**ReqVista - Assumptions and Constraints**

**1. Project Assumptions**

**1.1 Business Assumptions**

* **Stakeholder Availability**: Key stakeholders will be available for requirements validation, reviews, and sign-offs within agreed timelines
* **User Access**: Access to representative end users will be provided for requirements validation and user acceptance testing
* **Business Continuity**: Any existing systems to be replaced will remain operational during the transition period
* **Scope Stability**: The core feature set as defined in requirements will remain stable, with change management processes used for modifications
* **Organizational Readiness**: The organization is prepared to adopt and support both web and mobile platforms for the ReqVista system
* **Existing Data**: If migrating from an existing system, data cleansing and preparation will be a shared responsibility

**1.2 Technical Assumptions**

* **Azure Environment**: Azure subscriptions with appropriate access levels will be available for development, testing, and production environments
* **Development Tools**: All developers will have access to required development tools, including Visual Studio and device emulators
* **Third-Party Services**: Licenses and access for any required third-party services or components will be procured in a timely manner
* **Mobile Devices**: A defined set of mobile devices and OS versions will be supported, as agreed during requirements phase
* **Network Access**: Appropriate network connectivity and access to environments will be provided to the development team
* **Technical Standards**: Existing technical standards and guidelines will be provided where applicable

**1.3 Project Management Assumptions**

* **Methodology**: The project will follow an Agile development methodology with 2-week sprints
* **Communication Channels**: Established communication channels and tools will be available for project collaboration
* **Decision Making**: A clear decision-making process will be in place with defined roles and authorities
* **Resource Availability**: Required resources with appropriate skills will be available according to the project schedule
* **Funding Stability**: Project funding will remain stable throughout the project lifecycle
* **Risk Management**: All parties will participate in ongoing risk identification and mitigation efforts

**2. Project Constraints**

**2.1 Schedule Constraints**

* **Fixed Deadlines**: Any fixed external deadlines that must be met (e.g., regulatory compliance, business events)
* **Resource Availability Windows**: Time periods when key resources may have limited availability
* **Deployment Windows**: Restrictions on when production deployments can occur
* **Sprint Calendar**: Predefined sprint schedule including holidays and organizational events
* **Stakeholder Availability**: Limitations on stakeholder availability for key decisions or approvals
* **Mobile Store Review Process**: Time required for app store review processes for mobile application

**2.2 Technical Constraints**

* **Technology Stack**: The defined C# .NET technology stack must be used as specified in requirements
* **Cloud Platform**: Azure must be used as the cloud platform for all hosted components
* **Database Technology**: Azure SQL must be used as the primary database technology
* **Mobile Framework**: .NET MAUI must be used for mobile application development
* **Web Framework**: Blazor must be used for web application development
* **Integration Limitations**: Any limitations with existing systems that must be integrated
* **Security Requirements**: Compliance with organization's security standards and requirements
* **Performance Requirements**: Minimum performance standards that must be met

**2.3 Resource Constraints**

* **Team Size Limitations**: Maximum team size and composition constraints
* **Specialized Skill Availability**: Limited availability of specialized skills (e.g., mobile development, DevOps)
* **Environment Limitations**: Limitations in development, testing, or production environments
* **Budget Constraints**: Fixed or limited budget allocations for specific project aspects
* **External Dependencies**: Reliance on external teams or vendors for certain components
* **Licensing Limitations**: Constraints related to third-party licenses or subscriptions

**3. Dependencies**

**3.1 Internal Dependencies**

* **Environment Provisioning**: Timely setup of Azure environments and resources
* **Access Management**: Provisioning of appropriate access to systems and tools
* **Design Approvals**: Timely review and approval of design artifacts
* **Configuration Management**: Establishment of configuration management processes
* **Quality Assurance Resources**: Availability of testing resources and environments
* **Data Availability**: Access to test data and potential migration data sources

**3.2 External Dependencies**

* **Third-Party Services**: Timely provision and configuration of any third-party services
* **Vendor Deliverables**: Timely delivery of any components from external vendors
* **Regulatory Approvals**: Any required regulatory or compliance approvals
* **User Training**: Scheduling and coordination of user training activities
* **Mobile App Store Processes**: App store submission and approval processes
* **External System Interfaces**: Coordination with owners of external systems for integration

**3.3 Critical Path Dependencies**

* **Architecture Definition**: Completion of architecture design before detailed component design
* **Data Model Finalization**: Finalization of data model before development of data-dependent components
* **API Development**: Completion of core API functionality before dependent front-end development
* **Integration Points**: Establishment of integration interfaces with external systems
* **User Acceptance Testing**: Completion of development and system testing before UAT
* **Security Review**: Security assessment prior to production deployment

**4. Project Exclusions**

**4.1 Scope Exclusions**

* **Legacy Data Migration**: Detailed data migration from legacy systems is not included unless explicitly stated
* **Advanced Analytics**: Advanced analytics and business intelligence features beyond basic reporting
* **Customized Mobile Versions**: Platform-specific mobile application versions beyond what .NET MAUI provides
* **Extensive Integrations**: Integrations with external systems beyond those explicitly defined in requirements
* **Multi-language Support**: Support for multiple languages unless explicitly stated in requirements
* **Custom Hardware Support**: Support for specialized hardware devices or peripherals
* **Extensive Training**: End-user training beyond defined training materials and sessions

**4.2 Technical Exclusions**

* **Unsupported Browsers**: Support for deprecated or uncommonly used web browsers
* **Offline Web Access**: Complete offline functionality for the web application (unless specifically required)
* **Custom Development Tools**: Development of custom tools for administration or development
* **Performance Engineering**: Extensive performance engineering beyond meeting defined requirements
* **Non-Azure Deployments**: Deployment to cloud platforms other than Microsoft Azure
* **Mobile Application Variants**: Development of native mobile applications beyond the .NET MAUI implementation

**5. Assumptions Validation Approach**

**5.1 Validation Process**

* **Regular Reviews**: Schedule regular reviews of assumptions throughout the project lifecycle
* **Milestone Validation**: Validate key assumptions at each project milestone
* **Risk Monitoring**: Track assumptions as potential risks in the risk register
* **Stakeholder Confirmation**: Obtain explicit confirmation of critical assumptions from stakeholders
* **Technical Validation**: Perform proof-of-concept or technical validation for critical technical assumptions
* **Documentation Updates**: Maintain assumptions log with validation status throughout the project

**5.2 Mitigation Strategies**

* **Contingency Planning**: Develop contingency plans for critical assumptions
* **Alternative Approaches**: Identify alternative approaches if key assumptions prove invalid
* **Scope Adjustment Process**: Define process for scope adjustment if assumptions impact feasibility
* **Escalation Path**: Clear escalation path for addressing invalidated assumptions
* **Change Management**: Framework for managing changes resulting from invalidated assumptions