**CoquiGames**

**Jesmarie Hernandez, Heriberto Bourdon, Abisai Ramos**

**{jesmarie.hernandez, heriberton.bourdon1, abisai.ramos}upr.edu**

**1. Introduction**

There is both a lack of competition in the video game store market, and in the options for customers to purchase their favorite video games and accessories, particularly in the island of Puerto Rico. In the island there is an absence of stores whose primary selling point is video games. Therefore, the gaming community in Puerto Rico are forced to choose between buying their video games online or in the island’s chain stores, which don’t offer a variety in games and prices. To solve this problem, we propose a video game web store, **CoquiGames**. **CoquiGames** is a web store that offers a viable solution to the gaming community in Puerto Rico by offering weekly deals, variety in titles available, optional payment methods etc.

**CoquiGames** is implemented as a responsive web application; it is designed to be viewed appropriately as desktop and as a mobile website. As a responsive web application, clients who either own a phone or a computer will be able to access it without the need of downloading and installing an application.

**CoquiGames** will be implemented using the following technologies

Client Side Tools: AngularJS, HTML/CSS.

Server Side Tools: Flask, PostgreSQL.

**2. Client App Description**

**CoquiGames** was implemented using HTML5, CSS, JavaScript, Bootstrap, and AngularJS. This application has two kinds of users the regular users and the administrator user. The regular users are persons that can browse the website for merchandise, view a specific product’s details, buy products, add them to a wish list, store their personal information, etc. Whereas an administrator users are the IT personnel in charge of supporting and maintaining the database.

A regular user is able to perform the following tasks: register, login, view, and modify their account and personal information, shipping and billing addresses, and the payment methods. The account information consists of: a unique username, and a password. Whereas personal information will have the customer’s full name, email, date of birth and phone number. A customer without an account (guest user) can only browse and view items, and register to become an account user and be granted buying privileges.

Users will be able to browse the following categories of products within their specific platforms: video games, consoles, and accessories. Additionally, products will be able to be browsed through genre tags, a means to search for a product with keywords, and an option to allow for products to be filtered to appear in increasing or decreasing order of price.

Users are able to view the details of a product. Details will consist of the product’s name, genre, release date, product code, availability, instant price, description, photo of the product, and ratings. Additionally, there will be means to place, view and update the shopping cart. Once an order has been placed, the customer will get an invoice of the order that has been placed by email. Also the web app will keep the invoice in the records for future reference. Order status information will be shown in the following format: open, closed, in transit, cancelled, pre-ordered. Finally, to review the products we will use a 5-star rating system with an option to leave a comment.

**3. Server side description**

The store data will be obtained from a Flask server using REST calls and encoding data with JSON. The data will be stored in PostgreSQL (object-relational database system). The data objects are maintained and access in terms of four operations, often described with the acronym CRUD (Create, Read, Update, Delete). For the security and integrity of the store data the functionality will be limited to the user role.

A regular user will be able to perform the following tasks in the server:

* Authenticate a customer account.
* Create\Read\Update a customer account.
* Read products by category.
* Read the details of a product.
* Read\Update a shopping cart.
* Create an order for items in the shopping cart.
* Create\Read\Email an invoice of an order that has been placed.
* Read a Review of a product, only logged user will Create\Update\Delete a review.

An administrator of the **CoquiGames** will be able to:

* Authenticate an administrator account.
* Create\Read\Update\Delete customer and administrator accounts.
* Read sales data.

