TECHNICAL PROJECT DOCUMENTATION

LUMENBORN

2D Multi-Layer Idle/Action RPG

Document Type	Project Design Document (PDD) / Game Design Document (GDD)
Version	1.0.0
Date	January 2026
Status	In Development
Classification	Professional Portfolio

Developed with agile methodologies and software engineering best practices

Table of Contents

1. Executive Summary

1.1 Overview

Lumenborn is a mobile/desktop game development project that combines idle game mechanics with Action RPG elements. The system was architected to support cross-platform deployment, real economy between players, and scalable infrastructure for dedicated servers and community modification support.

1.2 Project Objectives

- Develop a cross-platform 2D game with scalable architecture
- Implement economy system with real transactions (P2P marketplace)
- Create infrastructure for dedicated servers and mod support
- Ensure balanced experience between paying and non-paying players
- Establish CI/CD pipeline for continuous deployments

1.3 Stakeholders

Stakeholder	Role	Responsibility
Developer	Full-Stack Dev	Architecture, code, deploy
End Users	Players	Feedback, beta testing
Community	Modders	Additional content

2. Project Scope

2.1 In Scope

- 2D side-scroller game engine with stage/level system
- Multi-layer progression system (Action, Village, Skills)
- Backend for authentication, persistence and synchronization
- Economy system with P2P marketplace
- Payment gateway integration for real transactions
- RESTful API for client-server communication
- Admin dashboard for management
- Anti-fraud and anti-bot systems
- Cross-platform support (Android, iOS, Windows, macOS)

2.2 Out of Scope - v1.0

- Virtual/augmented reality
- Synchronous multiplayer mode (real-time PvP)
- Console versions (PlayStation, Xbox, Switch)
- Separate companion app

2.3 Assumptions

- User has device with Android 8.0+, iOS 13+, or Windows 10+
- Internet connection required for online features
- Offline mode available with later synchronization

2.4 Constraints

- Limited budget - solo/small team development
- 10-12 month timeline for MVP
- Compliance with store policies (Google Play, App Store)

3. Functional Requirements

3.1 Core Gameplay Module

ID	Requirement	Priority	Status
FR001	Automatic character movement system	High	Planned
FR002	Touch/click input detection for interactions	High	Planned
FR003	Combat system with damage, health, death	High	Planned
FR004	Enemy spawner with per-level configuration	High	Planned
FR005	Resource collection system (orbs, items)	High	Planned
FR006	Combo system with multiplier	Medium	Planned
FR007	Level progression (1-10, boss, new biome)	High	Planned
FR008	Level selection (free migration)	Medium	Planned

3.2 Progression Module

ID	Requirement	Priority	Status
FR009	Eco-Prestige system (reset 80%, gain 20% permanent)	High	Planned
FR010	Prestige currency and permanent upgrade shop	High	Planned
FR011	4 skill trees with active/passive skills	Medium	Planned
FR012	Class system with specific skills	Low	Future

3.3 Inventory and Items Module

ID	Requirement	Priority	Status
FR013	6 equipment slots (weapon, helm, chest, gloves, boots, amulet)	High	Planned
FR014	Rarity system (Common → Primordial)	High	Planned
FR015	Legendary items with unique powers	Medium	Planned
FR016	Set system with 2/4/6 piece bonuses	Medium	Planned
FR017	Upgrade system +1 to +15 (MU-style)	Low	Future

3.4 Village Module

ID	Requirement	Priority	Status
FR018	Blacksmith - weapon upgrade and crafting	High	Planned
FR019	Merchant - item buy/sell	High	Planned
FR020	Church - healing and light buffs	High	Planned

FR021	Alchemist - potions and consumables	High	Planned
FR022	Angelic Altar - power extraction (safe)	Medium	Planned
FR023	Demonic Altar - power extraction (risky)	Medium	Planned

3.5 Economy Module

ID	Requirement	Priority	Status
FR024	Auction House with in-game currency (Light)	High	v2.0
FR025	Auction House with real money (gateway)	High	v2.0
FR026	Digital wallet system for players	High	v2.0
FR027	15% transaction fee (reinvestment)	High	v2.0
FR028	Withdrawal system with KYC verification	High	v2.0

3.6 Online Module

ID	Requirement	Priority	Status
FR029	Authentication system (email, OAuth)	High	v1.5
FR030	Cloud save with synchronization	High	v1.5
FR031	Global and seasonal leaderboards	Medium	v2.0
FR032	Vanilla/Modded separation with validation	High	v2.0
FR033	API for community dedicated servers	Low	v3.0
FR034	Mod system with runtime loading	Low	v3.0

