

Design Document

Online Casino djigitBET

by Radoslav Radev

ICT & Software Engineering semester 3

9/10/2022

ver. 2.0

Document Versioning

Date	14/09/2022
Version	V1
State	Defined (Initial Draft)
Author	Radoslav Radev

Date	07/10/2022
Version	V2
State	SemiComplete
Author	Radoslav Radev

Contents

Document Versioning	1
Introduction	3
General Information	3
Architectural Approach	3
Project Methodology	3
Frameworks	4
Feature Prioritization	4
Quality Assurance	4
User Stories	5
Technical Architecture	10
API URL tables	10
How is SOLID guaranteed?	11
Framework selection:	11
C4 Diagram	12

1. Introduction

1.1 General Information

The application's technical and architectural components are the primary emphasis of this Design paper (online Casino).

The document contains things such as:

- *User stories*
- *RESTful API design tables*

1.2 Architectural Approach

Object-oriented design based on SOLID principles will establish a Backend as a Service application, fully separated from the front end

1.3 Project Methodology

The Methodology used in this project is **Agile**.

Agile methodology:

“a software development and project management method that is continuous, and aids teams in delivering value to clients more quickly and with fewer hassles”

1.4 Frameworks

Frameworks used in this project are as follows:

- **Java & Spring Boot** - RestAPI
- **JPA - ORM**
- **MySQL** - database
- **React & MUI** - front end

Project is going to be continuously deployed using **CI Pipelines** and **Docker Compose**.

1.5 Feature Prioritization

To understand which features are more important than others, a prioritization system is implemented/included.

It goes as followed.

- **MUST** – *“Feature must be implemented. It has the highest priority.”*
- **SHOULD** – *“Feature should be implemented if everything goes smoothly and by schedule. It has medium priority.”*
- **COULD** – *“Feature could be implemented if there is time upon finishing the project. It has lowest priority.”*

1.6 Quality Assurance

Quality of the project is going to be assured through test-driven development as follows:

- *Unit tests*
- *Integration tests*
- *End-to-end tests*

2. User Stories

A user story is a comprehensive explanation of a software feature written from the viewpoint of the client or end user.

User story 1

Title		Priority	Estimate
Register		MUST	sprint 2
As a customer, I want to Register at the website. So that I can gamble.			
Acceptance Criteria			
Given provided information, When the Register button, in the Registration component of the website, is pressed The system: <ul style="list-style-type: none">- Creates an account- Redirects to the home page in case of success No/Insufficient information entered:			

- error message displaying which data is field incorrectly

User story II

Title	Priority	Estimate
Log-In	MUST	sprin2
As a customer, I want to Log-In to the website So that I can gamble.		
Acceptance Criteria		
Given provided information, When the Log-In button, in the Login component of the website, is pressed The system, depending on the type of the account: <ul style="list-style-type: none">- Logs in the user and redirects to the home page in case of success No/Insufficient information entered: <ul style="list-style-type: none">- display and error message		

User story III

Title	Priority	Estimate
Profile update	MUST	sprint 4
As a customer, I want to update my Account information, So that my information stays safe, secure, and up to date.		
Acceptance Criteria		

**Given provided information,
When the Update button, in the Profile component of the website, is pressed
The system:**

- Updates the information in case of success

Insufficient information entered:

- error message displaying which fields are incorrerct

User story IV

Title	Priority	Estimate
Account deletion	Should	sprint 5
As a customer, I want to delete my account, Because I do not need the services of the online casino anymore.		
Acceptance Criteria		
When the Delete my Account button, in the Profile component of the website, is pressed The system: <ul style="list-style-type: none">- Deletes the account after a confirmation pop-up		

User story V

Title	Priority	Estimate
Add funds via a payment system	Could	sprint 6
As a customer, I want to add funds to my Account So that I can gamble them on the slot-machine.		