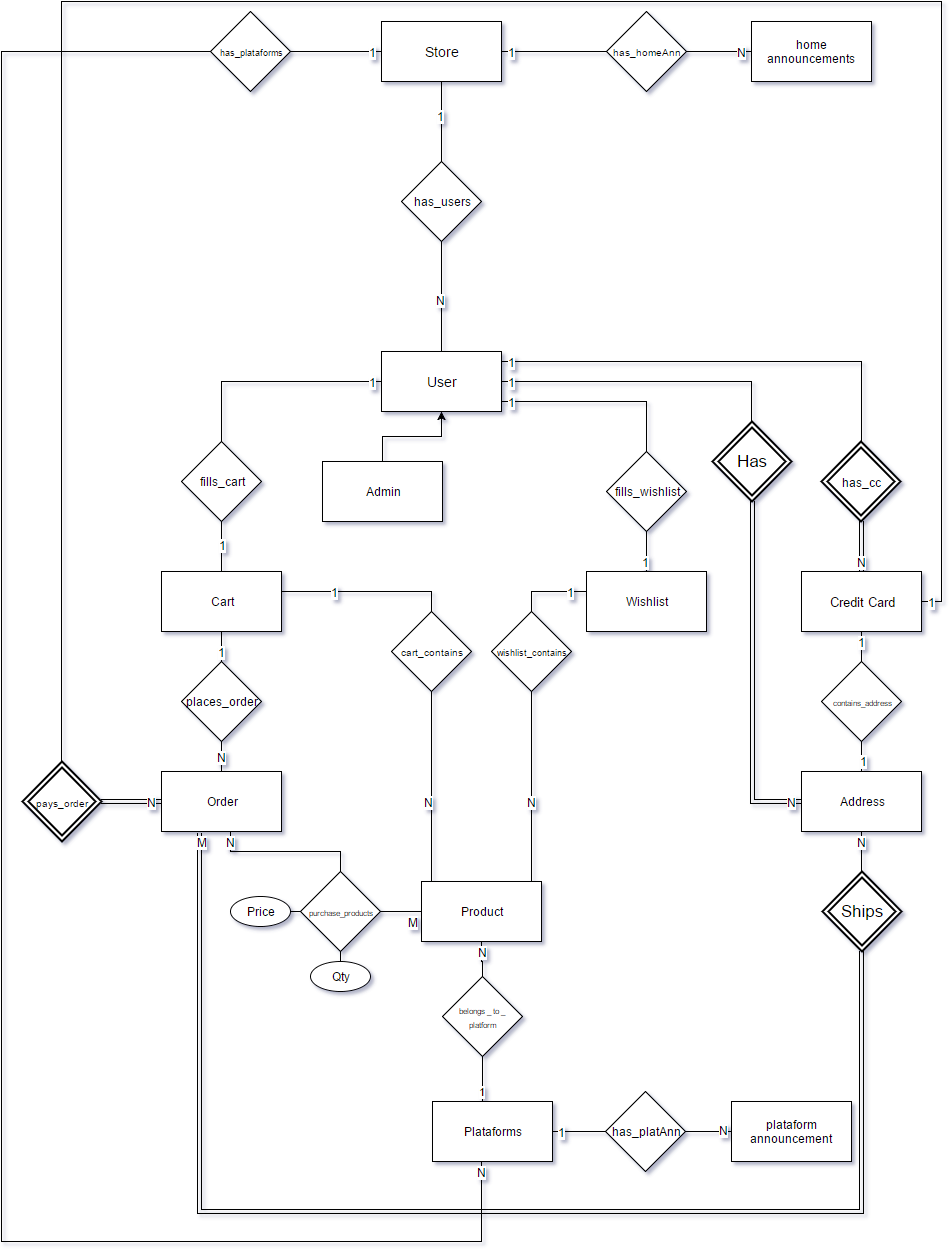
The signup process will be handled by way of a script configured for the application, which connects to the application server, handles the verification process and creates a session in the application server (Flask Server). User sensitive information such as credit numbers must be encrypted, for safety and protection of all the clients. Our front-end (which will be AngularJS) is going to request the back-end (Flask) for an encryption key. The front end will then encrypt the credit card information and then return it to the back end. Finally, the back end will decrypt information and the it will be stored in the database, which will once again encrypt the information.

**Planned Tables:**

|  |  |
| --- | --- |
| *Table* | *Description* |
| User: | Stores the user personal information. |
| User Account: | Holds the information to access and account. |
| User Role: | Has the account role and if its active. |
| Address: | Keeps all of the corresponding fields for an address. |
| Order: | Retains the order customer, date amount, tax. |
| Order Details | Stores the products in an order their quantity and price. |
| Order Status | Holds order status information such as open, cancelled, in transit etc. |
| Wish List | Has the products a user wishes to buy in the near future. |
| Shopping Cart | Keeps the selected products by a customer. |
| Reviews | Retains user reviews. |
| Platform | Stores additional information regarding a platform |
| Game | Holds additional information regarding a game |
| Accessories | Has additional information regarding an accessory. |
| Product Details | Keeps contain the general information of a product such as name, price, type, quantity, etc. |
| Photos | Stores the photo name and location in the server. |

