**ReqVista - Estimates and Timeline**

**1. Effort Estimation**

**1.1 Estimation Approach**

The estimation for the ReqVista project is based on the following approach:

* **T-Shirt Sizing**: Initial high-level estimates using T-shirt sizing (S, M, L, XL)
* **Story Points**: Refined estimates using story points for backlog items
* **Function Point Analysis**: Detailed estimation for complex components
* **Historical Data**: Leveraging previous similar project metrics
* **Expert Judgment**: Input from experienced team members
* **Three-Point Estimation**: Using optimistic, most likely, and pessimistic scenarios

**1.2 Estimation Factors**

The following factors were considered in the estimation process:

* Complexity of requirements
* Technical challenges and unfamiliar technologies
* Integration points with external systems
* Cross-platform development needs
* Team experience and familiarity with technologies
* Testing requirements and quality expectations
* Non-functional requirements (performance, security, etc.)

**1.3 Effort Summary by Workstream**

| **Workstream** | **Estimated Effort (Person-Days)** | **Confidence Level** |
| --- | --- | --- |
| Analysis and Planning | 45-60 | High |
| Design | 80-100 | Medium |
| Database and API Development | 120-150 | High |
| Web Application Development | 150-180 | Medium |
| Mobile Application Development | 180-220 | Medium |
| Shared Component Development | 60-80 | High |
| Integration Development | 40-60 | Medium |
| Testing | 140-170 | Medium |
| DevOps and Infrastructure | 60-80 | Medium |
| Documentation | 50-70 | High |
| Deployment and Rollout | 30-40 | Medium |
| Project Management | 90-110 | High |
| **TOTAL** | **1045-1320** | **Medium** |

**1.4 Effort Breakdown by Role**

| **Role** | **Estimated Effort (Person-Days)** |
| --- | --- |
| Project Manager | 90-110 |
| Solution Architect/Technical Lead | 80-100 |
| Backend Developers | 180-220 |
| Web Developers | 150-180 |
| Mobile Developers | 180-220 |
| Shared Component Developers | 60-80 |
| QA Engineers | 140-170 |
| DevOps Engineers | 60-80 |
| UX/UI Designers | 45-60 |
| Database Specialists | 40-50 |
| Documentation Specialists | 20-30 |
| **TOTAL** | **1045-1320** |

**1.5 Key Assumptions in Estimation**

* Team members are fully dedicated to the project
* Required environments and tools are available when needed
* Requirements as defined in the ER diagram remain stable
* Stakeholder feedback is received in a timely manner
* No significant scope changes occur during development
* Team has the necessary experience with .NET, Blazor, and MAUI

**2. Project Timeline**

**2.1 High-Level Timeline**

| **Phase** | **Duration** | **Start Date** | **End Date** |
| --- | --- | --- | --- |
| Inception and Planning | 4 weeks | Week 1 | Week 4 |
| Analysis and Design | 8 weeks | Week 5 | Week 12 |
| Development - Iteration 1 | 6 weeks | Week 13 | Week 18 |
| Development - Iteration 2 | 6 weeks | Week 19 | Week 24 |
| Development - Iteration 3 | 6 weeks | Week 25 | Week 30 |
| System Testing | 4 weeks | Week 31 | Week 34 |
| User Acceptance Testing | 3 weeks | Week 35 | Week 37 |
| Deployment Preparation | 2 weeks | Week 38 | Week 39 |
| Go-Live and Stabilization | 3 weeks | Week 40 | Week 42 |
| **TOTAL** | **42 weeks** | **Week 1** | **Week 42** |

**2.2 Detailed Timeline by Milestone**

**2.2.1 Inception and Planning (Week 1-4)**

* **Week 1**: Project kickoff, team onboarding
* **Week 2**: Requirements review, initial architecture planning
* **Week 3**: Project planning, environment setup
* **Week 4**: Initial backlog creation, planning completion
* **Milestone**: Project Plan Approved

**2.2.2 Analysis and Design (Week 5-12)**

* **Week 5-6**: Database design, API design
* **Week 7-8**: UI/UX design, architecture design
* **Week 9-10**: Integration design, security design
* **Week 11-12**: Design review and approval
* **Milestone**: Design Documentation Approved

