Playvera - C4 Diagram document

By: Ivan Ovcharov, 4090993

What is a C4 diagram?

A C4 Model describes or defines software architecture at different levels of detail. A C4 Model is a set of diagrams representing the context, containers, components, and code of a piece of software. The hierarchy of these diagrams allows different audiences to understand the software architecture at the level of detail they need.

What are the four levels of a C4 diagram?

Level 1: Context Diagrams

Context diagrams are the most general description of what your system does, who will use it, and what other systems it will interact with. A context diagram will help you describe the scope of your project and help you pinpoint who the user is and what problem you're going to solve.

Level 2: Container Diagrams

The container diagram takes the first step into describing the software system and shows the APIs, applications, databases, and microservices that the system will use. Each of these applications or services is represented with a container and the interactions between them are shown at a high level.

Level 3: Component Diagrams

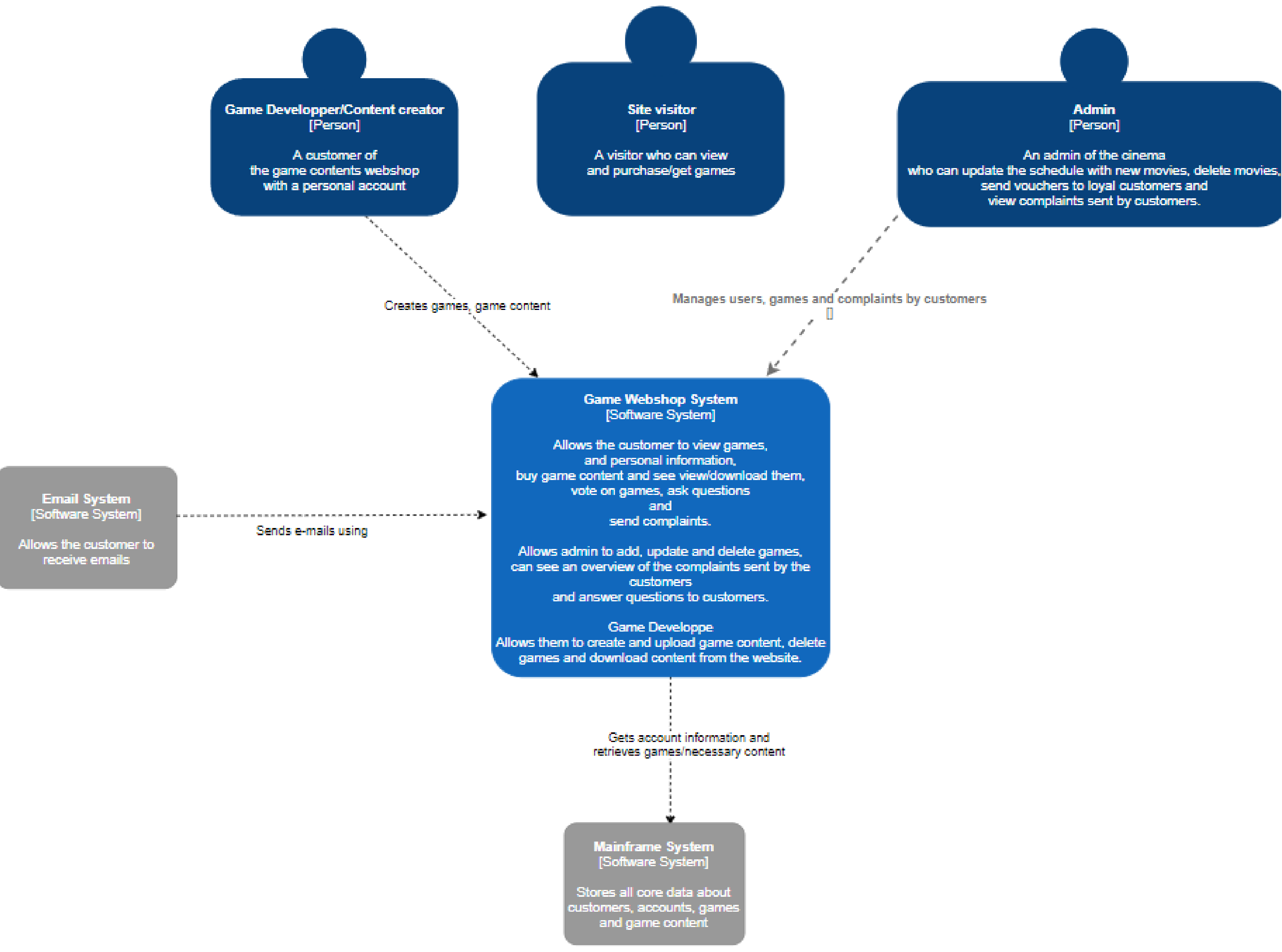
One step deeper than the container diagram, the component diagram details groups of code within a single container. These components represent abstractions of your codebase.
This diagram type is comparable to a UML component diagram, but follows a less-strict set of "rules" in order to create the software architecture diagram.