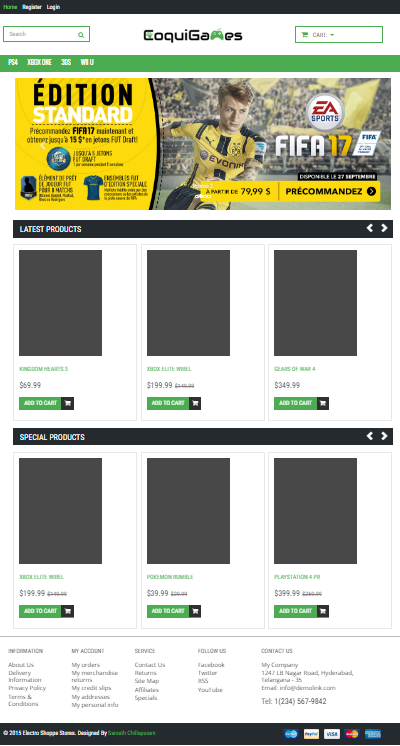
# 

**4. E-R Diagram**

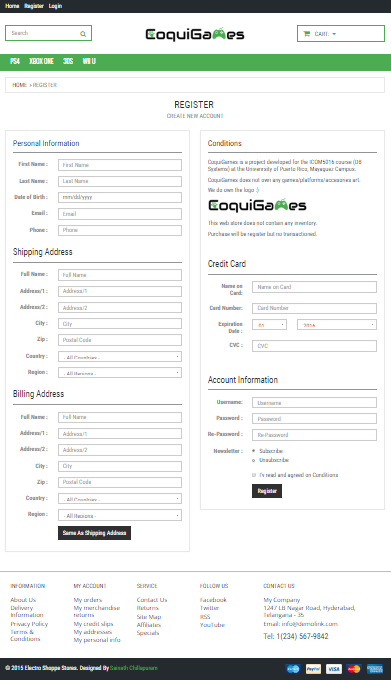


This E-R Diagram represents the Coqui Games Web store. The store entity contains home- announcements and platforms. Each platform contains a distinct set of announcements and catalog of products. The user entity has various relationships with other entity’s, such as: a user contains personal information, for instance their address and credits cards, and a user can place an item into their cart list or the wish list. From the cart list the user may place an order which contains product details of the selected item. That order placed must be payed with a credit card and shipped to the user’s address.