**2.2.3 Development - Iteration 1 (Week 13-18)**

* **Week 13-14**: Database implementation, core API development
* **Week 15-16**: Initial web application components, shared components
* **Week 17-18**: Initial mobile application components, CI/CD setup
* **Milestone**: Core Functionality Demonstrated

**2.2.4 Development - Iteration 2 (Week 19-24)**

* **Week 19-20**: Portfolio and project management features
* **Week 21-22**: Requirements management, stakeholder management
* **Week 23-24**: Risk tracking, milestone planning
* **Milestone**: Feature Complete - Portfolio and Project Management

**2.2.5 Development - Iteration 3 (Week 25-30)**

* **Week 25-26**: User management, notification system
* **Week 27-28**: Reporting, mobile offline capabilities
* **Week 29-30**: Integration completion, final features
* **Milestone**: Feature Complete - All Requirements

**2.2.6 System Testing (Week 31-34)**

* **Week 31-32**: Functional testing, integration testing
* **Week 33-34**: Performance testing, security testing
* **Milestone**: System Testing Complete

**2.2.7 User Acceptance Testing (Week 35-37)**

* **Week 35**: UAT preparation, training
* **Week 36-37**: UAT execution, defect fixes
* **Milestone**: UAT Complete

**2.2.8 Deployment Preparation (Week 38-39)**

* **Week 38**: Production environment setup, final documentation
* **Week 39**: Deployment rehearsal, go-live planning
* **Milestone**: Ready for Deployment

**2.2.9 Go-Live and Stabilization (Week 40-42)**

* **Week 40**: Production deployment, go-live
* **Week 41-42**: Production support, stabilization
* **Milestone**: Project Complete

**2.3 Critical Path Items**

The following items are on the critical path and require close monitoring:

* Database design and implementation
* Core API development
* Shared component development
* Integration between web and mobile applications
* User acceptance testing
* Production environment setup

**2.4 Timeline Risks and Mitigations**

| **Risk** | **Impact** | **Mitigation Strategy** |
| --- | --- | --- |
| Requirements changes | Schedule delay | Change control process, buffer in timeline |
| Resource availability | Critical path delays | Resource management plan, cross-training |
| Technical challenges | Quality issues, delays | Prototyping, technical spikes, expert consultation |
| Integration issues | System functionality | Early integration testing, interface agreements |
| Performance issues | System usability | Early performance testing, scalability design |
| Mobile platform complexity | Mobile app delays | Early prototyping, platform-specific testing |
| UAT feedback volume | Deployment delay | Incremental UAT, defect prioritization process |

**3. Resource Allocation**

**3.1 Team Composition**

| **Role** | **Count** | **Allocation** | **Duration** |
| --- | --- | --- | --- |
| Project Manager | 1 | 100% | 42 weeks |
| Solution Architect | 1 | 100% | 42 weeks |
| Backend Developers | 3 | 100% | 36 weeks |
| Web Developers | 2 | 100% | 36 weeks |
| Mobile Developers | 2 | 100% | 36 weeks |
| Shared Component Developers | 1 | 100% | 36 weeks |
| QA Engineers | 3 | 100% | 32 weeks |
| DevOps Engineers | 1 | 100% | 36 weeks |
| UX/UI Designers | 1 | 100% | 20 weeks |
| Database Specialists | 1 | 50% | 36 weeks |
| Documentation Specialists | 1 | 50% | 36 weeks |

**3.2 Resource Loading Chart**

**3.2.1 Inception and Planning (Week 1-4)**

* Project Manager: 100%
* Solution Architect: 100%
* Backend Developers: 50%
* Web Developers: 25%
* Mobile Developers: 25%
* DevOps Engineers: 50%
* UX/UI Designers: 100%

**3.2.2 Analysis and Design (Week 5-12)**

* Project Manager: 100%
* Solution Architect: 100%
* Backend Developers: 75%
* Web Developers: 75%
* Mobile Developers: 75%
* Shared Component Developers: 100%
* DevOps Engineers: 75%
* UX/UI Designers: 100%
* Database Specialists: 100%

**3.2.3 Development (Week 13-30)**

* Project Manager: 100%
* Solution Architect: 75%
* Backend Developers: 100%
* Web Developers: 100%
* Mobile Developers: 100%
* Shared Component Developers: 100%
* QA Engineers: 75%
* DevOps Engineers: 100%
* UX/UI Designers: 50%
* Database Specialists: 50%
* Documentation Specialists: 50%

