E-Commerce
Team Project
Final Report

Team Members:
Kofi Omari Asante
Papa Kojo Oseku-Afful
Kelvin Akakpo
Israel Nwachukwu Orevaoghene

Site: https://e-commerce-lab.000webhostapp.com/

Business Plan
AMENO

Ameno hopes to provide its customers with an easy and convenient way of purchasing sound equipment used during church services. The goal is to allow customers to easily shop for their goods online, pay with any available means such as Mobile Money, visa card etc and also have their goods delivered to them in the comfort of their auditorium.

Equipment such as sound mixers, guitars, speakers, drum sets, consoles, string instruments would be available for purchase on the site. Products listed will hold a valid warranty of a minimum of 2 years. On our website, customers will have the functionality to login, view available items, access to sales and discounts and add items to their cart, access to delivery, as they shop on the site. When a user is about to checkout, the user can make any changes they wish to, and users can fill a data collection form that would allow us to contact them concerning delivery. The main revenue stream of the business will mainly be through buying of equipment on the only site. As the website develops and gains traction, the goal is to add other revenue streams such as advertisements of other services on the website.

As of now we believe our main competitors to be all and any company selling church equipment. However we believe that we have the advantage because not many of these sellers have an online presence. And so this would give us an advantage as we can reach a wider market this in turn means that we can offer much more convenience to our customers. For our marketing strategy we would target 3 consumer groups. These groups are church organizations, musical groups, and individuals. Marketing campaigns will be carried out through social media marketing, online marketing and word of mouth.

For financial planning and projections, we are estimating a Total Available Market (TAM) of 80,000 and a Total Obtainable Market (TOM) 50,000 consumers.

When it comes to the management team and its structure; Kofi Omari Asante is the overall manager and is responsible for frontend design and providing support to the backend developers. Kelvin Akakpo is responsible for the database creation, its management and would be our public relations officer. Papa Kojo and Israel are both mainly responsible for our backend development.

Architecture & Design

Systems Development Lifestyle Cycle

Business objectives, system functionalities, and information requirments			
business objectives	**system functionalities**	**information requirements**	**members**
display goods	digital catalog	dynamic text and graphic catalog	Kelvin, Omari
provide product information	product database	product description, inventory level	Israel
excecute transaction	shopping cart/payment system	secure credit card clearing and payment options	Omari
accumulate customer information	customer database	name, address, phone, email, customer registration	Papa Kojo
provide after sale support	sales database	customer id, product, date, payment, shipment date	Papa Kojo
coordinate marketing and advertising	email, campaign manager	customer linked to email, campaigns	Israel
understand marketting effetiveness	site tracking and reporting system	number of unique visitors, products purchased, pages visited.	Omari
provide production and supplier links	inventory managing system	product and inventory evels, supplier contacts, order quantity data	Kelvin
engage customer in conversation	emails and contact us page	customers linked to email	Kelvin

Framework

Model-View-Controller

MVC is a popular industry-standard web development framework for developing scalable and flexible projects. The model-view-controller (MVC) is an architectural pattern that divides an application into three main logical components: the model, the view, and the controller. Each of these components is designed to handle specific parts of application development.

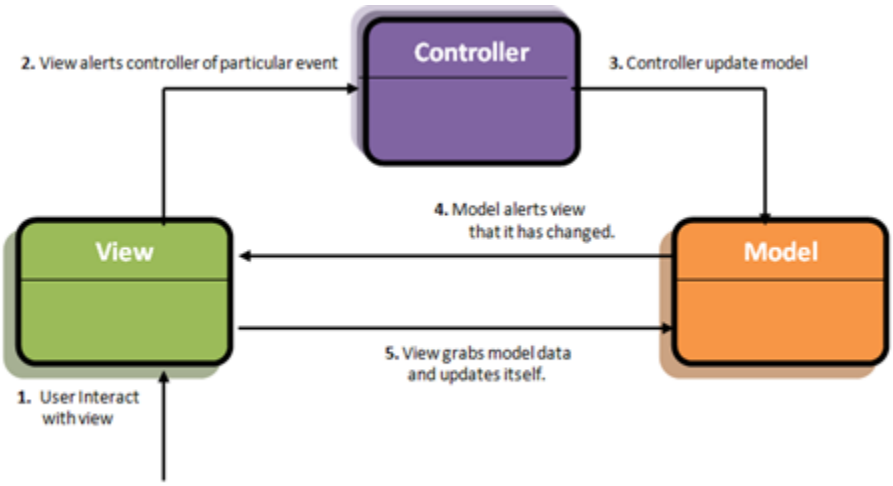
- Model Layer: The model layer encompasses all data-related logic that the user interacts with. Data in the model layer can be data that is being transferred between the view and controller components, or data that is required by other components. Essentially, the Model layer's job is to handle the data that is used throughout the program. The model layer will keep track of our numerous devices' data. It will be implemented with the help of phpMyAdmin and MySQL.