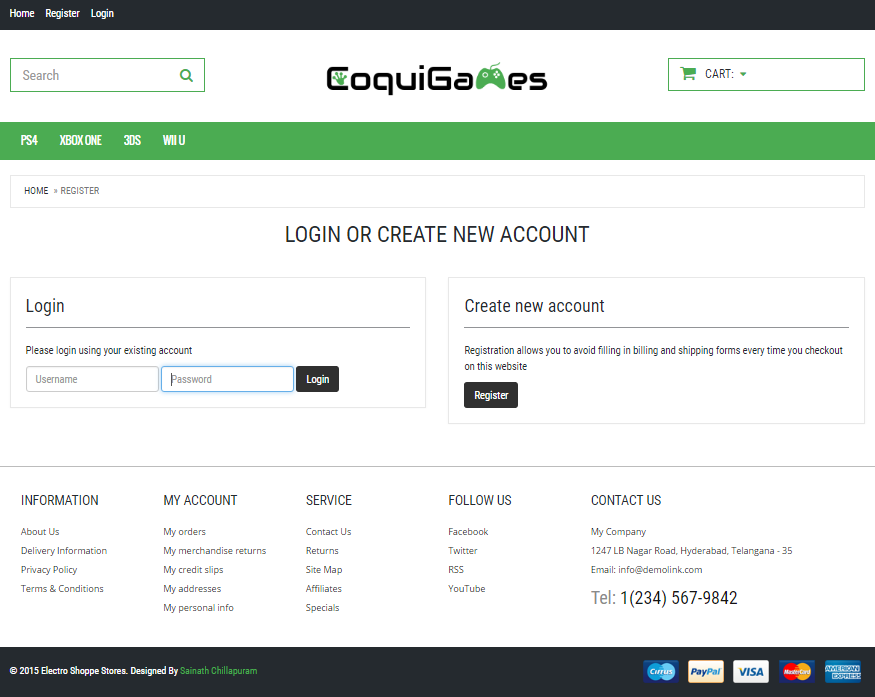
**5. Client – Side Screen Shots**



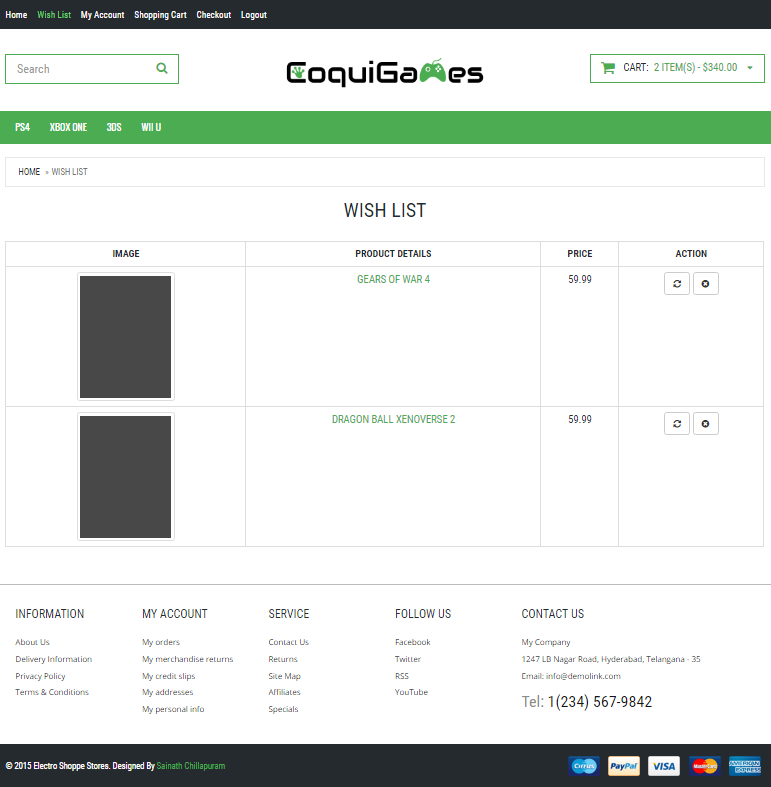
**Fig.1** Homepage of CoquiGames



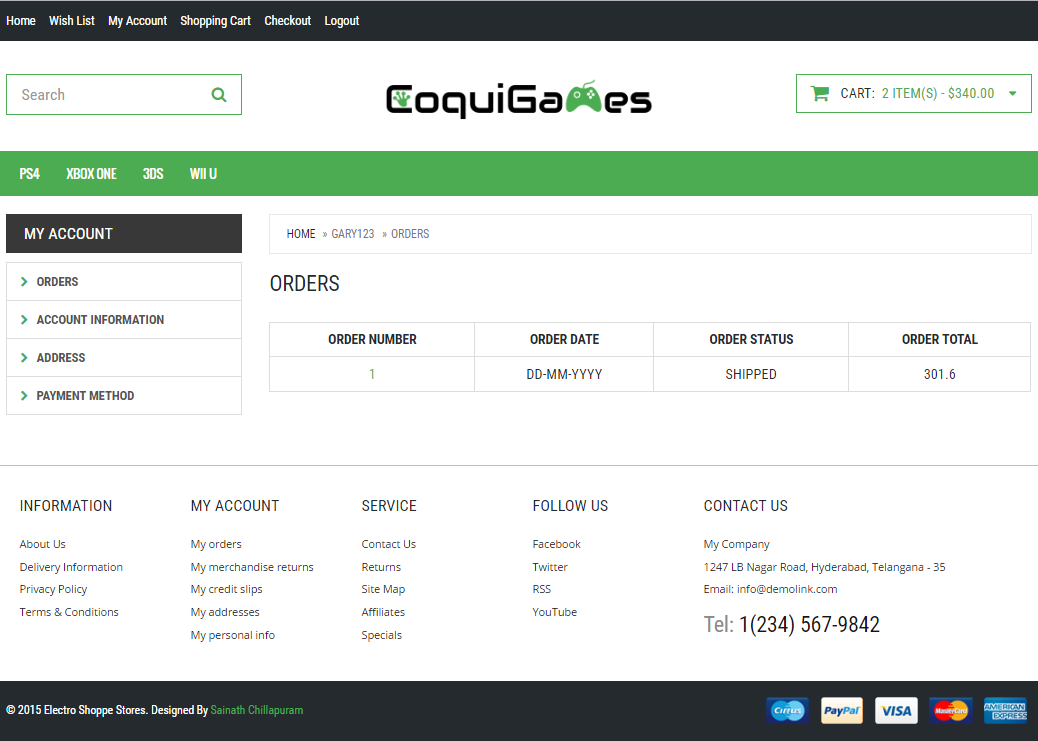