**3.2.4 Testing (Week 31-37)**

* Project Manager: 100%
* Solution Architect: 50%
* Backend Developers: 75%
* Web Developers: 75%
* Mobile Developers: 75%
* QA Engineers: 100%
* Documentation Specialists: 75%

**3.2.5 Deployment and Stabilization (Week 38-42)**

* Project Manager: 100%
* Solution Architect: 75%
* Backend Developers: 50%
* Web Developers: 50%
* Mobile Developers: 50%
* QA Engineers: 75%
* DevOps Engineers: 100%
* Documentation Specialists: 100%

**3.3 Resource Constraints and Considerations**

* **Specialized Skills**: Mobile development with .NET MAUI requires specific expertise
* **Shared Resources**: Some team members may be shared with other projects
* **Ramp-up Time**: Allow time for team members to become familiar with the project
* **Knowledge Transfer**: Plan for knowledge transfer for long-term support
* **Vacation/Time Off**: Account for team member vacations and time off
* **Surge Capacity**: Identify potential resources for peak demand periods
* **Support Transition**: Plan for transition to support team at project completion

**4. Cost Estimation**

**4.1 Cost Breakdown by Category**

| **Category** | **Estimated Cost** | **Notes** |
| --- | --- | --- |
| Labor - Development | $XXX,XXX | Based on average daily rates |
| Labor - Management | $XX,XXX | Project management and oversight |
| Labor - QA and Testing | $XX,XXX | Quality assurance activities |
| Infrastructure - Development | $X,XXX | Development and test environments |
| Infrastructure - Production | $X,XXX | Production hosting and services |
| Software and Tools | $X,XXX | Licenses and subscriptions |
| Training and Documentation | $X,XXX | Training materials and documentation |
| Contingency (15%) | $XX,XXX | Buffer for unforeseen expenses |
| **TOTAL** | **$XXX,XXX** | **Total Estimated Project Cost** |

*Note: Actual cost figures would be filled in based on organization-specific rates and pricing.*

**4.2 Payment Schedule**

* 10% upon project initiation
* 20% upon completion of analysis and design phase
* 25% upon completion of development iteration 1
* 20% upon completion of development iteration 3
* 15% upon completion of system testing
* 10% upon project completion and acceptance

**4.3 Cost Management Approach**

* Monthly budget tracking and reporting
* Earned Value Management (EVM) metrics
* Change order process for scope changes affecting budget
* Regular financial reviews with stakeholders
* Transparent reporting of budget variances
* Contingency management process

**5. Agile Project Management Approach**

**5.1 Sprint Schedule**

* Sprint duration: 2 weeks
* Sprint planning: First day of sprint
* Daily stand-ups: 15 minutes daily
* Sprint review: Last day of sprint
* Sprint retrospective: Last day of sprint
* Backlog refinement: Weekly

**5.2 Scrum Roles**

* Product Owner: Client representative
* Scrum Master: Project Manager or dedicated resource
* Development Team: Cross-functional team of developers, designers, and testers

**5.3 Ceremonies and Artifacts**

* **Sprint Planning**: Determine sprint goals and select backlog items
* **Daily Stand-up**: Synchronize activities and identify impediments
* **Sprint Review**: Demonstrate completed functionality to stakeholders
* **Sprint Retrospective**: Identify improvements for future sprints
* **Product Backlog**: Prioritized list of all desired features
* **Sprint Backlog**: Selected items for current sprint
* **Definition of Done**: Clear criteria for when an item is complete
* **Burndown Charts**: Visual tracking of sprint progress

**5.4 Release Planning**

* Release frequency: Every 3 sprints (6 weeks)
* Release planning: 1-day workshop before each release cycle
* Release notes: Documentation of features in each release
* Release retrospective: Review of release process

**5.5 Agile Metrics**

* Velocity: Story points completed per sprint
* Sprint burndown: Tracking of work remaining in sprint
* Release burndown: Tracking of work remaining in release
* Defect density: Number of defects per feature
* Escaped defects: Defects found after sprint completion
* Team satisfaction: Measured in sprint retrospectives