# Revised Quest Feature Integration Index for Master Technical Blueprint

## Overview

This document provides a comprehensive indexing of the refined Quest system for integration into the Setarcos app master technical blueprint. The integration incorporates the skill tree design, enhanced Concepts integration, accessibility improvements, and cognitive load reductions while maintaining alignment with the existing blueprint structure and utilizing the established XP system.

## Integration Index

### 1. Core Definitions and Architecture

#### 1.1 Definitions and Glossary (Section 1.4)

* Add “Skill Tree” definition for quest navigation structure
* Add “Challenge Node” definition for special quest steps with bonus rewards

#### 1.2 System Architecture (Section 3.2)

* Update Section 3.2.5: “Quest Service” with new skill tree management capabilities
  + Skill tree rendering service
  + Quest progression tracking service
  + Quest recommendation engine

### 2. Feature Integration

#### 2.1 Quest Core Feature (Section 5.2)

* Update feature overview to include skill tree design
* Replace existing user experience flow with refined flow:
  1. Quest browsing and selection
  2. Skill tree visualization
  3. Step completion with Concepts integration
  4. Progress tracking with standard XP system
* Update technical implementation with new components
* Add accessibility considerations

#### 2.2 Concepts Integration (Section 5.9)

* Add quest-concept linking mechanisms
* Implement concept mastery progression through quest completion
* Create mini-tree visualization for related concepts
* Add seamless navigation between quests and concepts

#### 2.3 XP System Integration (Section 6.2)

* Update XP rewards structure for quest steps and completion
* Implement milestone badges for quest completion
* Use existing XP system for unlocking advanced quests or branches
* Leverage existing XP visualization for quest progression

#### 2.4 Accessibility Integration (Section 6.5)

* Add high-contrast mode for skill tree visualization
* Implement screen reader support for quest navigation
* Add voice input for reflections and quiz responses
* Create audio narration for quest content

### 3. Technical Implementation

#### 3.1 Database Schema Updates (Section 7.2)

-- Update existing quests table  
ALTER TABLE quests  
ADD COLUMN skill\_tree\_structure JSONB DEFAULT '{}',  
ADD COLUMN xp\_threshold INTEGER DEFAULT 0,  
ADD COLUMN accessibility\_options JSONB DEFAULT '{}';  
  
-- Update existing quest\_steps table  
ALTER TABLE quest\_steps  
ADD COLUMN node\_position JSONB DEFAULT '{}',  
ADD COLUMN connected\_nodes JSONB DEFAULT '[]',  
ADD COLUMN concept\_links JSONB DEFAULT '[]',  
ADD COLUMN is\_challenge\_node BOOLEAN DEFAULT FALSE,  
ADD COLUMN bonus\_xp INTEGER DEFAULT 0;  
  
-- New quest\_badges table  
CREATE TABLE quest\_badges (  
 id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),  
 name VARCHAR(100) NOT NULL,  
 description TEXT NOT NULL,  
 xp\_threshold INTEGER NOT NULL,  
 quest\_category VARCHAR(50) NOT NULL,  
 icon\_url VARCHAR(255) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
-- New user\_quest\_badges table  
CREATE TABLE user\_quest\_badges (  
 id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),  
 user\_id UUID NOT NULL REFERENCES users(id),  
 badge\_id UUID NOT NULL REFERENCES quest\_badges(id),  
 earned\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 UNIQUE(user\_id, badge\_id)  
);

#### 3.2 API Endpoints (Section 8.3)

* Update existing endpoints:

GET /api/v1/quests - Add skill\_tree parameter for tree view  
GET /api/v1/quests/{id} - Include skill tree structure and node positions  
POST /api/v1/quests/{id}/start - Track XP initialization  
POST /api/v1/quests/{id}/steps/{step\_id}/complete - Award XP

* Add new endpoints:

GET /api/v1/quests/skill-tree - Get full skill tree visualization  
GET /api/v1/quests/recommended - Get personalized quest recommendations  
POST /api/v1/quests/{id}/concepts/{concept\_id}/explore - Track concept exploration from quest

