UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

Group Project Development Final Report

Inventory management system



Group 11 -ICT&CS

Đặng Hoàng Phúc BA9-050

Đoàn Đình Nam

Nguyễn Thị Nhàn

Hanoi, February 2022.

# Introduction

## Context and Motivation

Inventories are all resources that are kept in reserve to meet current or future needs. Inventories include not only finished goods inventory, but also work in progress, raw materials/components and tools and tools used in production, etc. Too much or too little inventory causes problems. affect the production and business process, so it is necessary to have appropriate inventory management methods.

### What if the inventory is too little?

Inventory is necessary and with a volume suitable for the operating conditions, production and business of the business, it is not dangerous. By keeping inventory, businesses can reduce some costs, or lower initial outlays such as start-up quality costs.

However, if the inventory is not enough, sales will decrease (for inventory of finished products), in addition, because the business cannot supply products according to demand, customers will definitely switch to purchase from competitors.

2. What about high inventory?

Inventory is too low, causing revenue to be affected, however, if the inventory quantity is too high, in addition to long-stocked goods, they will be damaged and lose quality, making it difficult to compete with the above competitors. market, some of the following costs will be higher, such as:

Storage costs: Costs incurred related to storage such as warehouse rent, warehouse insurance, costs of equipment and facilities, costs for human resources for supervision and management, costs Warehouse management, loss and loss costs...

Cost of satisfying customers: If the inventory of semi-finished products is too large, it will hinder the production system. As the time it takes to produce and deliver customer orders, the ability to respond to changes in customer orders weakens.

Production coordination costs: Due to the large amount of inventory that hinders the production process, more labor is needed to clear bottlenecks, solve bottlenecks related to production and scheduling coordination program.

The cost of the quality of large batches: When producing large batches of goods, there will be large inventories. In some cases, some will be damaged and some parts of the production batch will have defects. Smaller batch sizes can reduce the amount of poor quality.

3. The role of inventory management

The task of inventory management is to answer 2 questions: How much inventory is optimal? When to place an order?

In a business, inventory is always one of the most valuable assets out of the total asset value of that business. Typically, the value of inventory accounts for 40% - 50% of the total asset value of a business.

For that reason, good inventory control is always an essential and essential issue in operational production management.

Inventory is the bridge between production and consumption. Every salesperson wants to increase inventory levels to quickly respond to customer needs; Operations and production staff also prefer to have a large inventory because it makes it easier for them to plan production. However, for the finance department, it is always desirable to keep the inventory to a minimum, because the money in the inventory will not be spent on other items. Therefore, inventory checking is an indispensable job, through which businesses can keep inventory at a "just enough" level. It means not "too much" but also not "too little".

The main goal of this project is to build a system for a company to digitize its warehouse management. First, to access the inventory management system, managers including administrators and users need to have a login account to be able to use the functions of the management system.

After logging into your account, the management page will appear and it is where the manager uses to manage information about products, categories, stores, orders, users. They can also view statistics on cash flow, orders, and products through this page.

## Report Structures

This thesis will contain all the information about the report and the project.

The sections are:

Section 1: Introduction

This section provides overall information about the project.

Section 2: Objectives

This section provides the features and expected outcomes of the system.

Section 3: Requirements Analysis

This section provides information about requirements, use cases, and use cases

descriptions.

Section 4: Methodology

This section provides the tools and techniques that are used in the project.

Section 5: Results

This section provides information about what has been done in this project.

Section 6: Test Cases

This section provides information about checks on the operation of the management system and errors encountered during implementation.

Section 7: Future plans

This section provides information about what can be done in the future to improve

the project.

# Objectives

This section will contain brief information about the expected function of the system:

## Desired Features

The goal of this project is to build basic services that help administrators control all activities taking place in the system and decentralize management for users of different groups.

Feature 1: Authentication: Login with user and password.

Feature 2: view dashboard

Feature 3: user management

- Sub-features3.1: Add user information and select authorized groups.

- Sub-features 3.2: View all users.

- Sub-features 3.3: Edit information or delete a specific user.

Feature 4: Group management

- Sub-features 4.1: Add groups and assign permissions to that group.

- Sub-features 4.2:View all groups.

- Sub-features 4.3: Edit or delete a specific group.

Feature 5: Brand management.

- Sub-features 5.1: View all brands.

- Sub-features 5.2: Add, edit or delete trademarks.

Feature 6: Manage categories.

- Sub-features 6.1: View all categories.

- Sub-features 6.2: Add, edit or delete categories.

Feature 7: Store management.

- Sub-features 7.1: View all stores that have received products.

- Sub-features 7.2: Add, edit or delete stores.

Feature 8: Manage item's attributes.

