**TURNQEY**

**PROJECT PLAN**

**VERSION 1.0 | 07/05/2023**

**Table of Contents**

[**1.** **Introduction** 3](#_Toc139486322)

[**Project Description** 3](#_Toc139486323)

[**Project Objectives** 3](#_Toc139486324)

[**Project Scope** 3](#_Toc139486325)

[**2. Assumptions and Constraints** 3](#_Toc139486326)

[**Assumptions** 3](#_Toc139486327)

[**Constraints** 3](#_Toc139486328)

[**3. Project Team** 4](#_Toc139486329)

[**4. Project Timeline** 5](#_Toc139486330)

[**Milestones** 5](#_Toc139486331)

[**Deliverables** 6](#_Toc139486332)

[**5. Current Analysis** 6](#_Toc139486333)

[**6. Quality Assurance** 6](#_Toc139486334)

[**Testing Plan** 6](#_Toc139486335)

[**Acceptance Criteria** 7](#_Toc139486336)

[**7. Communication Plan** 8](#_Toc139486337)

[**8. Ticketing System** 9](#_Toc139486338)

[**Software** 9](#_Toc139486339)

[**Materials** 9](#_Toc139486340)

[**9. Resources** 9](#_Toc139486341)

[**10. Project Plan Approval** 9](#_Toc139486342)

# **1.** **Introduction**

## **Project Description**

The project aims to develop a comprehensive solution for aggregating and consolidating portfolio data from various cryptocurrency exchanges and wallets into a single, centralized platform. By doing so, the project will provide users with a convenient and efficient way to manage their cryptocurrency portfolios, monitor their investments, and make informed trading decisions.

## **Project Objectives**

* **Data Retrieval:** Download user data from various exchanges and wallets.
* **Data Consolidation:** Provide an API to aggregate and consolidate portfolio data.
* **Portfolio Analysis:** Offer portfolio analysis features including trade analysis, transfer analysis, balance tracking, cost basis calculation, return calculation, and annual return calculation.

## **Project Scope**

* **Data Aggregation:** The project scope includes developing a system that can securely and accurately aggregate portfolio data from multiple cryptocurrency exchanges and wallets. This involves integrating with APIs or establishing secure connections to retrieve relevant data such as account balances, transaction history, and asset prices.
* **User Management and Security:** The scope of the project encompasses implementing user management functionalities, user authentication, and authorization. Security measures such as encryption, data privacy, and secure connections will be integrated into the platform to ensure the protection of users' sensitive information and assets.

# **2. Assumptions and Constraints**

## **Assumptions**

* 100 accounts will be onboarded by the end of August with data of trade/transactions/withdrawal/deposits/buy/sell.
* 5000 accounts will be onboarded by end of December.

## **Constraints**

* **API and Integration Limitations:** The project's success relies on the availability and functionality of APIs provided by cryptocurrency exchanges and wallets. The project team must work within the constraints imposed by these APIs, including any limitations on data access, rate limits, or authentication requirements. Integration challenges may arise if certain exchanges or wallets do not offer suitable APIs or require additional permissions for data retrieval.
* **Security and Compliance:** Cryptocurrency transactions involve sensitive financial data, and security is of utmost importance. The project must adhere to industry best practices for data encryption, secure storage, and access control to ensure the confidentiality and integrity of users' portfolio information.
* **Market Volatility and Data Accuracy:** Cryptocurrency markets are known for their volatility, and the project must account for this when aggregating and presenting portfolio data. Real-time data updates and accurate pricing information are crucial for users to make informed trading decisions. The project team must address challenges such as delays in data updates, price discrepancies across exchanges, and ensuring data accuracy and reliability.