Acceptance Criteria

When the Amount is specified and confirmed by a button, in the Funds component the system redirects to a payment provider and confirms the transaction

The transaction failed:

- funds are not added to the account

User story VI

Tittle	Priority	Estimate
Adding the course to the cart	MUST	sprint 4
As a customer, I want to try my luck on the slot-machine		
Acceptance Criteria		
When the spin button is pressed on the slots component of the website:		
The algorithm decides on giving a prize to the customer,		
<ul style="list-style-type: none">- the amount betted by the user is substracted from thair account- if a prize is won the funds will be added to the customer`s account		

User story VII

Tittle	Priority	Estimate
Modify User Account	MUST	sprint 2
As an Administrator, I want to Update Delete and Review all aspects of a user`s account with a usefull search engine		

Acceptance Criteria

When text is entered in the search box of the User_Management Component, the system:

- Refines the presented data according to the search criteria

When the Edit button next to a user account is pressed in the User-Management Component, the system:

- presents the user with a form in which he can modify the information and confirm it

When the Delete button next to a user account is pressed in the User-Management Component, the system:

- the user can confirm the deletion of the account

User story VIII

Title	Priority	Estimate
view Statistics	MUST	4
As an Administrator, I want to view statistics on profit and winCoefficient by user bases, an overall win rate of the slot machine		
Acceptance Criteria		
The system presents the required statistic data with filter options in the Statistics component of the website		

3. Technical Architecture

3.1 API URL tables

This section of the paper contains API tables per user story.

URL	Resource	Operations	Description
/users/	User	GET	get a collection of all users
/users/\$id	User	GET	get the specified user
/users/	User	POST	add a new user account to the system
/users/\$id	User	PUT	Update the contents of the account object
/users/\$id	User	DELETE	Delete the selected account
/authenticate/login	User	POST	Login,generate JWT
/authenticate/register	User	POST	Create a new Player account
/unprivilegeduser/	User	PUT	get the user who is logged in this unprivileged account

3.2 How is SOLID guaranteed?

- S– Every class has only one responsibility.
- O– Classes are open for extension but closed for modification
- L– If class A is a subtype of class B, we should be able to replace B with A
- I– Proper use of interfaces and abstract classes
- D– Dependency injection and inversion are by the Spring Bean system

3.3 Framework selection:

The frameworks used in this project are as followed:

- Java with Spring Boot and JPA, is used for the Rest API,
- JPA ORM is used to manage the database
- MySQL database
- React is used for the presentation layer of the application (FE)

Why did I use these technologies?

React - is the most popular front-end framework and it is recommended as the first one to learn, although the industry is moving towards Svelte it's not recommended for beginners

Java and spring boot - it is a main requirement of the curriculum.

- I would have preferred to do the restAPI service with ASP.NET WebAPIs in the dot.net ecosystem, with the comfort of modern developer features, more stable code, and overall ease of use and support because:

- As one of the main high-level OOD-oriented languages, java is missing a lot of features
 - Disadvantages: poor garbage collector, no null safety, less maintained
 - Main features I miss from dot.net: parallel iteration, Span, LINQ, primitive type integration (e.g String), modern syntax

- Spring-Boot encourages taking the control out of the developer and leaving it up to chance in a lot of cases, but on the opposite side requires a large amount of configuration code,