Level 4: Representing Code with Class Diagrams

The last level requires lots of detail to show how the code of a single component is actually implemented. To do this, you would want to make a UML class diagram or entity relationship diagram that describes the component.

Reference: https://www.gliffy.com/blog/c4-model

Level 1:
n Context (C1)

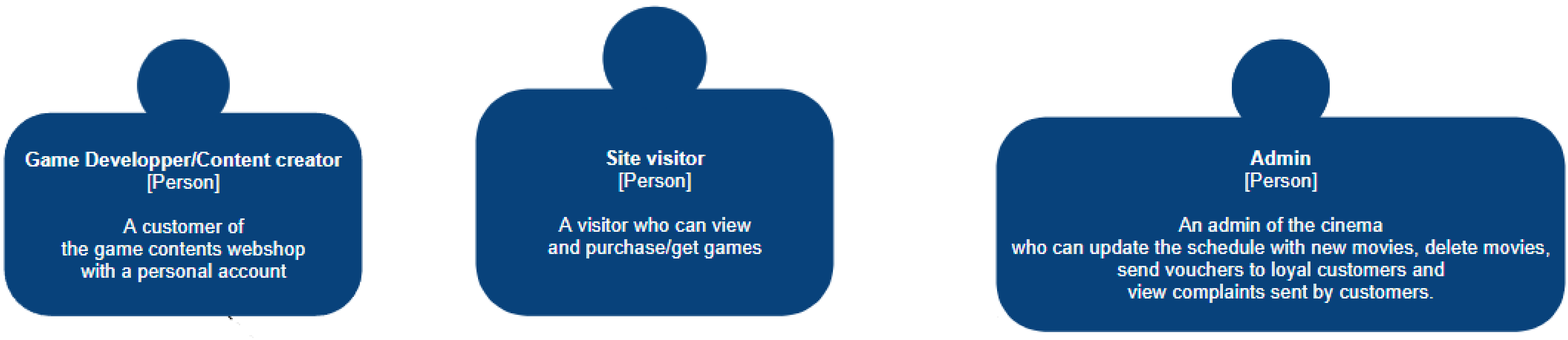

Game Developper/Content creator
[Person]
A customer of the game contents webshop with a personal account
Site visitor
[Person]
A visitor who can view and purchase/get games
Admin
[Person]
An admin of the cinema who can update the schedule with new movies, delete movies, send vouchers to loyal customers and view complaints sent by customers.
Creates games, game content
Manages users, games and complaints by customers
Game Webshop System
[Software System]
Allows the customer to view games, and personal information, buy game content and see view/download them, vote on games, ask questions and send complaints.
Allows admin to add, update and delete games, can see an overview of the complaints sent by the customers and answer questions to customers.
Game Developpe Allows them to create and upload game content, delete games and download content from the website.
Email System
[Software System]
Allows the customer to receive emails
Sends e-mails using
Gets account information and retrieves games/necessary content
Mainframe System
[Software System]
Stores all core data about customers, accounts, games and game content

The above shown diagram reflects the entirety of the first level of the C4 diagram for Playvera webshop.

Level 1 Component breakdown:

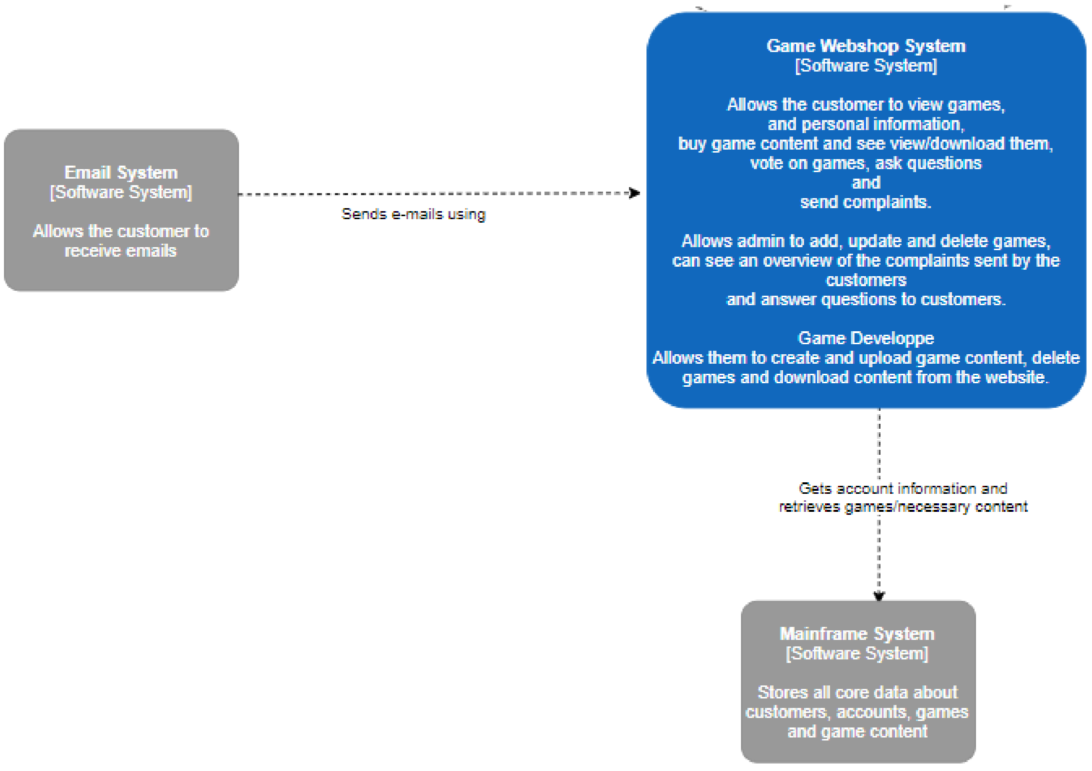
Website users and administrators

- The website consists of 3 different users: Site visitor, administrator and a game/content creator
- All users have different type of access in the website