4. Non-Functional Requirements

4.1 Performance

ID	Requirement	Metric
NFR001	Stable frame rate on mid-range devices	≥ 60 FPS
NFR002	Initial loading time	< 5 seconds
NFR003	RAM memory usage	< 500 MB
NFR004	APK/IPA size	< 150 MB
NFR005	API response time	< 200 ms

4.2 Scalability

ID	Requirement	Metric
NFR006	Supported concurrent users	10,000+ CCU
NFR007	Marketplace transactions per second	1,000+ TPS
NFR008	Architecture must allow horizontal scaling	Auto-scaling

4.3 Security

ID	Requirement	Metric
NFR009	Sensitive data encryption	AES-256
NFR010	Client-server communication	TLS 1.3
NFR011	Token authentication	JWT + Refresh
NFR012	GDPR compliance	100%
NFR013	Server-side anti-cheat system	Full validation

4.4 Availability

ID	Requirement	Metric
NFR014	Service uptime	99.9% SLA
NFR015	Functional offline mode	100% gameplay
NFR016	Data backup	Daily + real-time

5. System Architecture

5.1 Architecture Overview

The system uses client-server architecture with clear separation of concerns. The client (game) is responsible for rendering and input, while the authoritative server validates all critical actions to ensure integrity.

High-Level Architecture Diagram:

```
 ┌─────────────────────────────────────────────────────────────────┐ │                          CLIENTS
│ │  ┌──────────┐  ┌──────────┐  ┌──────────┐  ┌──────────┐        │ │  │ Android  │  │   iOS    │  │
Windows  │  │  macOS   │        │ │  └────┬─────┘   └─────────┘   └────┬────┘  └────┬─────┘        │ │
└─────────────┴─────────────┴─────────────┘               │ │                            │ HTTPS/WSS
│ └───────────────────────────┼─────────────────────────────────────┘                             ▼
┌───────────────────────────────────────────────────────────────┐ │                      LOAD BALANCER
│ │                    (AWS ALB / Nginx)                           │
└───────────────────────────┬───────────────────────────────────┘                             ▼
┌───────────────────────────────────────────────────────────────┐ │                    API GATEWAY
│ │              (Rate Limiting, Auth, Routing)                    │
└───────────────────────────┬───────────────────────────────────┘                             ▼
┌─────────────────────────────────────────────────────────────────┐ │                      MICROSERVICES
│ │  ┌─────────────┐  ┌─────────────┐  ┌─────────────┐              │ │  │ Auth Service│  │Game Service
│  │Economy Svc  │              │ │  └──────────────┘  └────────────┘  └─────────────┘              │ │
┌─────────────┐  ┌─────────────┐  ┌─────────────┐              │ │  │Ranking Svc  │  │ Anti-Cheat  │  │
Mod Service │             │ │   └─────────────┘   └────────────┘  └─────────────┘               │
└───────────────────────────┬─────────────────────────────────────┘                              ▼
┌─────────────────────────────────────────────────────────────────┐ │                        DATA LAYER
│ │  ┌─────────────┐  ┌─────────────┐  ┌─────────────┐              │ │  │ PostgreSQL   │ │    Redis
│  │     S3      │              │ │  │  (Primary)   │  │  (Cache)   │  │  (Assets)    │             │ │
└─────────────┘  └─────────────┘  └─────────────┘              │
└─────────────────────────────────────────────────────────────────┘
```

5.2 Architectural Patterns

- **Client:** Component-Based Architecture (Unity ECS-like)
- **Backend:** Microservices with RESTful API
- **Communication:** Request-Response (HTTP) + Event-Driven (WebSocket)
- **Persistence:** Repository Pattern with Unit of Work
- **Cache:** Cache-Aside Pattern with Redis

6. Technology Stack

6.1 Client (Game)

Component	Technology	Justification
Game Engine	Unity 2022 LTS	Cross-platform, market share
Language	C#	Unity native, strong typing
Networking	Unity Netcode / Mirror	Sync, dedicated servers
UI Framework	Unity UI Toolkit	Performance, flexibility
Serialization	JSON / MessagePack	Compatibility, performance
Analytics	Unity Analytics + Firebase	Metrics, events

6.2 Backend

Component	Technology	Justification
Runtime	Node.js / .NET Core	Performance, ecosystem
Framework	Express / ASP.NET	Maturity, support
Database	PostgreSQL	ACID, JSON support
Cache	Redis	Performance, pub/sub
Message Queue	RabbitMQ / Redis Streams	Async processing
Payments	Stripe / PayPal	Global, reliable

6.3 Infrastructure

Component	Technology	Justification
Cloud Provider	AWS / GCP	Scalability, services
Containers	Docker + Kubernetes	Orchestration, scaling
CI/CD	GitHub Actions	Integration, automation
CDN	CloudFront / CloudFlare	Assets, latency
Monitoring	Prometheus + Grafana	Metrics, alerts
Logging	ELK Stack	Centralization, search