**Fig.2** Register view



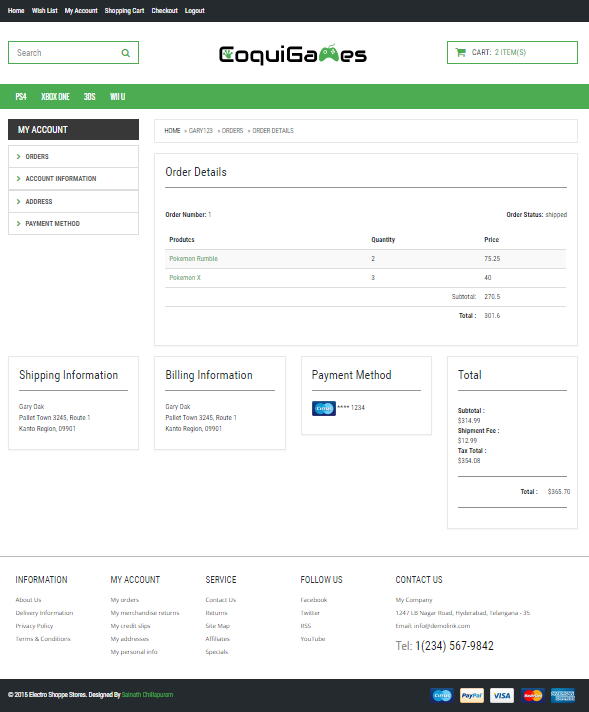
**Fig.3** Login view



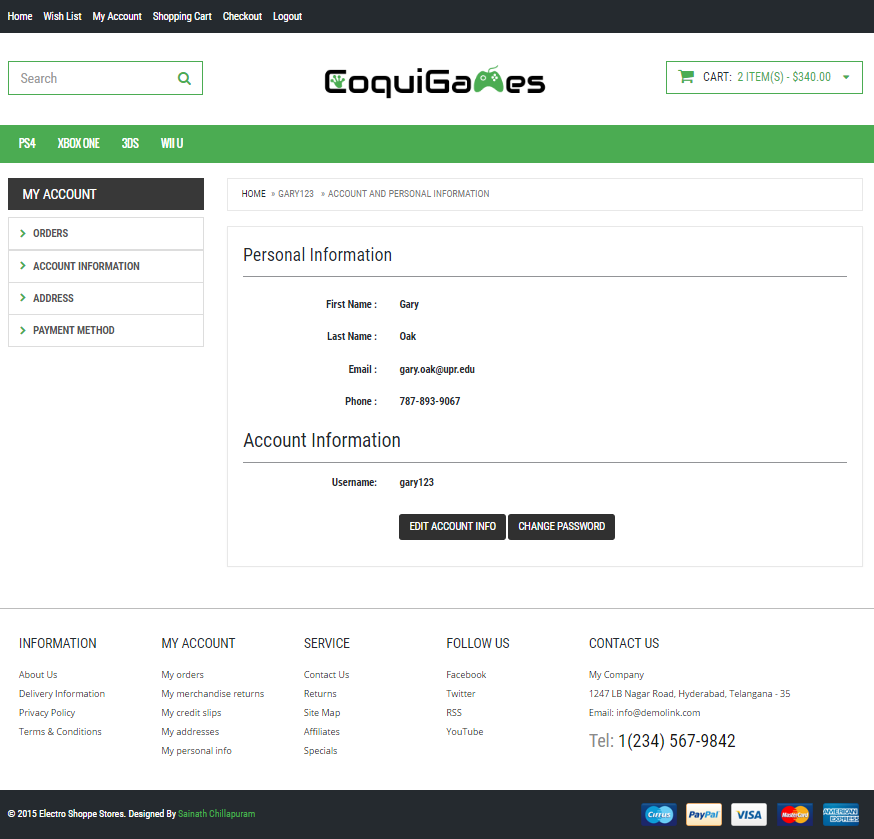
**Fig.4** Wishlist view for an account user