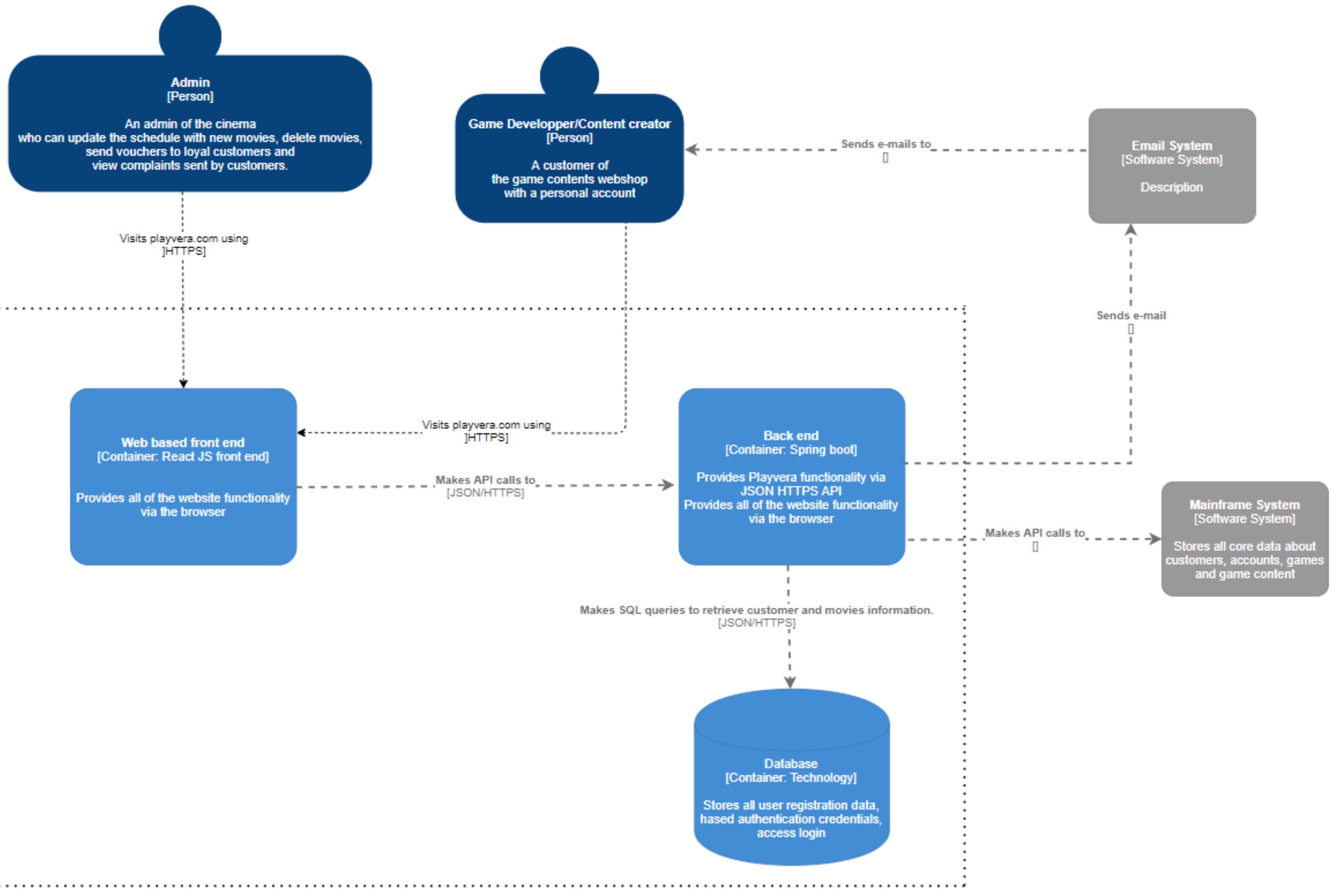

Game Developper/Content creator
[Person]
A customer of the game contents webshop with a personal account
Site visitor
[Person]
A visitor who can view and purchase/get games
Admin
[Person]
An admin of the cinema who can update the schedule with new movies, delete movies, send vouchers to loyal customers and view complaints sent by customers.

Game webshop system breakdown:

- The main part of the software system allows the user to view games, personal information, upload/download content and ask complaints
- Administrators can manage users and content that is uploaded in the website
- The software also includes an email system


Email System
[Software System]
Allows the customer to receive emails
Sends e-mails using
Game Webshop System
[Software System]
Allows the customer to view games, and personal information, buy game content and see view/download them, vote on games, ask questions and send complaints.
Allows admin to add, update and delete games, can see an overview of the complaints sent by the customers and answer questions to customers.
Game Developpe
Allows them to create and upload game content, delete games and download content from the website.
Gets account information and retrieves games/necessary content
Mainframe System
[Software System]
Stores all core data about customers, accounts, games and game content

Level 2:

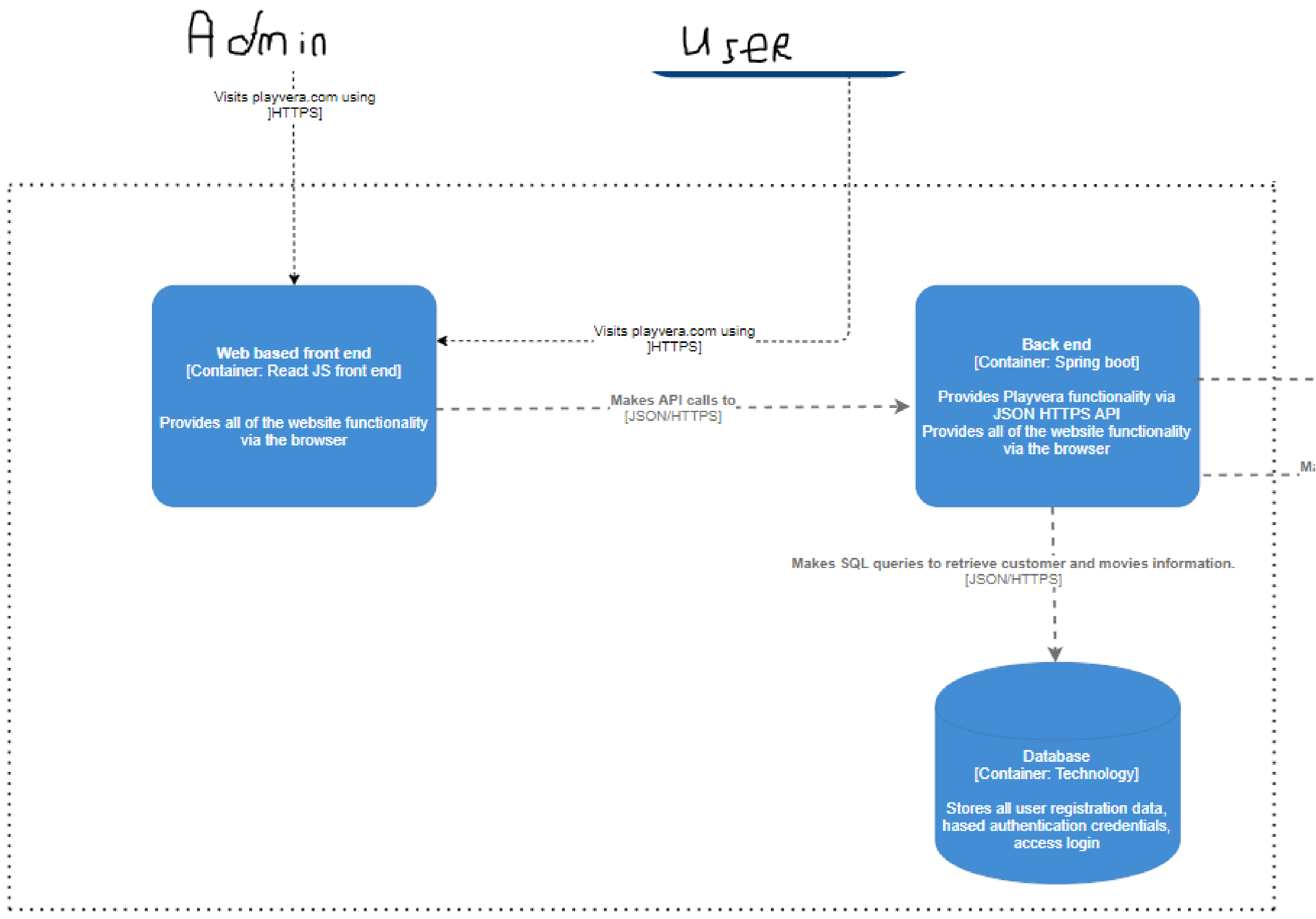

Admin
[Person]
An admin of the cinema
who can update the schedule with new movies, delete movies,
send vouchers to loyal customers and
view complaints sent by customers.
Game Developper/Content creator
[Person]
A customer of
the game contents webshop
with a personal account
Sends e-mails to
Email System
[Software System]
Description
Visits playvera.com using
]HTTPS]
Sends e-mail
Web based front end
[Container: React JS front end]
Provides all of the website functionality
via the browser
Visits playvera.com using
]HTTPS]
Makes API calls to
[JSON/HTTPS]
Back end
[Container: Spring boot]
Provides Playvera functionality via
JSON HTTPS API
Provides all of the website functionality
via the browser
Makes API calls to
Mainframe System
[Software System]
Stores all core data about
customers, accounts, games
and game content
Makes SQL queries to retrieve customer and movies information.
[JSON/HTTPS]
Database
[Container: Technology]
Stores all user registration data,
hased authentication credentials,
access login

Level 2 of the diagram represents the connection between the users and systems and the back/front end with the database entity.

Level 2 Component breakdown:

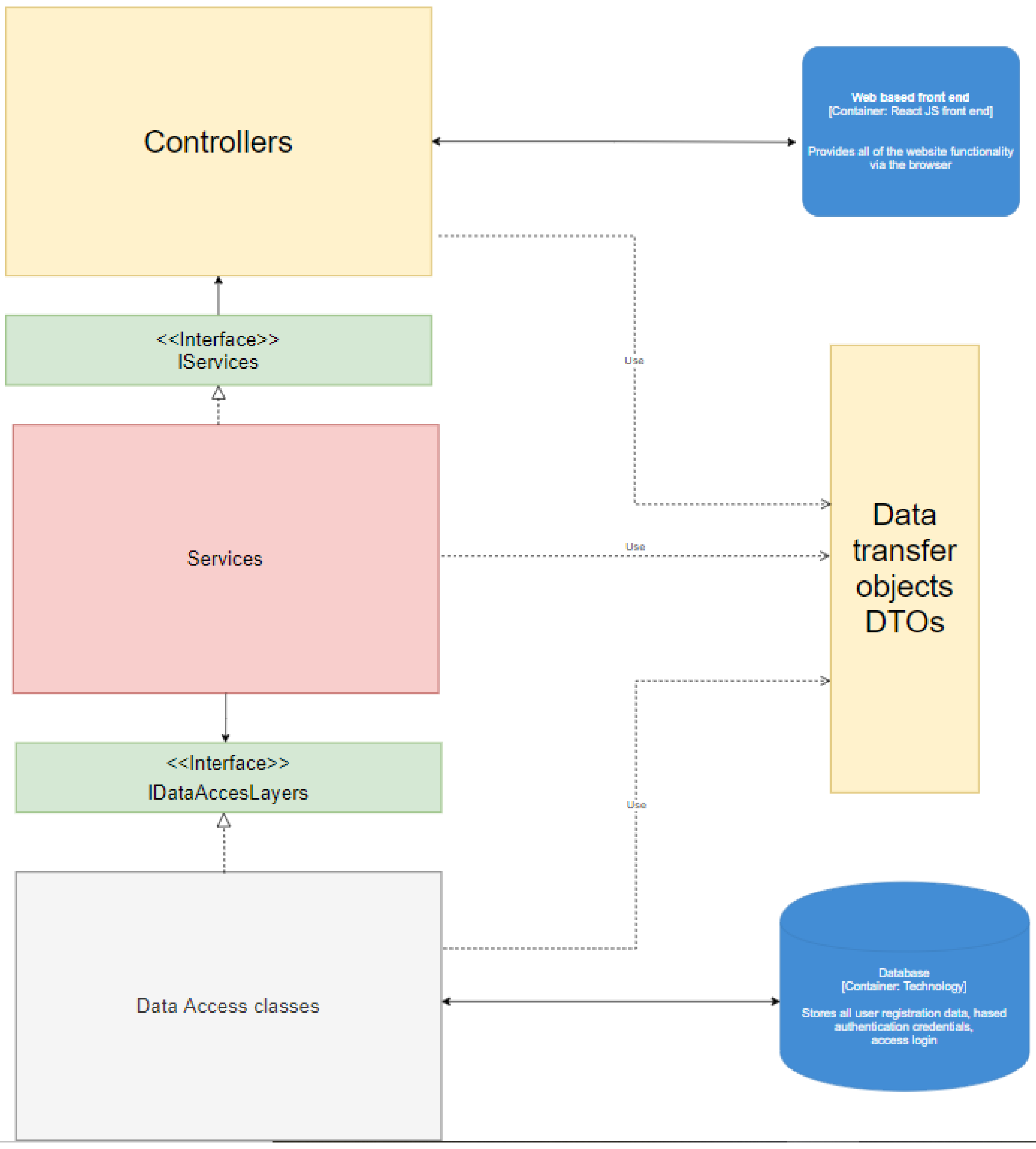
Backend and front end connection between users and administrators:

- REACT front end - it provides the UI and all of the website functionality within a browser environment. Users and administrators use the interface to interact with the back-end of the application
- Back end - it provides all of the data and logical functionality of the website using springboot and Json https api. It makes the front-end functionality possible.
- Database layer - it is used to store all user/content information and directly interacts with the back-end when a request for specific information is created (retrieving/pushing/updating/deleting information).



Admin
User
Visits playvera.com using]HTTPS]
Visits playvera.com using]HTTPS]
Web based front end [Container: React JS front end]
Provides all of the website functionality via the browser
Makes API calls to [JSON/HTTPS]
Back end [Container: Spring boot]
Provides Playvera functionality via JSON HTTPS API Provides all of the website functionality via the browser
Makes SQL queries to retrieve customer and movies information. [JSON/HTTPS]
Database [Container: Technology]
Stores all user registration data, hased authentication credentials, access login
Ma

Level 3:


Controllers
<<Interface>>
IServices
Services
<<Interface>>
IDataAccesLayers
Data Access classes
Web based front end
[Container: React JS front end]
Provides all of the website functionality via the browser
Use
Use
Use
Data transfer objects DTOs
Database
[Container: Technology]
Stores all user registration data, hased authentication credentials, access login

Level 3 of the diagram represents all of the specific components that are used in the functionality of the application.

Level 3 Component breakdown:

Controllers: Controllers are used to connect with the front-end (UI) of the application. The also function with the help of interfaces, allowing them for a flexible connection with the service components.

Service component: Service components provide the logic for the controllers to use. They are directly connected to data access layers and can CRUD different data.

Database: it is used to store all user/content information and directly interacts with the back-end when a request for specific information is created (retrieving/pushing/updating/deleting information)