- Sub-features 8.1: View all attributes added to the product's information.

- Sub-features 8.2: Add, edit or delete key properties.

- Sub-features 8.3: Add, edit or delete selection inside each main property.

Feature 9: Product management.

- Sub-features 9.1: Add products.

- Sub-features 9.2: View all products.

- Sub-features 9.3: Edit or delete a specific product.

Feature 10: Order management.

- Sub-features 10.1: Add order.

- Sub-features 10.2: View all paid or unpaid orders.

- Sub-features 10.3: Edit or delete a specific order.

- Sub-features 10.4: Print out the invoice of the order.

Feature 11: View report; Company and manager information.

- Sub-features 11.1: View a report of total paid orders over time.

- Sub-features 11.2: View and edit company information.

- Sub-features 11.3: View account's profile.

- Sub-features 11.4: Edit account information and change password.

- Sub-features 11.5: Log out of the account.

## Expected Outcomes:

This system aims to create a basic and convenient way for managers to do their job. Specific goals are:

- Build a management system with a user-friendly interface for users to interact with.

- Development of an auxiliary tool for product management.

- The system fulfills all the features mentioned in the desired features.

## Requirements Analysis

This section will provide information about use cases, functions of the systems, and diagrams that are needed for the project.

### Requirements Analysis

This section will provide information about use cases, functions of the systems, and diagrams that are needed for the project.

- A login system for authentication.

- Features:

- View dashboard.

- Create new user.

- View and manage user.

- Create new group.

- View and manage group.

- View and manage brand.

- View and manage category.

- View and manage store.

- View and manage attribute.

- Add value to attribute.

- View and manage product.

- View and manage Order.

- View Report.

- View and edit Company.

- View Profile.

- Edit account and change password in setting.

- Logout.

## Use Cases

### Use cases Diagram

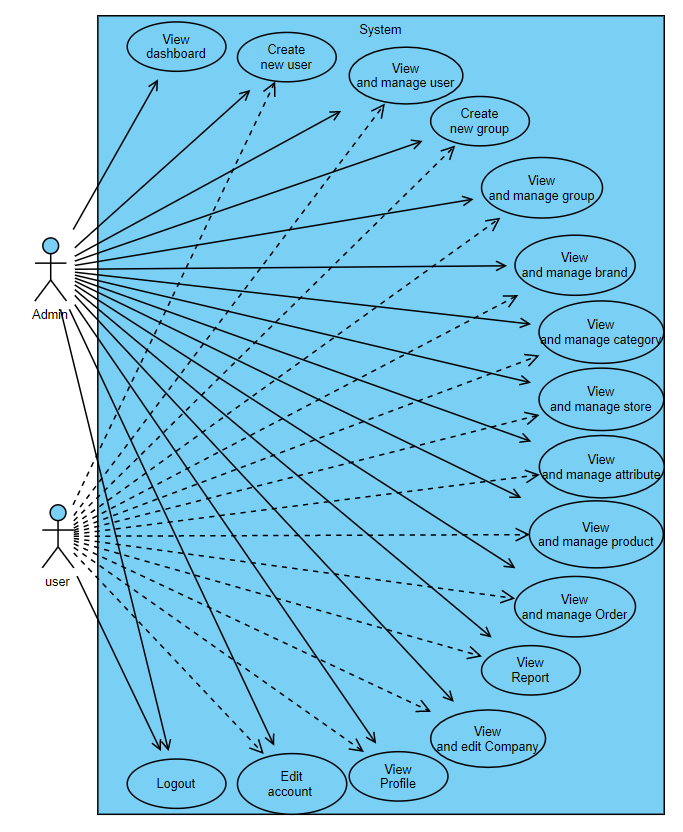


Figure 1: Use Cases

## Use Cases Specifications

### Log in

- Brief description: This use case describes how an actor can log in the pages.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| **Actor Action** | **System Action** | **Data Needed** |
| 1. Actor wishes to login an  account | 2. The system displays a login form | Email  Password |
|  | 3. If the data is valid, the system will confirm that the guest login successfully |  |

Alternative Flow: Step 3

|  |  |  |
| --- | --- | --- |
| **Actor Action** | **System Action** | **Data Needed** |
|  | 3. If the data filled in is invalid, the form will inform and the actor has to fill in the form again. | Email  Password |
| 4. The actor fills in valid data | 5. The system confirms that the guest signed up successfully |  |

- Special Requirements: None.

- Pre-conditions:

The email and password must match the data in the database.

- Post-conditions:

If the use case is successful, the actor logged in the system. Otherwise, the

system state show incorrect username/password combination.