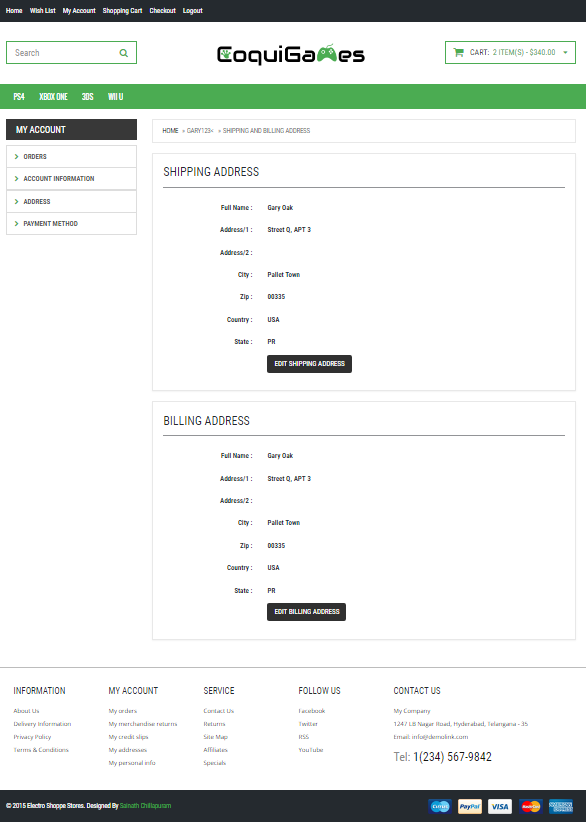
**Fig.5** MyAccount -> Orders view for an account user



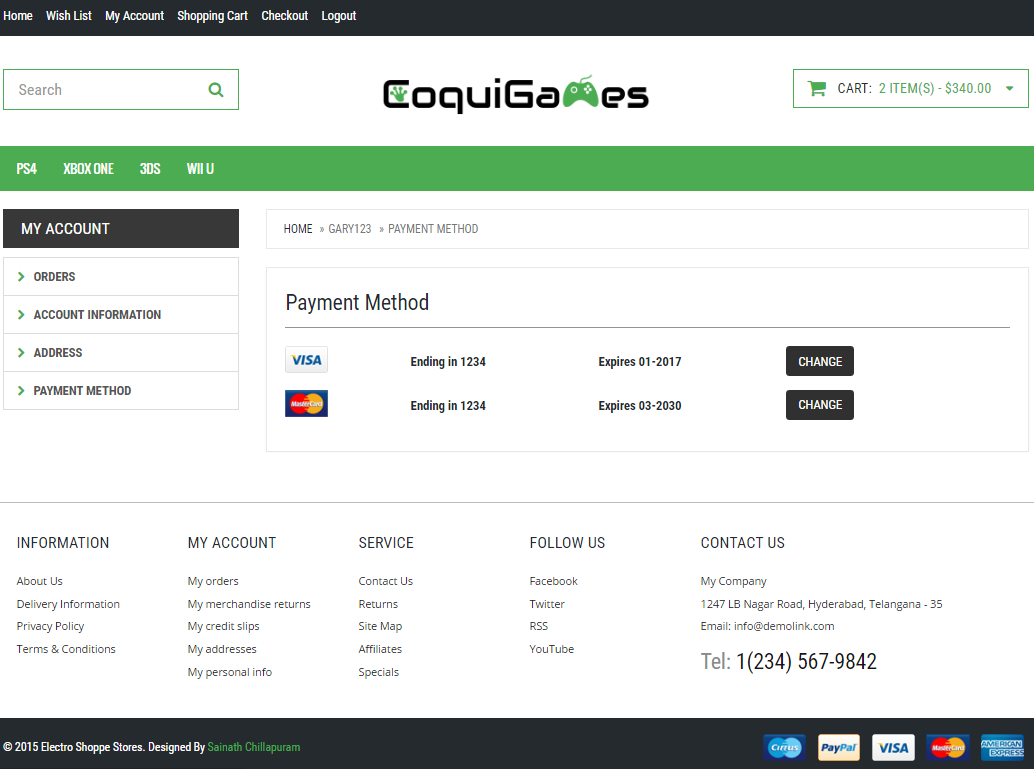
**Fig.6** MyAccount -> Orders -> OrderDetails view for an account user