- documentation, is poor and version mismatched,

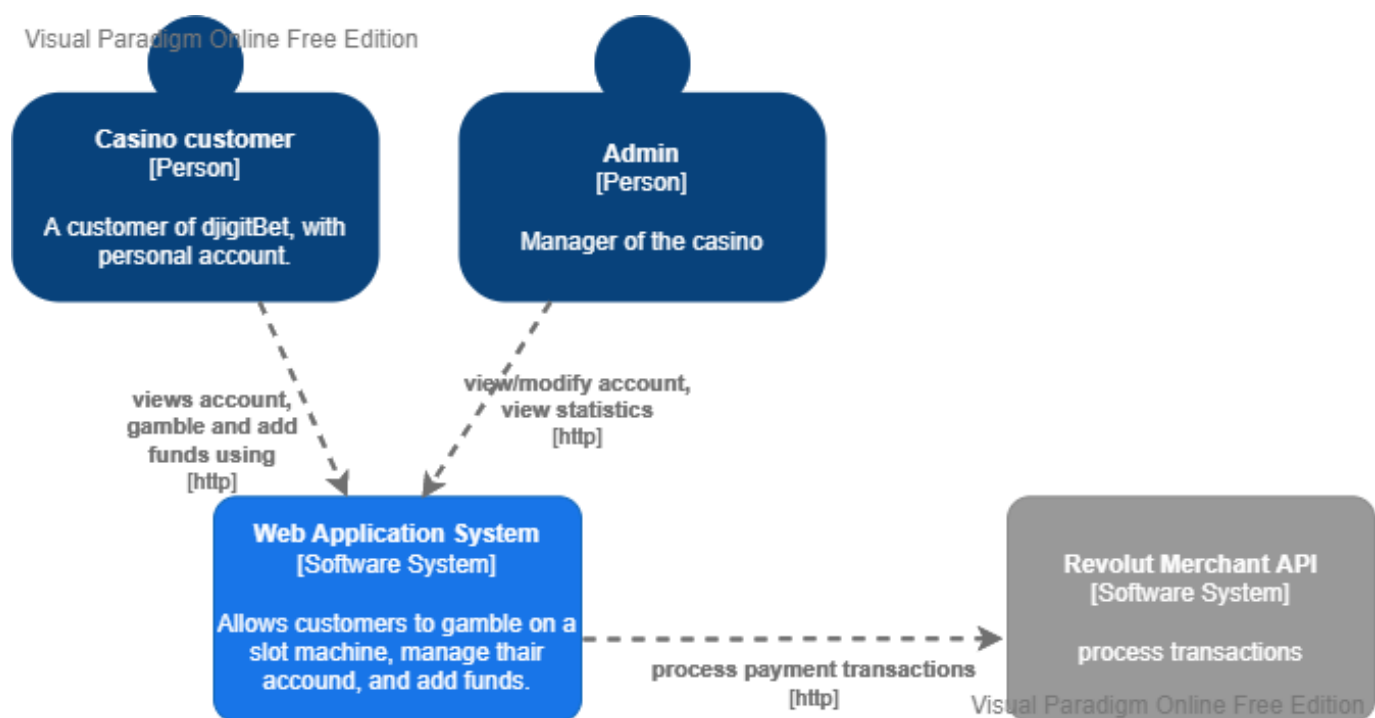
- SpringBoot it's hard to debug and produces a lot of bugs

- JPA-hibernate lacs features in comparison with Entity Framework

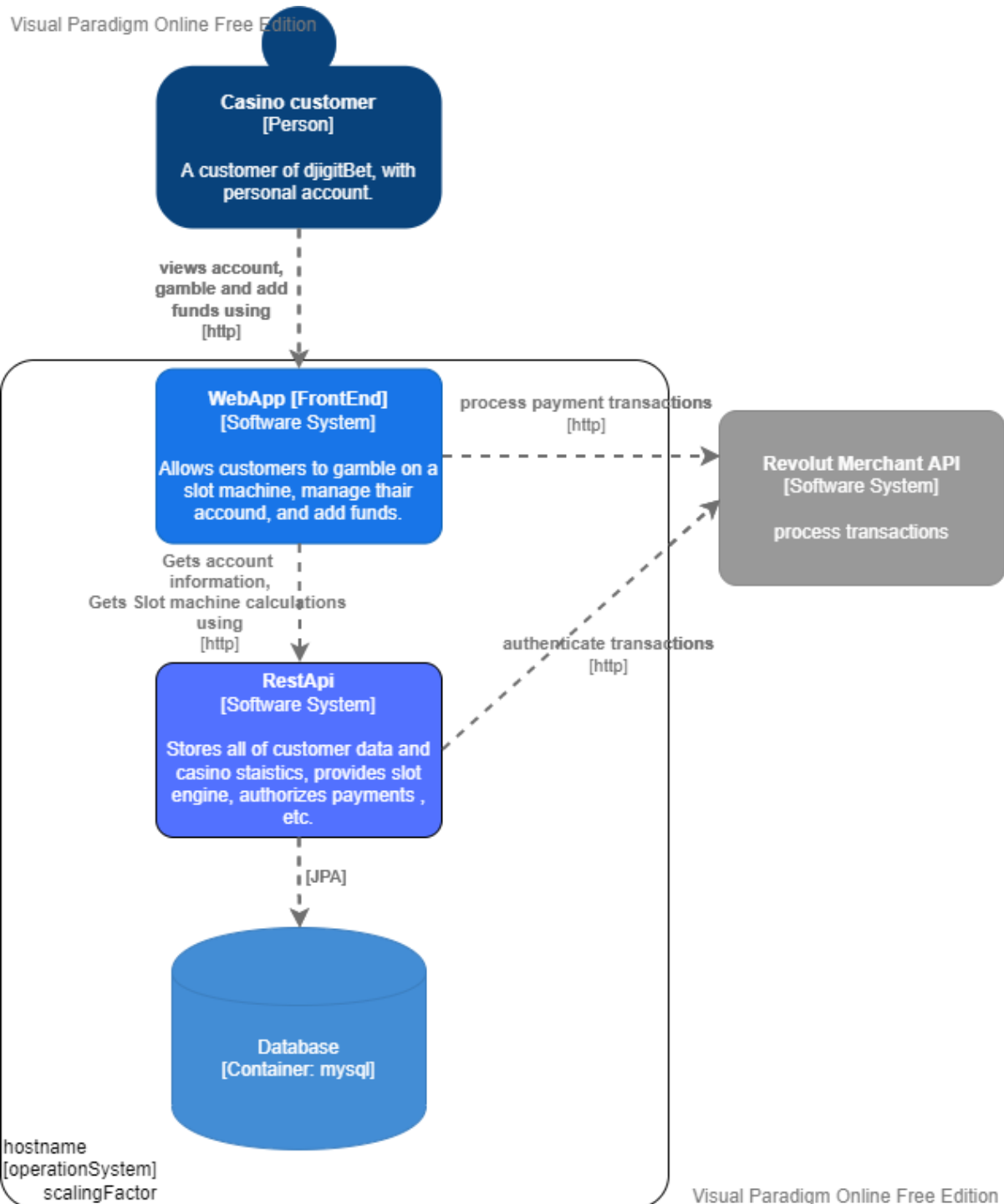
- dot.net by default uses Internal-Language compiler which is similar to the java VM, but much more flexible and faster. Furthermore, dot.net introduced Native AOT(ahead-of-time) compilation which uses even less memory and it's even faster (compiles to native machine code for the selected architecture)