### View dashboard.

- Brief description: This use case describes how an actor can view the information

of the dashboard.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the  dashboard | 2. The system retrieves and  displays the products |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### Create new user

- Brief description: This use case describes how an actor can create the information

of the user.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to create a new user | 2. The system displays a form for new user |  |
| 3. Actor fills in the form with necessary data | 4. If the data is valid, the system will assign the user to the database that the new user create successfully. | Groups  Username  Email  Password  Confirm password  First name  Last name  Phone  gender |

Alternative Flow: Step 4

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
|  | 4. If the data filled in is invalid, the form will inform and the actor has to fill in the form again. |  |
| 5. Actor fills in the form with necessary data | 6. If the data is valid, the system will assign the user to the database that the new user create successfully. | Groups  Username  Email  Password  Confirm password  First name  Last name  Phone  gender |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the new user is added to the

database

### View user.

- Brief description: This use case describes how an actor can view the information

of the user.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the user | 2. The system retrieves and displays the user |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage user

- Brief description: This use case describes how an actor can manage the information of a user.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of user | 2. The system retrieves and  displays a list of user |  |
| 3. Actor chooses a user to manage the information | 4. The system display the chosen user’s information and a form for actor to update information |  |
| 5. Actor fills in the form with  the new data | 6. The system updates the  user. | Groups  Username  Email  Password  Confirm password  First name  Last name  Phone  gender |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the user is updated.

### Create new group

- Brief description: This use case describes how an actor can create the information

of the group.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to create a  new product | 2. The system displays a form for new group |  |
| 3. Actor fills in the form with  necessary data | 4. The system assigns the  group to the database | Group Name  Permission |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the new group is added to the

database

### View group

- Brief description: This use case describes how an actor can view the information

of the group.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the group | 2. The system retrieves and displays the group |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage group

- Brief description: This use case describes how an actor can manage the information of a group.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of group | 2. The system retrieves and  displays a list of group |  |
| 3. Actor chooses a group to manage the information | 4. The system display the chosen group’s information and a form for actor to update information |  |
| 5. Actor fills in the form with  the new data | 6. The system updates the  group. | Group Name  Permission |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the group is updated.

### View brand

- Brief description: This use case describes how an actor can view the information

of the brand.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the brand | 2. The system retrieves and displays the brand |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage brand

- Brief description: This use case describes how an actor can manage the information of a brand.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of brand | 2. The system retrieves and  displays a list of brand |  |
| 3. Actor chooses a brand to manage the information | 4. The system display the chosen brand’s information and a form for actor to update information |  |
| 5. Actor fills in the form with  the new data | 6. The system updates the  brand. | Brand name  Status |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the brand is updated.

### View category

- Brief description: This use case describes how an actor can view the information

of the category.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the category | 2. The system retrieves and displays the category |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage category

- Brief description: This use case describes how an actor can manage the information of a category.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of category | 2. The system retrieves and  displays a list of category |  |
| 3. Actor chooses a category to manage the information | 4. The system display the chosen category’s information and a form for actor to update information |  |
| 5. Actor fills in the form with  the new data | 6. The system updates the  category. | Category name  Status |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the category is updated.

### View store

- Brief description: This use case describes how an actor can view the information

of the store.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the store | 2. The system retrieves and displays the store |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage store

- Brief description: This use case describes how an actor can manage the information of a store.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of store | 2. The system retrieves and  displays a list of store |  |
| 3. Actor chooses a store to manage the information | 4. The system display the chosen store’s information and a form for actor to update information |  |
| 5. Actor fills in the form with  the new data | 6. The system updates the  store. | Store name  Status |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the store is updated.

### View attribute

- Brief description: This use case describes how an actor can view the information

of the attribute.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the attribute | 2. The system retrieves and displays the attribute |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage attribute

- Brief description: This use case describes how an actor can manage the information of a attribute.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of attribute | 2. The system retrieves and  displays a list of attribute |  |
| 3. Actor chooses a attribute to manage the information | 4. The system display the chosen attribute’s information and a form for actor to update information |  |
| 5. Actor fills in the form with  the new data | 6. The system updates the  attribute. | Attribute name  Total value  Status |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the attribute is updated.

### Add value to attribute

- Brief description: This use case describes how an actor can add the information of a value of a attribute.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to create a new value of attribute | 2. The system displays a form for new value |  |
| 3. Actor fills in the form with necessary data | 4. the system will assign the value to the attribute value database that the new user create successfully. | Attribute Value |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the attribute value is updated.