# **3. Project Team**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Current Team** | **Projected Team** | **Responsibilities** |
| **Project Manager** |  | **1** | * Establish project goals, define project scope, and create a comprehensive project plan that includes timelines, resources, and milestones. * Ensure that the project team understands their roles and responsibilities, and they provide guidance and direction throughout the project. |
| **System Developer** | **3** | **3** | * Responsible for designing, coding, and implementing software systems or applications. * They analyze requirements, design system architecture, develop software components, and conduct thorough testing to ensure the functionality and performance of the system. |
| **Quality Assurance Manager** |  | **1** | * Establish and enforce quality standards, policies, and procedures to ensure compliance with industry regulations and best practices. * Provide strategic direction and guidance to the QA team, ensuring that quality objectives are defined, communicated, and achieved. |
| **Configuration Management Manager / Admins** | **1** | **1** | * Define the configuration management strategy, including version control, change management, and release management practices. * Manage the release process, coordinating with development, testing, and deployment teams to ensure that changes are released in a controlled and coordinated manner. |
| **Database Administrator** | **1** | **1** | * Responsible for managing and maintaining the organization's databases. * Monitor database performance, conduct regular backups, and ensure data integrity and security. * Responsible for data administration tasks, including data modeling, schema design, and data dictionary management. * Manages user access and permissions, implements database upgrades and patches, and troubleshoots issues related to database performance or functionality. |
| **Testing Analyzer** |  | **1** | * Responsible for analyzing project requirements and test documentation to develop a comprehensive test plan and strategy. * Analyze software specifications, user stories, and other relevant documentation to identify test scenarios, test cases, and test data. * Identify and report defects, investigate their root causes, and provide detailed analysis and documentation for the development team. |
| **Morningstar Support Team** |  | **2** | * Responsible for aiding and resolving issues faced by users of a product or service. They respond to user inquiries, troubleshoot problems, and provide guidance on how to use the product effectively. * Plays a role in providing user training and education. They conduct training sessions, create user documentation, and develop resources such as knowledge bases or FAQs. * The team ensures that users have access to the necessary information and resources to maximize their understanding and utilization of the product or service. |
| **Data Analyst** |  | **1** | * Make sure data and calculation is correct. * SQL, Crypto knowledge. |

# **4. Project Timeline**

## **Milestones**

|  |  |  |
| --- | --- | --- |
| **Milestone** | **Description** | **Deliverables / Notes** |
| Data Retrieval Implementation | Develop and implement the data retrieval mechanisms for downloading user data from exchanges and wallets. | Data downloaders/services integration, successful retrieval of sample data from selected platforms. |
| API Development | Design and develop the API that will aggregate and consolidate the downloaded user data. | API documentation, functional API endpoints, and successful integration with data retrieval mechanisms. |
| Web Application Development | Build the web application that will serve as the user interface for the project. | User interface design, functional web application with basic features, such as wallet and exchange linking. |
| OAuth Implementation | Implement the OAuth for secure and user-friendly access. | OAuth application integration, successful user authentication and authorization with exchange and wallet accounts. |
| Portfolio Analysis Features | Develop and integrate portfolio analysis features, including balances, cost basis calculation, return calculation, and annual return calculation. | Functional portfolio analysis tools and reports. |
| Performance Optimization | Identify and address performance-related issues to ensure the system operates efficiently. | Performance optimization plan, implemented optimizations. |
| Deployment and User Acceptance Testing (UAT) | Deploy the project in a production environment and conduct UAT with a select group of users. | Deployed web application, user feedback and acceptance. |
| Establish Support Infrastructure | Set up the necessary infrastructure and tools to support ongoing maintenance activities. | Maintenance environment, issue tracking system, and knowledge base. |
| Continuous Monitoring and Maintenance | Continuously monitor the system, address maintenance tasks, and promptly resolve any issues that arise. | Ongoing monitoring reports, regular maintenance tasks completed. |
| Issue Prioritization and Management | Establish a process for prioritizing and managing reported issues or requests for enhancements. | Issue prioritization framework, issue management system. |
| Bug Fixes and Patching | Address and resolve reported bugs and vulnerabilities in the system. | Bug fixes, patches, and updated system versions. |
|  |  |  |

## **Deliverables**

* Web Application: A web-based application that serves as the user interface for the project.
* Data Downloaders [Services]: Services or components responsible for downloading user data from exchanges and wallets.
* API: An interface that aggregates and consolidates portfolio data, providing a standardized way to access the combined information.
* OAuth: An authentication and authorization mechanism for secure user access to the project.

# **5. Current Analysis**

* **Production Database**
* Total Size: 50 GB
* Current Use:19 GB
* **Test Database**
* Total Size: 20 GB
* Current Use: 440 MB
* **Virtual Machine**
* Ram: 8 vCPUs, 32 GB
* **Disk**: 127 GB
* **Consumption**: 120 GB
* **Web Apps [11]**
* Ram: 1.7 GB
* **Storage**: 10GB

# **6. Quality Assurance**

## **Testing Plan**

**Objectives**

* Ensure functional correctness, reliability, and performance of the software.
* Validate user experience and usability.
* Identify and resolve defects to enhance product quality.