7. Data Modeling

7.1 Main Entities

Player

```
 Player {   id: UUID (PK)   email: VARCHAR(255) UNIQUE   username: VARCHAR(50) UNIQUE
password_hash: VARCHAR(255)   created_at: TIMESTAMP   last_login: TIMESTAMP   is_banned:
BOOLEAN   wallet_balance: DECIMAL(10,2)   total_playtime: INTEGER   settings: JSONB }
```

Character

```
 Character {   id: UUID (PK)   player_id: UUID (FK -> Player)   name: VARCHAR(50)   class_id:
INTEGER   level: INTEGER   experience: BIGINT   current_stage: INTEGER   max_stage_unlocked:
INTEGER   is_vanilla: BOOLEAN   prestige_count: INTEGER   prestige_currency: INTEGER
base_stats: JSONB   light_dark_balance: INTEGER (-100 to 100)   created_at: TIMESTAMP }
```

Item

```
 Item {   id: UUID (PK)   template_id: INTEGER (FK -> ItemTemplate)   character_id: UUID (FK
-> Character)   rarity: ENUM(common, magic, rare, legendary, set, ancestral, primordial)
upgrade_level: INTEGER (0-15)   stats: JSONB   unique_power_id: INTEGER   is_equipped:
BOOLEAN   equipment_slot: ENUM(weapon, helm, chest, gloves, boots, amulet)   is_soulbound:
BOOLEAN   acquired_at: TIMESTAMP }
```

10. Economy System

10.1 Economic Philosophy

"Time invested > Money invested"

The economic system was designed so dedicated players always have an advantage over players who only spend money. This is achieved through a power split where the majority comes from non-purchasable sources.

10.2 Power Distribution

Power Source	% of Total	Purchasable?
Equipped Items	35%	Yes (via Auction House)
Skill Levels	25%	No (gameplay only)
Eco-Prestiges + Statues	20%	No (gameplay only)
Village (Buildings)	15%	Acceleration only
Mastery	5%	No (gameplay only)

10.3 MU x100 Style Progression

Playtime	Expected Progress
30 minutes	Level 10, first boss defeated, blue items
2 hours	Level 30, 3 bosses, some rare items
1 day	Level 50+, basic build formed
1 week	Level 100+, legendaries, first prestige
1 month	Multiple prestiges, optimized builds

10.4 Anti-Bot/Anti-P2W Strategy

- Generous drops = abundant items = naturally low prices
- Best items are Soulbound (non-tradeable)
- Stat caps based on Prestige (can't use OP items without playing)
- 7-day cooldown to resell purchased items
- Daily purchase limits (5 legendaries, 2 sets, 1 ancestral)

Result: Bots don't profit because prices are low. Buying gives no real advantage because 65% of power is non-purchasable.

14. Development Roadmap

Version	Timeline	Deliverables
v1.0	Months 1-10	Complete single-player game, all core mechanics, publication
v1.5	Months 11-13	Account system, cloud save, basic leaderboards
v2.0	Months 14-18	Vanilla/Modded, Auction House, rankings, seasons, anti-bot
v2.5	Months 19-21	Advanced anti-bot (ML), improved report system
v3.0	Months 22-28	Mod API, dedicated servers, mod hub
v3.5+	Months 29+	Expansions, new biomes, classes, seasonal events

15. Estimates and Metrics

15.1 Effort Estimate (v1.0)

Module	Estimate	Complexity
Setup + Unity Learning	80 hours	Medium
Core Gameplay	120 hours	High
Item System	80 hours	High
Progression System	60 hours	Medium
Village and Buildings	40 hours	Medium
UI/UX	60 hours	Medium
Art and Assets	100 hours	High
Audio	20 hours	Low
Testing and Polish	60 hours	Medium
TOTAL	**~620 hours**	-

15.2 Project KPIs

- DAU (Daily Active Users) - Target: 10,000+
- Retention D1/D7/D30 - Target: 40%/20%/10%
- ARPU (Average Revenue Per User) - Target: $0.50/month
- Transaction Volume - Target: $10,000/month
- Bug Rate - Target: < 0.1% sessions with crash

16. Technical Glossary

Term	Definition
Eco-Prestige	Partial reset system where player sacrifices 80% of current progress to gain 20% as permanent bonus
Vanilla	Official game mode without modifications, with ranking and official economy
Modded	Game mode with community modifications, without official ranking
Soulbound	Item that cannot be transferred or sold between players
Auction House	P2P marketplace where players can buy/sell items
CCU	Concurrent Users - simultaneously connected users
KYC	Know Your Customer - identity verification for financial transactions
Idle Game	Game genre where progression continues even without active interaction

END OF DOCUMENT

Lumenborn - Project Design Document v1.0

Document prepared following software engineering standards