WBS Dictionary

|  |  |  |
| --- | --- | --- |
| WBS Code | Name | Description |
| 1 | Project planning |  |
| 1.1 | Define Project Scope | Define project scope, objectives, and requirements. |
| 1.1.1 | Project plan scope | Outlines the scope, timeline, resources, and budget. |
| 1.2 | Identify requirements | Determine functional and non-functional requirements for the project. |
| 1.2.1 | Requirements document | Specifies functional and non-functional requirements including user stories, use cases, and acceptance criteria. |
| 1.3 | Plan project timeline | Create a schedule of project tasks and deliverables. |
| 1.3.1 | Risk management plan and milestones | Define the goals and success criteria for the project as well as risks created by project assumptions. |
| 2 | Design |  |
| 2.1 | Design document | Provides a detailed technical design for the project. |
| 2.1.1 | Design system architecture and database schema | Create a high-level design of the system architecture and infrastructure. |
| 2.2 | UI design | Outline the look and feel of the user interface and user experience. |
| 2.2.1 | Design VBA Excel UI | Outline the look and feel of the VBA Excel user interface and user experience. |
| 2.2.2 | Design web-based UI | Outline the look and feel of the user web interface and user experience. |
| 2.3 | Design security | Configure security and access control measures that ensure the confidentiality, integrity, and availability of the system. |
| 2.3.1 | Authentication and authorization | Develop account permissions specifications for each user account level. |
| 3 | Development |  |
| 3.1 | Develop UI | Develop the look and feel of the user interface and user experience. |
| 3.1.1 | Develop menu-based UI | Design the look and feel of the menu-based UI. |
| 3.1.2 | Develop UI legacy mode prompt | Design the look and feel of the legacy mode prompt. |
| 3.1.3 | Develop business analyst UI | Design the look and feel of the business analyst report UI. |
| 3.2 | Develop Excel VBA system | Develop the VBA system logic. |
| 3.2.1 | Develop database to schema specification | Develop the Access database of customer data to schema spec. |
| 3.2.2 | Develop user permission models | Develop the database user permissions and system logic. |
| 3.2.3 | Develop system testing and QA procedures | Create testing strategy and procedures to validate system functionality against stakeholder acceptance criteria. |
| 3.3 | Develop web implementation | Develop web implementation that builds off Excel VBA and Access system. |
| 3.3.1 | Develop web interface | Create the web interface that runs the web implementation. |
| 3.3.2 | Develop database and server-side logic | Ensure report queries and table modifications are to spec per each user abilities. |
| 3.3.3 | Integrate Excel VBA components | Connect VBA menu macros to the Access database. |
| 3.3.4 | Develop system documentation and user manuals | Create a user manual for support staff to easily reference commands and system instructions as well as a separate administrator manual for the owner and administrative staff. |
| 4 | Deployment |  |
| 4.1 | Testing and validation | Plan system testing. |
| 4.1.1 | Conduct validation testing | Perform security testing of VBA vulnerabilities, authentication, and authorization. |
| 4.1.2 | Conduct user acceptance testing | Conduct system testing ensuring functionality and requirements validation. |
| 4.1.3 | Conduct load testing | Execute load testing to ensure that the system can handle large data and high user requests. |
| 4.2 | Integration deployment | Ensure correctly integrated Excel VBA and web-based components. |
| 4.2.1 | Configure and deploy system to production server | Deploy integrated Excel VBA and web-based components to dedicated production server. |
| 4.2.2 | Migrate data from existing system | Migrate data from current ledger. |
| 4.3 | Post-deployment plan |  |
| 4.3.1 | Develop post-deployment and maintenance plan | Create a user manual for support staff outlining maintenance plan. |
| 5 | Maintenance and Support | Include system instructions as well as a separate administrator manual for the owner and administrative staff |
| 5.1 | Address system bugs | Address user bugs found from initial user testing. |
| 5.1.1 | Perform updates and patches | Conduct another iterative cycle of updates to address critical issues. |
| 5.1.2 | Monitor system performance and uptime | Throughout testing and launch, keep track of system performance and uptimes to gauge system reliability. |
| 5.2 | Provide user training | Provide initial user training for launch. |
| 5.2.1 | Maintain documentation | Outline documentation procedures for users to perform future changes. |
| 5.2.2 | Provide technical support | While monitoring the system during the initial launch, provide bug support. |
| 5.3 | System upgrades | Create a backlog of stakeholder upgrade features for potential future development. |
| 5.3.1 | Develop enhancement requests | If upgrades are requested within project scope, perform them or plan for a new project to support the new features. |
| 5.3.2 | Plan and implement new features and upgrades | Plan to submit a new project to support new requested features. |
| 5.4 | System recovery | Outline a system recovery procedure in the user documentation. |
| 5.4.1 | Communicate system changes to users | Keep in communication with users on system changes and patches. |
| 5.4.2 | Develop a disaster recovery plan | Outline a disaster recovery procedure in the user documentation. |
| 5.4.3 | Schedule regular system backups | In the recovery procedure directions, specify timed system backups and ensure that admin has instructions to perform them. |
| 5.4.4 | Perform data recovery testing | Test the recovery procedures outlined in the manual to ensure functionality. |

* **Speculate what features would you propose to deliver in each of 3 releases.**

**Alpha Release:**

1. **User Authentication: Implement the user authentication system for users to log in securely with their credentials to ensure access control and security.**
2. **Excel VBA Menu Integration: Develop and integrate the Excel VBA menu-driven application so users can easily interact with the new system.**
3. **Basic Ledger Access: Provide read-only access to the property ledger data for support staff and raw access to Sparky.**
4. **User Documentation: Create initial user manuals for support staff, outlining basic system commands and instructions. This will be necessary for early user training.**

**Beta Release:**

1. **User Role-Based Access: Define different user roles (e.g., administrator, business analyst, regular users) and assign appropriate permissions.**
2. **Business Analyst Reporting: Develop the user interface for business analysts to generate and view property management reports.**
3. **Web Solution Prototype: Build a prototype of the web-based user interface. Users can navigate and interact with the web interface, but not all functionalities are fully implemented.**
4. **Initial System Testing: Begin system testing, ensuring that core functionalities are working as expected.**

**Production Release:**

1. **Full Web-Based Interface: Complete the web-based user interface with all the features and functionalities. Users can access and manage property ledger data through the web interface on any device.**
2. **Database Integration: Ensure that the Excel VBA components are fully integrated with the Access database, allowing for data retrieval, storage, and manipulation based on user access permissions.**
3. **Administrator Features: Provide exclusive administrator access to logging data, customer data, and other restricted data in the Access database.**
4. **Complete User Documentation: Expand the user manuals to include comprehensive instructions for users, especially administrators. Document system maintenance, data recovery procedures, and disaster recovery plans.**
5. **Final System Testing and Validation: Conduct thorough testing, including security testing, user acceptance testing, and load testing. Address any remaining bugs and ensure that the system meets all requirements and quality standards.**