved from  
<https://www.weatherapi.com/>