**Scope**

* Testing all functional modules and features of the software.
* Compatibility testing across supported platforms and browsers.

**Testing Approach and Methodology**

* Test-driven development (TDD) will be employed, with test cases created before development.
* Testers will collaborate closely with developers and stakeholders throughout the project.

**Test Deliverables**

* Test plan document outlining testing objectives, approach, and scope.
* Test cases and test scripts with step-by-step instructions for executing tests.
* Test data and test environment setup documentation.
* Test execution reports and defect tracking logs.
* Test closure report summarizing testing activities, results, and lessons learned.

## **Acceptance Criteria**

**Test Execution Process**

* Test case execution based on prioritized test scenarios.
* Defect logging, tracking, and retesting of resolved issues.
* Regression testing after defect fixes or software changes.

**Test Completion Criteria**

* All high-priority test cases executed and passed.
* Defects of severity level 'Critical' and 'High' resolved and retested.

**Test Reporting**

* Weekly test status reports including test progress, defect summary, and metrics.
* Defect reports with detailed information on each identified issue.
* Test metrics on test coverage, pass/fail ratios, and defect densities.

**Risks Plan**

* Resource constraints impacting testing timelines.
* Technical issues in the test environment.
* Dependencies on external systems for integration testing.

**Contingency Plans**

* Prioritize testing activities based on critical features and risk assessment.
* Allocate additional resources if necessary.
* Establish communication channels with external system providers to address integration challenges.

**Roles and Responsibilities**

* Test Manager: Overall responsibility for the QA testing activities.
* Test Lead: Test planning, test case creation, and coordination with the development team.
* Testers: Test execution, defect reporting, and regression testing.
* Developers: Collaborate with testers, resolve defects, and ensure timely fixes.

# **7. Communication Plan**

**Communication Objectives**

* Facilitate effective and timely communication among project stakeholders.
* Ensure transparency, alignment, and clarity of project information.
* Provide a framework for resolving issues, managing risks, and making decisions.

**Communication Channels**

* Regular project meetings weekly with agenda and meeting minutes.
* Email communications for announcements, updates, and formal documentation.
* Status reports monthly to provide project updates and progress.

**Communication Frequency and Timing**

* Project Team: Daily communication for progress updates, issue resolution, and task coordination.
* Stakeholder Meetings: Meetings scheduled as per project needs and milestones.
* Status Reports: Weekly status reports distributed every Tuesday.
* Urgent Communications: Immediate communication for critical issues and risks.

**Communication Content**

* Project updates, including progress, milestones, and achieved deliverables.
* Risks and mitigation strategies.
* Change requests and their impacts.
* Issues and their resolution status.
* Training and support requirements.
* Feedback and input from stakeholders.

**Escalation Procedure**

* Identify a hierarchy of escalation contacts and their respective roles.
* Define the process and criteria for escalating issues and risks.
* Establish a clear chain of communication for quick resolution of escalated matters.

**Feedback Mechanism**

* Encourage stakeholders to provide feedback and suggestions.
* Set up a designated feedback channel or use project collaboration tools.
* Review and address feedback in a timely manner to improve communication effectiveness.

# **8. Ticketing System**

## **Software**

**Jira**

* **Description:** Jira is a robust ticketing system with advanced features and comprehensive project management capabilities. It offers extensive customization options and supports various agile methodologies.
* **Price**: Jira offers various pricing tiers, including free and paid options. The pricing varies based on the number of users and additional features required. It also provides enterprise plans for larger organizations.

**Trello**

* **Description**: Trello is a user-friendly ticketing system that provides basic issue tracking and collaboration features. It offers a simple and straightforward workflow management approach.
* **Price**: Trello offers a free plan with limited features and affordable paid options. The paid plans provide additional functionality and customization options.

## **Materials**

API document: <https://turnqey.gitbook.io/api-docs/>

Website document – In Progress

Ticket System Management document - Pending

# **9. Resources**

After completing the stress testing and relevant testing, we will provide a comprehensive resource document that includes detailed information about the testing process, methodologies, and results.

## **Scaling of record**

## **Projected Resources**

# **10. Project Plan Approval**

The undersigned acknowledge that you have reviewed the TURNQEY Project Plan and agree with the information presented within this document.

|  |  |  |
| --- | --- | --- |
| Signature | Name | Designation |
|  |  |  |