- View Layer: The View component of the MVC architecture refers to the application's user interface, or frontend which are the physical components displayed for users to see. This could include features like buttons, colors, text, and images. The View focuses on the appearance of the application's frontend. The View component of our web application will be built using

HTML5, CSS3, bootstrap and JavaScript. These markup and programming languages allows users to create interactive areas of online applications.

- Controller layer: The controller component serves as a link between the Model and View components, processing all business logic and incoming requests, manipulating data using the Model, and interacting with Views to produce the final output. The Customer controller, for example, will handle all interactions and inputs from the Customer View and use the Customer Model to update the database. The Customer data will be seen using the same controller.

MVC framework architecture



Software Development Life Cycle

Software Development Life Cycle (SDLC) is a set of steps used to create software applications. These steps divide the development process into tasks that can then be assigned, completed, and measured. The following steps of software development include:

- Planning- The planning step also includes determining the project's quality assurance requirements and identifying the project's risks. The technical feasibility study's goal is to identify the many technical techniques that can be used to successfully implement the project with the least amount of risk.

- Requirement analysis - The most important step of the SDLC is requirement analysis. It is carried out by the team's top members, with input from the customer, the sales department,

market surveys, etc. This data is then utilized to establish the main project approach and conduct product feasibility studies in the areas of economics, operations, and technology.

- Design product Architecture - A design approach identifies all of the product's architectural modules, as well as its communication and data flow representation with external and third-party sources.

- Software development - The actual development of the product begins at this stage of the SDLC. During this stage, the programming code is generated according to DDS. Code generation can be done quickly and easily if the design is done in a precise and organized manner. Developers must adhere to their organization's coding rules, and programming tools such as compilers, interpreters, and debuggers are used to develop code. Code is written in a variety of high-level programming languages, including PHP. The programming language is chosen based on the type of software that is being created.

- Testing - As testing operations are mainly included in all phases of SDLC in modern SDLC models, this stage is usually a subset of all stages. However, this stage only refers to the product's testing step, during which flaws are reported, monitored, repaired, and retested until the product meets the SRS's quality criteria.

- Deployment- The product is formally released in the right market once it has been tested and is ready to deploy. Product rollout can also be done in stages, depending on the company's business strategy. The product might be tested in a real-world business environment before being released to a wider audience (UAT- User acceptance testing).

The product may then be released as is or with proposed enhancements in the intended market niche, depending on the feedback received. After a product is launched, it is maintained for existing customers.

Implementation

We used the MVC model in implementation. The model-view-controller (MVC) is an architectural pattern that divides an application into three main logical components: the model, the view, and the controller. We separated the processing files in the action folder, the views in another folder, the controllers and classes in another folder, the settings in another folder which contains the database and connection. We made use of bootstrap, and front-end templates for our designs. A database was set up with nine tables and a connection was created. The website has two roles/types of users this was ensured by checking if a user has logged in and what permission they have, given by a user's role.

The main languages we used were php and javascript for validations, html and css. Php was used for server side scripting. Php functions were used in combination with html to populate tables. The website contains a client side model and an Admin side Model. For the client side when a user enters the address of the site they land on the homepage, here the user can choose to create an account after which they can login. The homepage displays featured products on the site, a user can browse through the products, the shop menu displays all the products available, here a user can view a single product which provides a detailed description on the product. A user can then add a product to cart, edit cart and checkout and can payment, with paystack. A user can also make use of the search bar to find items on the site with a keyword. A user can add items to a favorites list for later. Admin Side Model. On the Admin side model users are defined under two roles: customers, 0 and administrators ,1. An administrator can add and edit brands in the shop, add and edit categories, add and edit products on display. Products in the shop belong to particular brands and categories. An administrator can update products. Product lists show all the products uploaded onto the site and this can be edited, (add, delete,update).