**Functional Requirements Specification (FRS)**

**Project Name:** Pi Game - Phase 2 Expansion  
**Version:** 1.0  
**Prepared by:** IT Business Analyst Team

**Table of Contents**

1. **Introduction**  
     1.1 Purpose  
     1.2 Document Conventions  
     1.3 Requirement Scope  
     1.4 References
2. **Overall Description**  
     2.1 Product Perspective  
     2.2 Product Features  
     2.3 User Classes and Characteristics  
     2.4 Operating Environment  
     2.5 Assumptions and Dependencies
3. **Functional Requirements**  
     3.1 Context  
     3.2 Data/User Flow Diagrams  
     3.3 Functional Requirements  
       3.3.1 Ticket Shop – In-App Purchase  
       3.3.2 Wallet Integration – TON Onchain Check-in
4. **Nonfunctional Requirements**  
     4.1 Performance Requirements  
     4.2 Safety Requirements  
     4.3 Security Requirements  
     4.4 Software Quality Attributes
5. **Other Requirements**

Appendix A: Glossary  
Appendix B: Analysis Models  
Appendix C: Issues List

### **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Changes** | **Version** |
| IT BA Team | 18/04/2025 | Initial Draft | 1.0 |
|  |  |  |  |
|  |  |  |  |

## 1. Introduction

### ***1.1 Purpose***

This document specifies the functional and non-functional requirements for the Phase 2 Expansion of the Ton Tycoon project, focusing on the In-App Purchase module (Ticket Shop) and TON Wallet-based features.

### ***1.2 Document Conventions***

All functional requirements are labeled as **FR-XXX** and non-functional as **NFR-XXX**. Priorities follow the BRD conventions (1: Critical, 2: High, etc.).

### ***1.3 Requirement Scope***

This FRS covers the enhancements and new feature sets related to:

* In-app purchase via Telegram Stars
* Wallet connection and onchain check-in
* Transaction visibility and error handling

### ***1.4 References***

* Business Requirements Document (BRD) – Ton Tycoon Phase 2
* Telegram API Documentation
* TON Wallet Integration Guide

## 2. Overall Description

### ***2.1 Product Perspective***

This module is an extension of the existing Ton Tycoon gaming ecosystem. It adds monetization, wallet integration, and transaction traceability functionalities. It connects to Telegram's in-app purchase system and external TON Wallet APIs.

### ***2.2 Product Features***

* Ticket Shop with dynamic package listing
* Purchase confirmation and history
* TON Wallet connection and status
* Blockchain check-in mechanism
* UI support for user purchase feedback

### ***2.3 User Classes and Characteristics***

|  |  |
| --- | --- |
| User Class | Description |
| Player | End-users of the Ton Tycoon game platform. Requires ease-of-use and low learning curve. |
| Admin | Internal users managing configurations and monitoring user purchases. |
| QA/Dev | System users involved in testing, debugging, and feature rollout. |

### ***2.4 Operating Environment***

* Telegram App (latest version)
* Supported Browsers (Chrome, Safari, Firefox)
* Mobile OS: iOS 14+, Android 10+
* External APIs: Telegram Payments, TON Wallet

### ***2.5 Assumptions and Dependencies***

* User already logged in to Telegram
* User has existing TON Wallet or creates one via Tonkeeper
* Internet connection is required at all times

## 3. Functional Requirements

### ***3.1 Context***

System interacts with users through UI components embedded in Telegram bots and WebViews. Wallet data is pulled from external TON APIs.

### ***3.2 Data/User Flow Diagrams***

To be defined during design phase (pending UX input).

### ***3.3 Functional Requirements***

#### 3.3.1 Ticket Shop – In-App Purchase

* **FR-001**: System shall display a list of purchasable packages to users.
* **FR-002**: User shall be able to initiate purchase using Telegram Stars.
* **FR-003**: On purchase confirmation, system shall deduct stars and issue tickets.
* **FR-004**: System shall log transaction (timestamp, amount, item ID).
* **FR-005**: System shall notify users of success/failure via modal UI.

#### 3.3.2 Wallet Integration – TON Onchain Check-in

* **FR-006**: System shall prompt users to connect their TON Wallet.
* **FR-007**: On successful connection, system shall retrieve wallet balance.
* **FR-008**: System shall validate if balance is sufficient for participation.
* **FR-009**: On chain check-in action shall be recorded on blockchain.
* **FR-010**: System shall handle wallet disconnect or timeouts gracefully.

## 4. Nonfunctional Requirements

### ***4.1 Performance Requirements***

* **NFR-001**: Wallet connection shall not exceed 3 seconds.
* **NFR-002**: Purchase confirmation shall respond within 2 seconds.

### ***4.2 Safety Requirements***

* No physical risks. Ensure financial integrity in ticket purchase process.

### ***4.3 Security Requirements***

* **NFR-003**: All wallet API communication must use HTTPS.
* **NFR-004**: System must validate tokens for every wallet transaction.

### ***4.4 Software Quality Attributes***

* Availability: 99.9% uptime for wallet and payment services
* Usability: Mobile-first design, simple flows for non-technical users
* Reliability: Retry mechanism for failed purchase transactions

## 5. Other Requirements

* Transaction records must be stored for 6 months minimum.
* Localization required for English and Vietnamese.

## Appendix A: Glossary

* **Telegram Stars**: In-app currency used for purchases
* **TON**: The Open Network blockchain
* **Check-in**: Daily interaction recorded onchain

## Appendix B: Analysis Models

(To be added in Design phase)

## Appendix C: Issues List

|  |  |  |
| --- | --- | --- |
| Issue ID | Description | Status |
| TBD-001 | Define UI flow for wallet error state | Open |
| TBD-002 | Confirm multi-language copy with content team | Open |