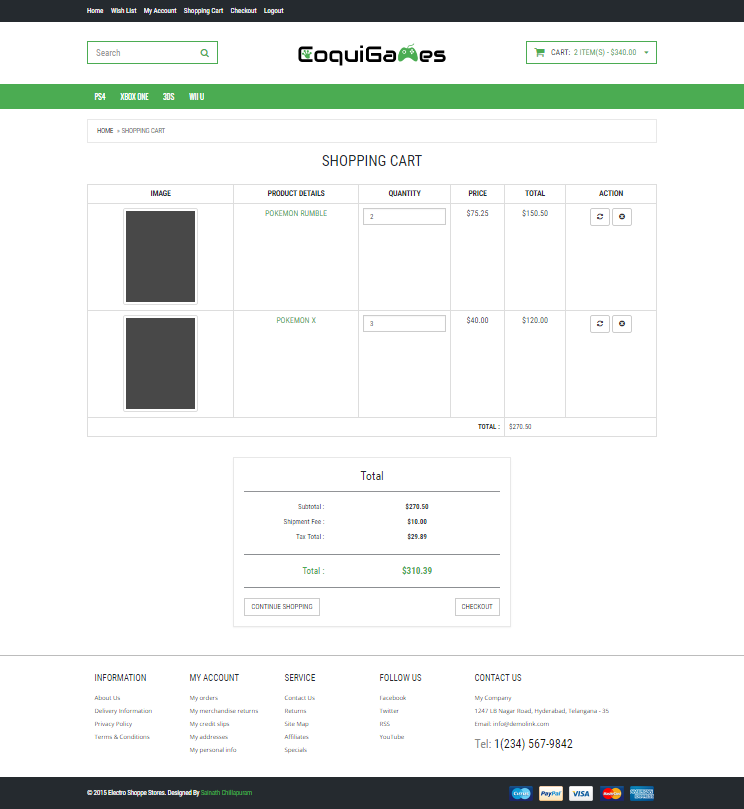
**Fig.7** My Account -> Account Information view for an account user



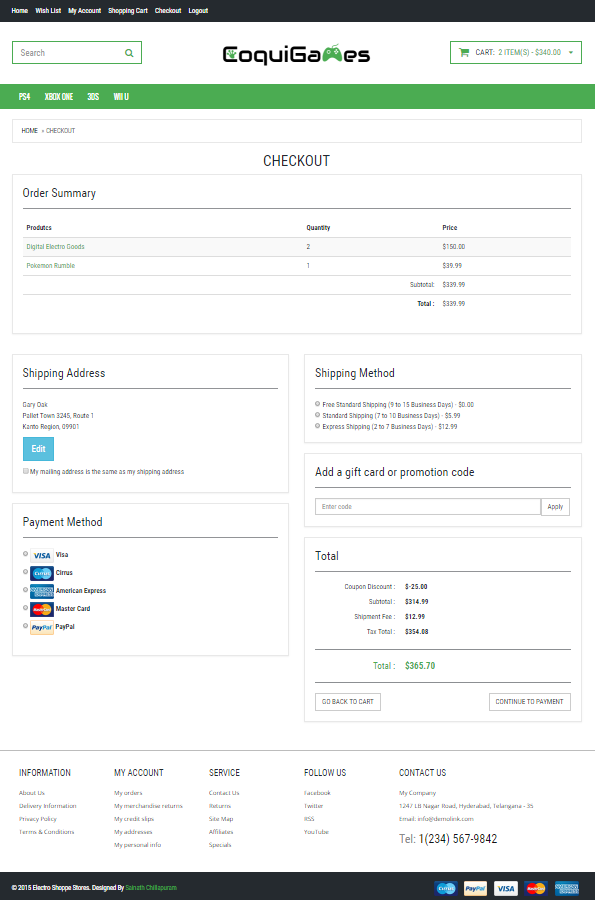
**Fig.8** My Account -> Address view for an account user



