Peggy E. González Ojeda

802-13-3175

Kimaki’s Game Place

**Domain:**

An E-commerce of electronic games and videogame. Customers interested in videogames and other types of games at Kimaki’s Game Place will be found (or at least something). Anyways, Customers will search their preference for the products since we have a variety of games, including consoles, equipment, even the latest technology of virtual reality AND augmented reality. The Customer can have access to this service via Online (Web Application) and Mobile Application. The Customer can register with various methods with email, Facebook Account or Google+ Accounts.

\*\*\*Note: All content of as games, video games, console VR and AR will be called as Product.

**Assumptions:**

* Many Account to many Customers
* Many Customers to Many Products
* Many payment method (paypal or international credit card)
* Many Developers
* Stock service has many suppliers
* Access(Login) WebApp and MobApp many Custumers
* CustomerSupport will be added as an option for Account problems (manage by Account)
* Each Customer will get notifications that can be about products or Cart updates (by Account)
* Each entity does not have any additional (unknown) attributes or relationships.

**Schemas:**

ACCOUNTS(Login, Cart, Setting, Notifications)

DEVELOPERS(AccManage, WebMobManage, StockSupplierPayIssues)

STOCK(Product, StockNotification, Price)

SUPPLIERS(SupplierProd, Price)

WEBAPP(AccountLog, ProductView, NotificationPurchase, ProdStock, ProdShipTime, Price)

MOBAPP(AccountLog, PlataformApp, ProductView, NotificationPurchase, ProdStock, ProdShipTime, Price)

PAYMENT(VerifyPurchase, NotificationPayment, PayMethod)

CUSTOMERS(InterestProd, AccLog)

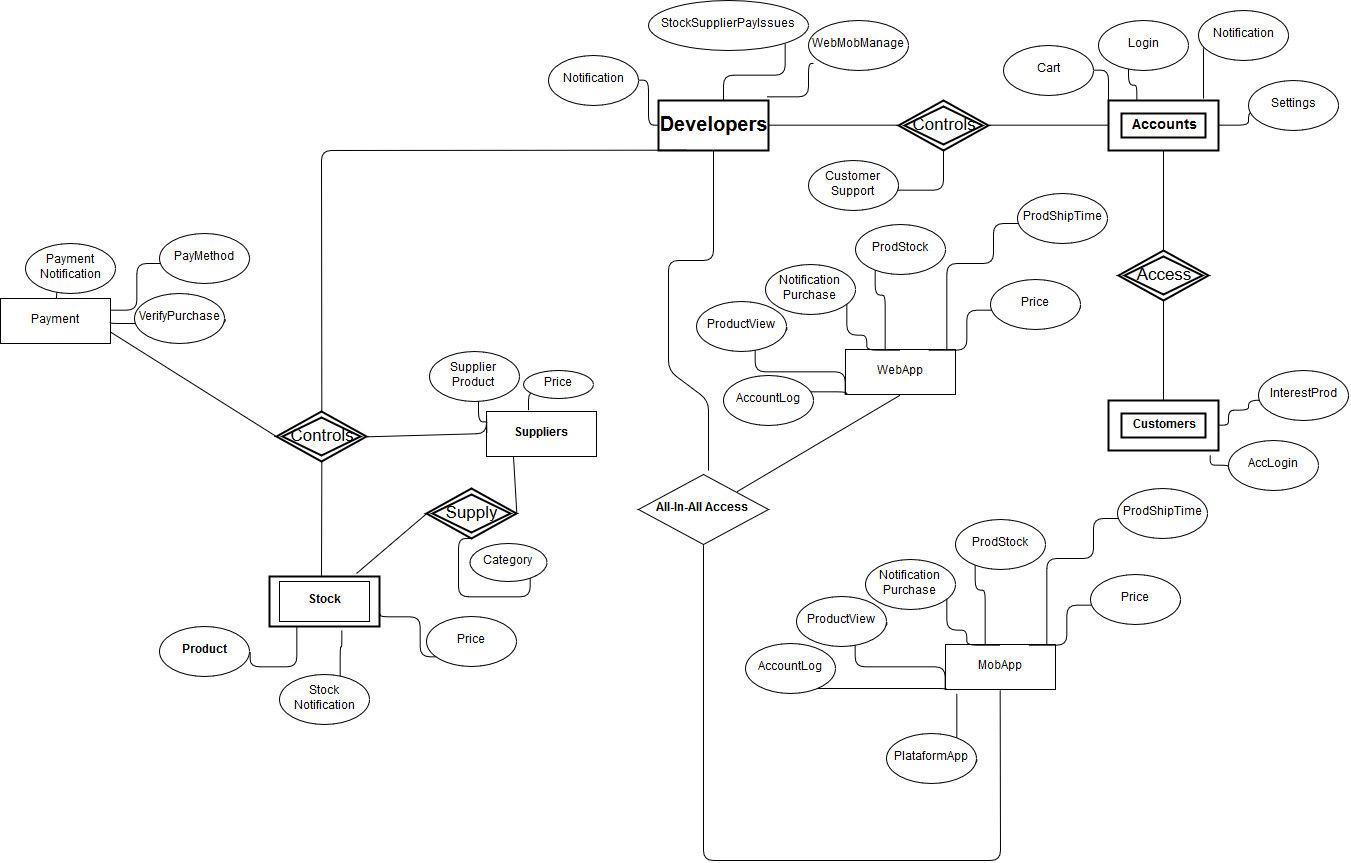
**Updates:**

* A lot, let’s start with “ADMI\_TECH” was replace with DEVELOPERS because the developers will have all the access to the accounts and manage every technical problem including payment, customer support, accounts, suppliers and any other problems.
* Divide WEBAPP with WEBAPP and MOBAPP as 2 different methods of access.
* Added STOCK and ACCOUNTS entities.
* Added in each entities new attributes.

**Explanation:**Accounts is a weak entity which include attributes Login (can be connected with Facebook, G+ or email), Cart (products to buy), Setting (to control the account), Notification (necessary to let the customer about updates, promotions, and any other important information) which will have a weak relationship with a weak entity of Customers with attributes of interested products and Account access (login) because the Accounts depends on Customers and Developers. Now Developers are a strong entity which controls all, having attributes as Account management, Customer Support, Web and Mobile management, also in charge of the Stock, Supplier, and Payment issues. Stock is another weak entity with attributes of products, Category, Stock Notification and Price having a relationship with Supplier as a strong entity with attributes of Product Supplier, Categorize products and Price. WebApp and MobApp are connected with the Accounts, Stock and Developers with an **All-In-All Access.**

**Note: All-In-All Access** relationship is the one that unites all the controls of all of the other entities. I mean, since Mobile and Web application needs to be connected with all other entities the best way is to be connected directly with the Developer so that can have access to all others entities.

**New version:**



**Old Version:**