#### 3.3 Frontend Components (Section 6.1)

* Add new components:
  + SkillTreeView.tsx - Zoomable skill tree visualization
  + QuestProgressBar.tsx - Progress indicator using standard XP visualization
  + QuestNodeComponent.tsx - Individual node in skill tree
  + ConceptMiniTree.tsx - Mini visualization of related concepts
  + AccessibilityControls.tsx - Controls for accessibility options

#### 3.4 Celery Tasks (Section 10.3)

* Update existing XP tasks:

@celery\_app.task(name="xp.award\_quest\_xp")  
def award\_quest\_xp(user\_id, quest\_id, step\_id=None, is\_completion=False):  
 """Award XP for quest activities using the standard XP system"""  
 # Implementation details  
  
@celery\_app.task(name="quests.check\_badge\_eligibility")  
def check\_badge\_eligibility(user\_id, quest\_category):  
 """Check if user is eligible for new badges based on XP thresholds"""  
 # Implementation details  
  
@celery\_app.task(name="quests.update\_concept\_mastery\_from\_quest")  
def update\_concept\_mastery\_from\_quest(user\_id, quest\_id, step\_id):  
 """Update concept mastery based on quest step completion"""  
 # Implementation details

### 4. User Experience

#### 4.1 Navigation Structure (Section 6.3)

* Update with bottom tab bar navigation pattern
* Add modal overlay for QuestStep component
* Implement seamless navigation between Quests and Concepts

#### 4.2 Accessibility Features (Section 6.5)

* Add high-contrast mode for skill tree
* Implement screen reader compatibility
* Add voice input and navigation
* Create adjustable text sizes
* Implement audio narration

### 5. Analytics and Monitoring

#### 5.1 Analytics Integration (Section 11.2)

* Add quest progression metrics
* Track concept exploration from quests
* Monitor skill tree navigation patterns
* Measure accessibility feature usage

### 6. Implementation Plan

#### 6.1 Phase 1: Core Infrastructure

* Database schema updates
* Basic API endpoint updates
* XP integration for quest progression

#### 6.2 Phase 2: Frontend Components

* Skill tree visualization
* Quest step refinements
* XP visualization for quest progression

#### 6.3 Phase 3: Concepts Integration

* Concept linking in quest nodes
* Mini-tree visualization
* Mastery progression tracking

#### 6.4 Phase 4: Accessibility and Optimization

* Accessibility features implementation
* Performance optimization for skill tree
* Voice input and navigation

## Technical Considerations

1. **Performance Optimization**
   * Implement progressive loading for large skill trees
   * Use WebGL for smooth animations and transitions
   * Cache quest content for offline access
2. **Accessibility Compliance**
   * Ensure WCAG 2.1 AA compliance for all new components
   * Test with screen readers and voice navigation
   * Implement keyboard navigation for all interactive elements
3. **Mobile Optimization**
   * Ensure touch-friendly targets (minimum 44x44 pixels)
   * Optimize skill tree for various screen sizes
   * Implement gesture controls for zooming and navigation
4. **Integration Testing**
   * Test seamless navigation between Quests and Concepts
   * Verify XP accumulation and badge awards
   * Validate concept mastery progression from quest completion

## XP System Integration Details

### XP for Quest Activities

* Starting quests: +5 XP
* Completing quest steps: +10-30 XP per step (based on difficulty)
* Completing full quests: +25-100 XP (based on difficulty)
* Challenge nodes: Additional bonus XP (+5-15)

### Quest Progression via XP

* Use standard XP thresholds to unlock advanced quests or branches
* Example: 500 XP required to unlock “Advanced Stoic Practices”
* Standard XP bar visualization shows progress toward unlocks
* Milestone badges awarded at specific XP thresholds (e.g., “Stoic Novice” at 100 XP)

### Technical Implementation

* Leverage existing XPService for all XP calculations and awards
* Use existing XP database tables for tracking
* Extend existing XP visualization components for quest-specific progress