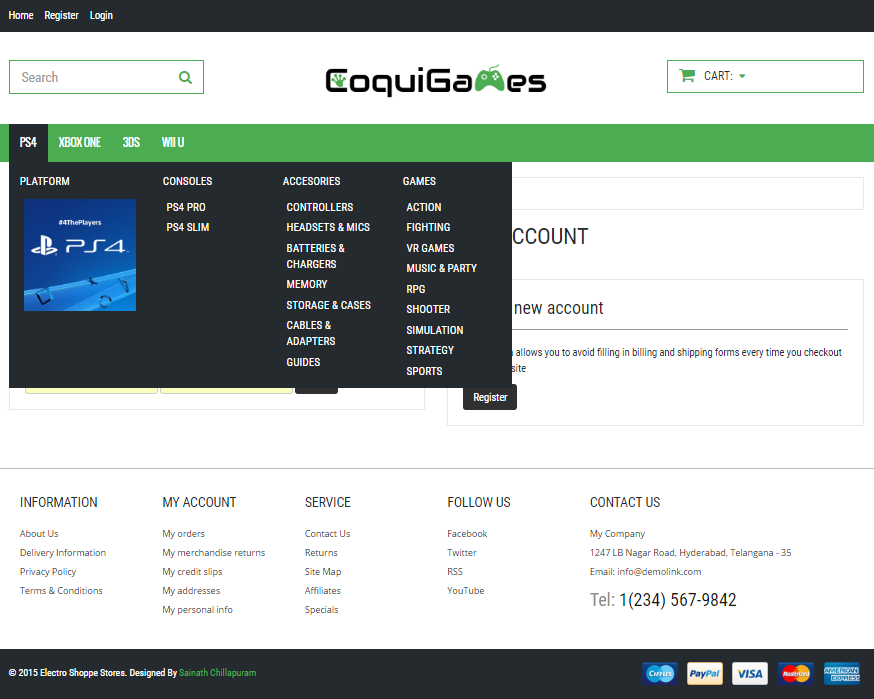
**Fig.9** My Account -> Payment Method view for an account user



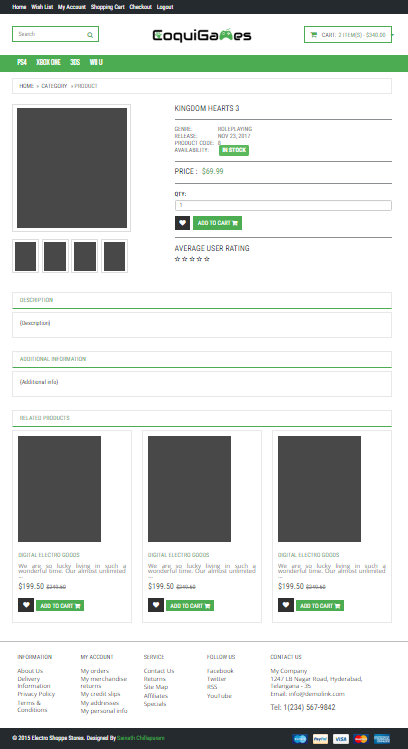
**Fig.10** Shopping Cart view for an account user



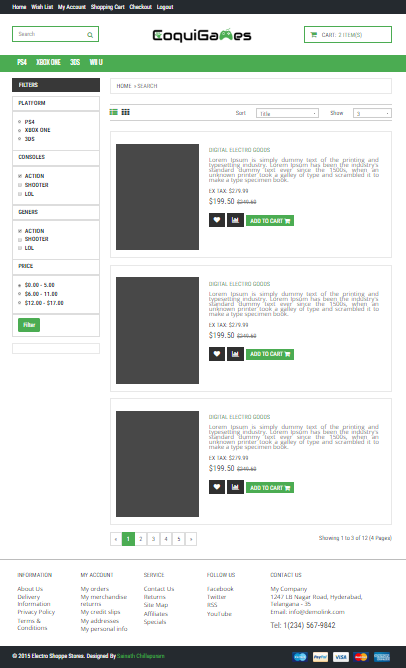
**Fig.11** MyAccount Orders view for an account user



**Fig.12** PS4 platform navigation bar hover view



**Fig.13** Product view for Kingdom Hearts 3



**Fig.14** Product list view