

**Scalable Services (SEZG583)**

**Assignment 1**

By:

Govind K Rajesh (2020MT93240)

Summaiya Sarvari V K (2020MT93179)

**Objective**

The aim of this project is to develop an application which simulates the working of a game rental store, titled JustGames. The application would be deployed in each store of JustGames, and would communicate with a common database. Users would be able to borrow any game within that specific store, or in case the game they want is not available in that store, they can place an order, which would have that game shipped over from another store in the area. Customers would also have the option of buying games from JustGames, at a discount over the MRP. This project aims to be a highly scalable system for the application of the above use case.

**Design**

This application has 3 functionalities, each of which is in the form of an individual microservice. The functionalities, along with their roles, are described below.

* Borrow: This service allows users to borrow games from the store’s library. Note that this service is restricted only to the store that it is deployed in, which restricts users to only be able to borrow games from that store. This service queries the shared database to retrieve the ID of the game being borrowed, and then changes the status of that game in the back end, marking the game as unavailable for further borrowing.
* Order: This service allows users to order games from other stores in the area. In case the requested game is not found in the current store, this service can be used to check in other stores, and if the game is found, it is then marked as unavailable, and is shipped to the current store.
* Buy: This service is used by customers who seek to buy games from JustGames. As these games are pre-owned, they are given to customers at a discount over MRP. As for how it works, it removes the sold game from the database, making it unavailable for further use.

**Technology Used**

This project is implemented in Python, using the Flask framework for back end and MySQL for the database. HTML is used to obtain a skeletal front end, for demonstration purposes. The separate services communicate by means of REST API, which is provided by Flask.

While we did intend to deploy these services on Docker Containers and then orchestrate them using Kubernetes, we were unable to do so due to time constraints.

**Team Contribution**

The contribution by the individual team members are as follows:

* Summaiya Sarvari V K: Worked on the first module (Borrow) using Python and Flask, as well as the database, using XAMPP. Attempted to get the services running on Docker Containers. Collaborated on the documentation and video.
* Govind K Rajesh: Worked on the second and third modules (Order and Buy) using Python and Flask. Collaborated on the documentation and video.

Github Link:

https://github.com/2020MT93179/2020MT93179\_2020MT93240\_JustGames.git