Gacha Game: Duck Spinning

Project Report

Advanced Software Engineering 2024/2025
University of Pisa

Students:

Alexandra Pavlova, Othman Alhammali Shoaib Alhadiri,
Ermias Mulugeta Teklehaimanot

December 7, 2024

Introduction

This report describes the development of the Gacha Game, a web application that allows users to spin and collect virtual ducks of varying rarities. The game provides an engaging experience by combining user registration, authentication, and a randomized collection system. The project was developed as part of the Advanced Software Engineering course at the University of Pisa for the 2024/2025 academic year.

Project Overview

The Gacha Game offers users the ability to:

- Register an account.
- Log in and manage their profile.
- Spin for random ducks that are categorized by rarity.
- Collect multiple ducks, including duplicates.
- View their collection of ducks.
- Participate in an auction system to trade ducks.

This report discusses the project architecture, technologies used, implementation challenges, and future steps for improvement.

Technologies Used

- **Backend:** Python 3.x, Django 5.1.2, Django REST Framework
- **Database:** SQLite, PostgreSQL
- **Other:** Docker for containerization

Features and Endpoints

Account Management

- **Create Account/Profile:** Already implemented (`register_api`).
- **Delete Account/Profile:** (`user_delete_api`).
- **Modify Account/Profile:** Update account details (`modify_user`).
- **Login/Logout:** (`login_api`, `user_logout`).
- **Security:** Added password validation and token-based authentication.

Collection Management

- **View Gacha Collection:** Endpoint to retrieve a player's collection (`GET /collection/`).
- **Detailed Gacha Info:** Retrieve details of specific gacha items (`GET /collection/<duck_id>/`).
- **System Gacha Collection:** View all available system gacha items (`GET /system-gacha/`).
- **Detailed System Gacha Info:** Retrieve detailed system gacha information (`GET /system-gacha/<duck_id>/`).

Currency Management

- **Use In-Game Currency to Roll Gacha:** (`roll_gacha_api`).
- **Buy In-Game Currency:** New endpoint (`POST /currency/buy/`).
- **Secure Transactions:** Enhanced validation for currency accuracy.

Auction Market Management

- **View Auction Market:** Retrieves all active auctions (`home_api`).
- **Set Auction for Gacha:** (`spin_duck_api`).
- **Bid for Gacha:** (`place_bid_api`).
- **Transaction History:** New endpoint (`GET /transactions/`).
- **Secure Auctions:** Validations to prevent tampering.

Security Enhancements

- Restricted sensitive actions to "Admin" group users using `django.contrib.auth.models.Group`.
- Applied robust permissions to segregate user roles and prevent unauthorized actions.

Deployment Instructions

This section describes the process of building and launching the project, which is implemented using a microservice architecture with Docker Compose.

Database Migrations

```
docker-compose exec auctionservice python manage.py migrate
docker-compose exec duckservice python manage.py migrate
docker-compose exec playerservice python manage.py migrate
docker-compose exec userservice python manage.py migrate
```

Docker Setup

```
# Build and run services
docker-compose up --build

# Remove orphan containers
docker-compose down --volumes --remove-orphans

# Access service container
docker-compose exec SERVICE_NAME bash

# Migrate main database
python manage.py migrate

# Create a superuser
python manage.py createsuperuser
```

Future Work

- Enhance UI/UX for users.