## Requirements Traceability Matrix (RTM) - Requirements Gathering Agent

This RTM traces requirements from their source to verification methods, ensuring comprehensive traceability throughout the project lifecycle.

| Requirement ID | Requirement Description | Requirement Type | Priority | Source | Success Criteria | Test Case Reference | Verification Method | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FR001 | Generate PMBOK-compliant Project Charter | Functional | High | Project Charter, README | Charter successfully generated, containing all mandatory PMBOK elements (Project Goal, Stakeholders, etc.) | TC001 | Inspection, Automated Validation against PMBOK template | Complete |  |
| FR002 | Generate Stakeholder Register | Functional | High | Stakeholder Register, README | Register accurately lists all identified stakeholders with relevant attributes (roles, contact info, influence) | TC002 | Inspection, Data completeness check | Complete |  |
| FR003 | Generate Scope Management Plan | Functional | High | Scope Management Plan, README | Plan defines scope baseline, WBS, change management process, etc., conforming to PMBOK standards | TC003 | Inspection, Review by PMBOK expert | Complete |  |
| FR004 | Generate Risk Management Plan | Functional | High | Risk Management Plan, README | Plan identifies, analyzes, and prioritizes project risks with mitigation strategies | TC004 | Inspection, Risk assessment review | In Progress | Requires further risk identification from stakeholders |
| FR005 | Integrate with Azure OpenAI | Functional | High | README, Architecture doc | Successful API calls to Azure OpenAI, error handling implemented | TC005, TC006 | Unit tests, Integration tests, API call monitoring | Complete |  |
| FR006 | Support multiple AI providers (Azure, Google, GitHub) | Functional | Medium | README, Architecture doc | Ability to switch providers via configuration, fallback mechanisms implemented | TC007, TC008, TC009 | Unit tests, Integration tests, CLI testing | In Progress | Google and GitHub integration pending |
| NF001 | Generate documents in JSON format | Non-functional | High | README | Output is valid JSON, schema validation successful | TC010 | Automated schema validation, data parsing | Complete |  |
| NF002 | System must be robust and handle errors gracefully | Non-functional | High | README | Error messages are informative, system recovers from errors without data loss | TC011, TC012 | Unit tests, Integration tests, exception handling tests | Complete |  |
| NF003 | CLI should be user-friendly and intuitive | Non-functional | High | README | CLI commands are clear, help documentation is comprehensive | TC013 | User testing, Usability review | Complete |  |
| BR001 | Reduce project documentation time by 50% | Business | High | Project Charter | Time tracking data compared to manual documentation process | TC014 | Performance testing, Time tracking | In Progress | Data collection ongoing |
| BR002 | Improve documentation consistency and accuracy | Business | High | Project Charter | Comparison of generated documents to PMBOK standards and manual documents | TC015 | PMBOK compliance audit, Accuracy review | In Progress | Audit and review pending |
| FR007 | Generate Work Breakdown Structure (WBS) | Functional | High | README | WBS is complete, accurate, and follows hierarchical decomposition | TC016 | Inspection, WBS review | Complete |  |
| FR008 | Generate Professional Word (.docx) documents | Functional | High | README | Documents are formatted correctly, metadata is included, compatible with MS Word | TC017 | Inspection, Compatibility testing | Complete |  |
| NF004 | System should be secure and protect sensitive data | Non-functional | High | README | Secure authentication mechanisms in place, data encryption used where necessary | TC018, TC019 | Security audit, Penetration testing | In Progress | Security audit scheduled |
| FR009 | Provide comprehensive documentation and support | Functional | Medium | README | User guides, FAQs, and support channels available |  | Documentation review, User feedback | Complete |  |

**Legend:**

* **Requirement ID:** Unique identifier for each requirement.
* **Requirement Description:** Clear and concise description of the requirement.
* **Requirement Type:** Categorization of the requirement (Functional, Non-functional, Business).
* **Priority:** Importance of the requirement (High, Medium, Low).
* **Source:** Origin of the requirement (e.g., stakeholder interview, document).
* **Success Criteria:** Measurable criteria to determine if the requirement is met.
* **Test Case Reference:** Link to the test case used to verify the requirement.
* **Verification Method:** Method used to verify the requirement (e.g., inspection, testing, review).
* **Status:** Current status of the requirement (Complete, In Progress, Not Started, etc.).
* **Notes:** Additional information or comments.

This RTM is a living document and will be updated throughout the project lifecycle. Test case references are placeholders and would be replaced with actual test case IDs in a real-world scenario. The “Notes” column allows for tracking issues or dependencies.