q

Airell Rasendriya Bachtiar

airell.bachtiar@student.fontys.nl

Kasintu

Web Collection Game

Applied Research Document

Semester 3 - Individual Project

Table of Contents

[**1.** **Version** 2](#_Toc99035617)

[**2.** **Introduction** 3](#_Toc99035618)

[2.1. Purpose 3](#_Toc99035619)

[2.2. Definitions, Acronyms, and Abbreviations 3](#_Toc99035620)

[**3.** **System Overview** 3](#_Toc99035621)

[3.1. Description 3](#_Toc99035622)

[3.2. Main User Activities 3](#_Toc99035623)

[3.3. Project Goal 3](#_Toc99035624)

[**4.** **Problem** 3](#_Toc99035625)

[**5.** **Main Question** 4](#_Toc99035626)

[**6.** **Sub Question** 4](#_Toc99035627)

[**7.** **Methodology** 4](#_Toc99035628)

[**8.** **Answer** 4](#_Toc99035629)

[**9.** **Conclusion** 4](#_Toc99035630)

[**10.** **References** 4](#_Toc99035631)

# **Version**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 0.1 | 25-03-2022 | Make research document |
| 0.2 | 14-04-2022 | Update research document based on feedback |
| 0.3 | 13-05-2022 |  |
| 0.4 | 3-06-2022 |  |
| 1.0 | 17-06-2022 |  |

# **Introduction**

## Purpose

The purpose of this document is to provide detailed architecture design of a game called Kasintu for individual project at Fontys University of Applied Science.

## Definitions, Acronyms, and Abbreviations

* Gacha: A method inspired by toy vending machine where you can get a toy randomly from what the vending machine provide. Instead of toy vending machine, here it is turned into an application game where you can get an item, in this project we called a creature, randomly with a set number of chances.
* Summon or Pull: The action performed when you are getting a creature from the gacha.
* Banner: The place where you summon or pull creature. Banner contains a list of creatures in which the player can obtained and a chance or percentage of how many chances you can obtain a specific creature.

# **System Overview**

## Description

This game is called Kasintu which means bird. Kasintu is a collection-based game where player can collect as much as they want. What they will collect is a different type of birds that is real and fictional thus the meaning of Kasintu is bird, a game where you collect birds. From now on these birds will be called creature.

## Main User Activities

The main feature of this game is called a gacha system. Gacha system is where player can get a chance to receive a virtual item using in game currency. This is where player mainly get a new creature that will be release or has been released by the developer. They called this action of obtaining new creature as a summoning or pulling. In this case we will call this action as summon or summoning. As the where they summoning these creatures is called a banner. A banner contains a certain amount or all the creature available that can be obtained by the player who summoned on that banner.

For a future feature, Kasintu will also include a marketplace and breeding system. Marketplace is where player can but, sell or trade creatures from the other player. Breeding system is where player can breed their own creature to make new creature which may become rarer that the previous creature.

## Project Goal

The goal of this project is to have a game that will entertain user by collecting creatures and to collect everything the game provides. For better user experience, this game will have to has a fast user interface to make user does not need to wait long in between action or input and a secure application so that data from user cannot be tracked or stolen by a third party.

# **Problem**

This game has 2 different kinds of user, player, and admin. Player has access to certain number of features and admin has additional features where for example they can add new creature. We must differentiate between player and admin when using the web application.

# **Main Question**

How can each user access their own features?

# **Sub Question**

1. What is the different between player account and admin account?
2. How one can sign up or make admin account?
3. How does the system recognize between player and admin?
4. Can admin play the game using admin account?
5. How to prevent someone accessing an admin account?

# **Methodology**

Method that suited best for answering these questions is literature study. It is very easy to do and there are already a lot of information regarding these questions. We are using the internet to implement this method. When making an application, these questions have been encountered by many people thus it has a lot of ways to solve this problem. We need to find what is the best and most optimised solution for this problem.

# **Answer**

1. The app will have 2 main interfaces, interface for admin and for player. Admin can configure creatures and banners. Meanwhile, player can play the game such as collecting creatures.
2. There will only be one admin account for an easy control on who can access the admin account.
3. Authorization is implemented in the system using Spring Boot and OAuth2. This application use Spring Boot for the back end for easier coding where basic things have been set up. OAuth2 can help with the security of the application that helps recognize account whether it is admin account or player account. OAuth2 use an external authentication for more secure way to access this web application. In this project we will use GitHub authentication to access admin account. We only need 1 GitHub account to use authenticate an admin account.
4. Admin cannot play the game; admin account’s purpose is only to add or edit creatures and/or add or edit banners. Admin can only add stuff based on a pre-configures settings provided from the app. Editing beyond that should be done in code.
5. From using an account that is authenticated by third party and it is not present in the web application’s database, it is harder to access the admin account.

# **Conclusion**

OAuth2 is the best choice for authentication a single account. Since it is quite popular, a lot of information are available on the internet. It uses trusted third party for its authentication, so it is easier to set up and safe to use.

# **References**

<https://spring.io/guides/tutorials/spring-boot-oauth2/>

<https://fhict.instructure.com/courses/12189/pages/authentication-and-authorization>