3.4 C4 Diagram

3.4.1 C1-Context

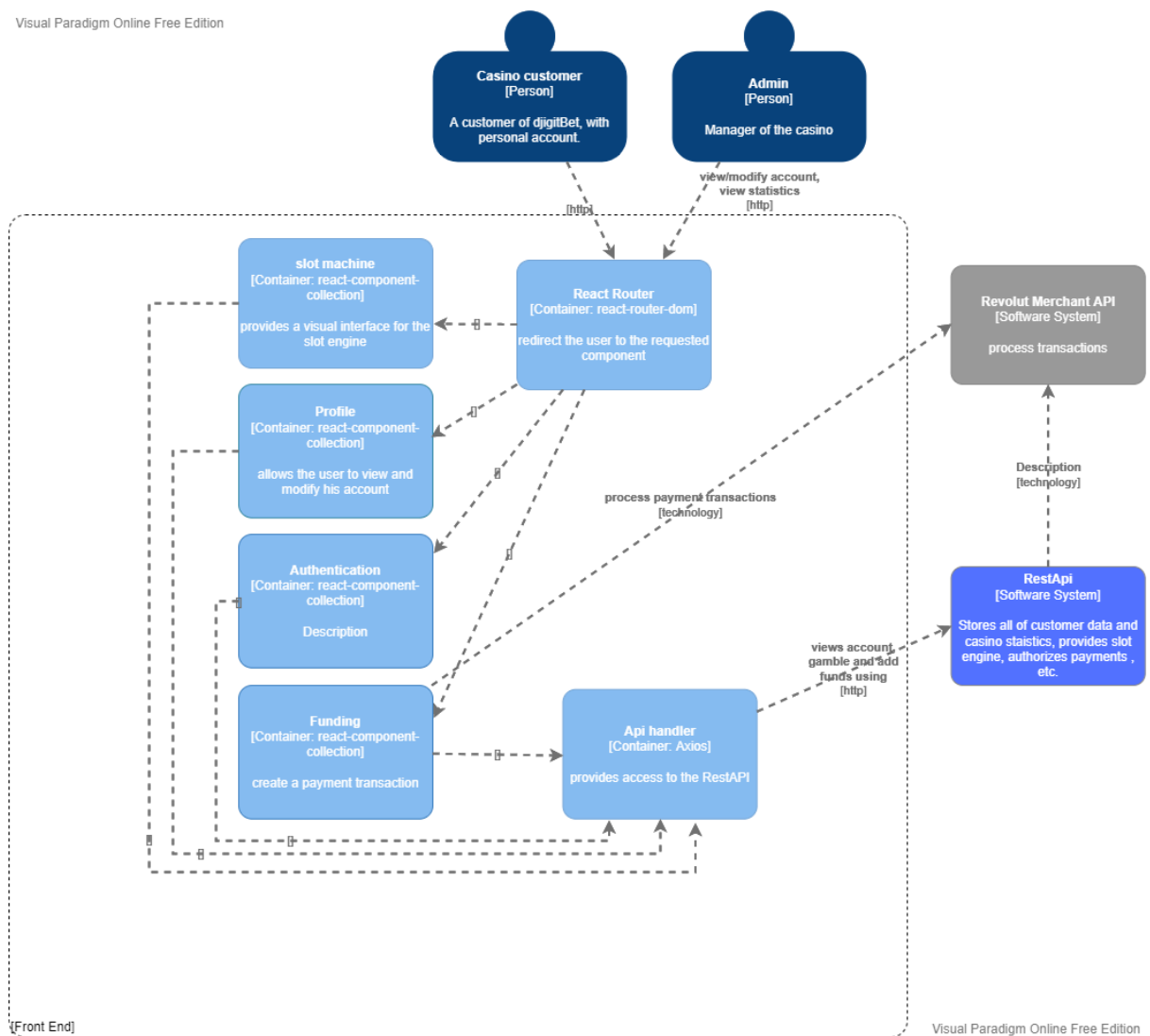


3.4.2 C2-Context



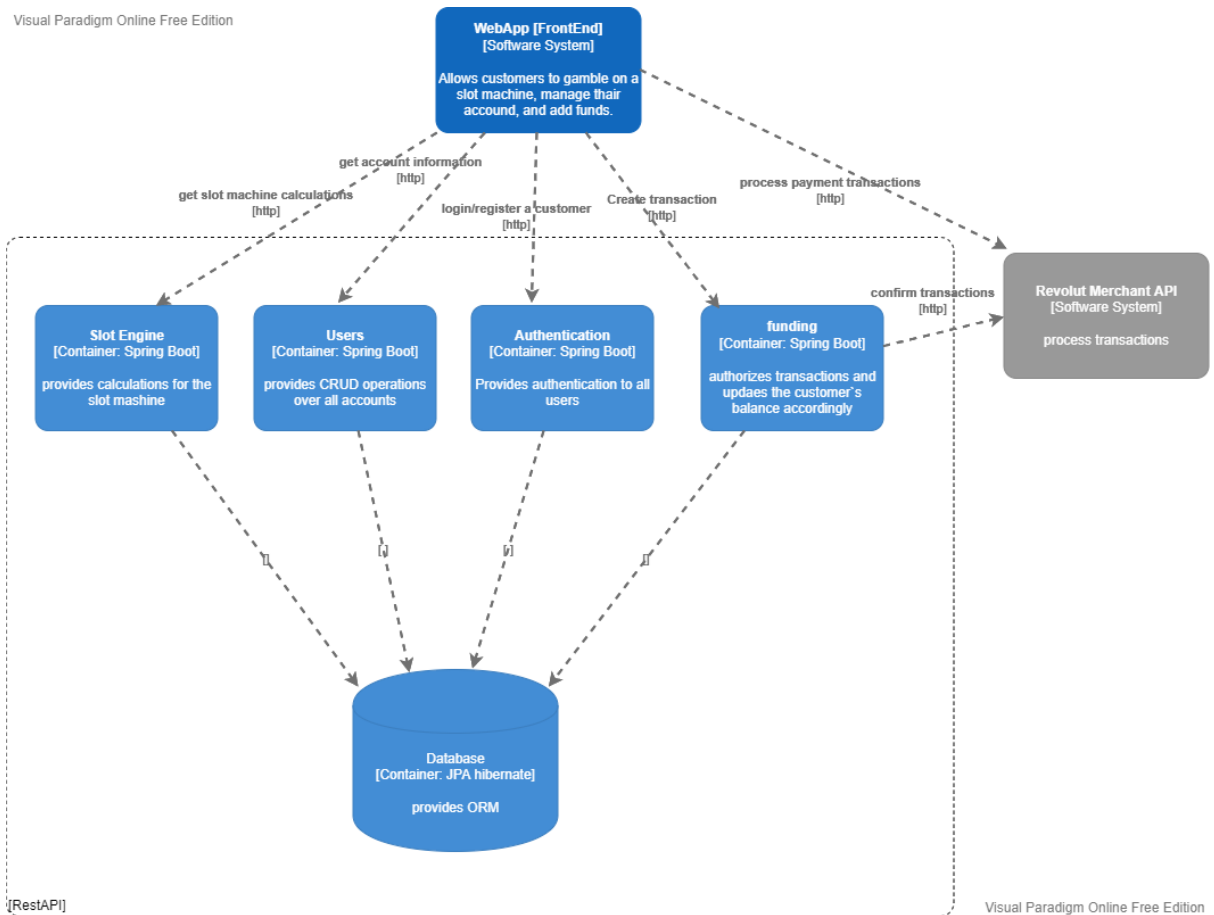
3.4.3 Zoom into the WebApp[Front End] [Software System]

Visual Paradigm Online Free Edition

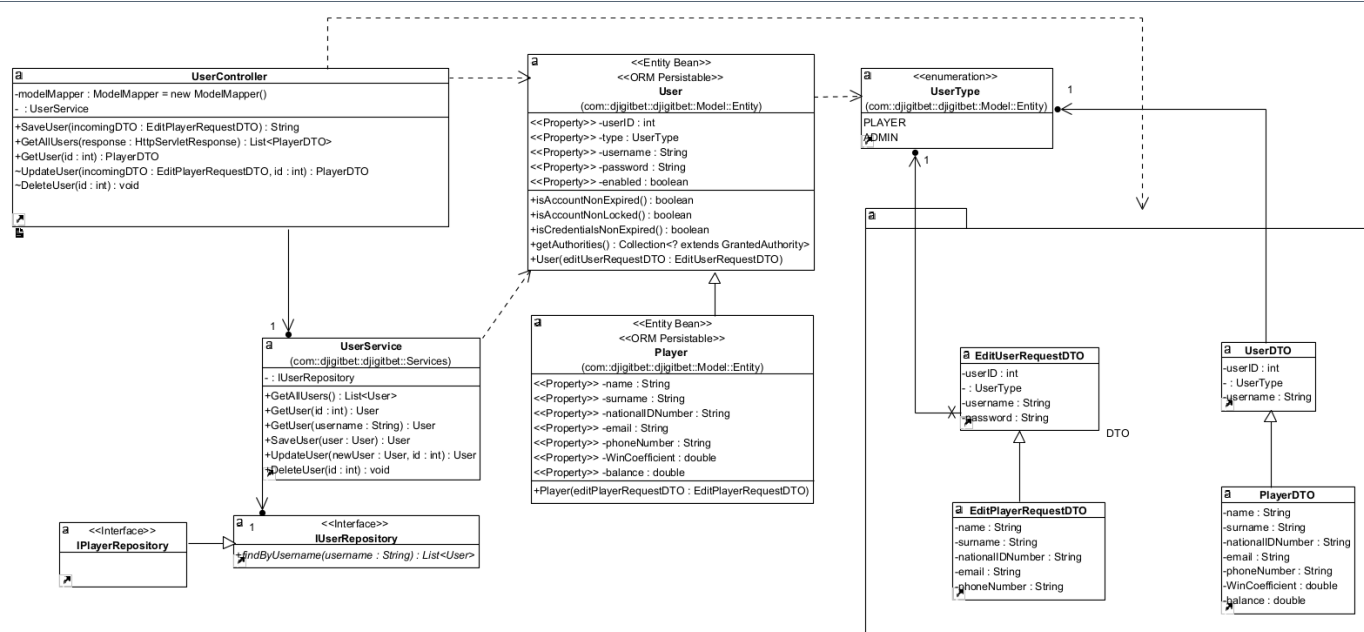


Visual Paradigm Online Free Edition

3.4.4 Zoom into the RestAPI[Software System]



3.4.5 Zoom into the User[Container]



3.4.6 Combined diagram of the authentication[Container]