### Create new product

- Brief description: This use case describes how an actor can create the information

of the product.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to create a new product | 2. The system displays a form for new product |  |
| 3. Actor fills in the form with necessary data | 4. If the data is valid, the system will assign the product to the database that the new user create successfully. | Product name  Seri  Price  Quantity  Description  Attribute create  Brands  Category  Store  Availability |

Alternative Flow: Step 4

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
|  | 4. If the data filled in is invalid, the form will inform and the actor has to fill in the form again. |  |
| 5. Actor fills in the form with necessary data | 6. If the data is valid, the system will assign the product to the database that the new user create successfully. | Image  Product name  Seri  Price  Quantity  Description  Attribute create  Brands  Category  Store  Availability |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the new product is added to the

database

### View product.

- Brief description: This use case describes how an actor can view the information

of the product.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the product | 2. The system retrieves and displays the product |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage product

- Brief description: This use case describes how an actor can manage the information of a product.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of product | 2. The system retrieves and  displays a list of product |  |
| 3. Actor chooses a product to manage the information | 4. The system display the chosen product’s information and a form for actor to update information |  |
| 5. Actor fills in the form with  the new data | 6. The system updates the  product. | Image  Product name  Seri  Price  Quantity  Description  Attribute create  Brands  Category  Store  Availability |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the product is updated.

### Create new order

- Brief description: This use case describes how an actor can create the information

of the order.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to create a  new order | 2. The system displays a form for new order |  |
| 3. Actor fills in the form with  necessary data | 4. The system assigns the  order to the database | Customer Name  Customer Address  Customer Phone  Product  Quantity  Rate  Amount  Gross amount  S-charge  Vat  Discount  Net amount  Pain status |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the new order is added to the

database

### View order

- Brief description: This use case describes how an actor can view the information

of the order.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the order | 2. The system retrieves and displays the order |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage order

- Brief description: This use case describes how an actor can manage the information of a order.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of order | 2. The system retrieves and  displays a list of order |  |
| 3. Actor chooses a order to manage the information | 4. The system display the chosen order’s information and a form for actor to update information |  |
| 5. Actor fills in the form with  the new data | 6. The system updates the  order. | Customer Name  Customer Address  Customer Phone  Product  Quantity  Rate  Amount  Gross amount  S-charge  Vat  Discount  Net amount  Pain status |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the order is updated.

### View report

- Brief description: This use case describes how an actor can view the information

of the report.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the report | 2. The system retrieves and displays the report |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### manage company

- Brief description: This use case describes how an actor can manage the information of a company.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes manage the  information of company | 2. The system display company information and a form for actor to update information |  |
| 3. Actor fills in the form with  the new data | 4. The system updates the  company. | Company name  Charge amount  Vat charge  Address  Phone  Country  Message  Currency |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the company is updated.

### View profile

- Brief description: This use case describes how an actor can view the information

of the profile.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the profile | 2. The system retrieves and displays the profile |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

### Setting

- Brief description: This use case describes how an actor can update the information

of the user.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to update user | 2. The system displays a form for user |  |
| 3. Actor fills in the form with necessary data | 4. If the data is valid, the system will assign the user to the database that the user update successfully. | Username  Email  First name  Last name  Phone  Gender  Password  Confirm password |

Alternative Flow: Step 4

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
|  | 4. If the data filled in is invalid, the form will inform and the actor has to fill in the form again. |  |
| 5. Actor fills in the form with necessary data | 6. If the data is valid, the system will assign the user to the database that the user update successfully. | Username  Email  First name  Last name  Phone  Gender  Password  Confirm password |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: If the use case is successful, the user is update to the

database

### Log out

- Brief description: This use case describes how an actor log out the pages.

- Flow of events:

Basic Flow

|  |  |  |
| --- | --- | --- |
| Actor Action | System Action | Data Needed |
| 1. Actor wishes to view the  dashboard | 2. The system retrieves and  confirm that the guest logout successfully |  |

- Special Requirements: None.

- Pre-conditions: None.

- Post-conditions: None.

# Methodology

In this section, the tools and techniques that are used in the project will be listed.

## Tools and Techniques

### PHP

What is PHP?

* PHP is an acronym for "PHP: Hypertext Preprocessor"
* PHP is a widely-used, open source scripting language
* PHP scripts are executed on the server
* PHP is free to download and use
* PHP is an amazing and popular language!
* It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!
* It is deep enough to run large social networks!
* It is also easy enough to be a beginner's first server side language!

What is a PHP File?

* PHP files can contain text, HTML, CSS, JavaScript, and PHP code
* PHP code is executed on the server, and the result is returned to the browser as plain HTML
* PHP files have extension ".php"

What Can PHP Do?

* PHP can generate dynamic page content
* PHP can create, open, read, write, delete, and close files on the server
* PHP can collect form data
* PHP can send and receive cookies
* PHP can add, delete, modify data in your database
* PHP can be used to control user-access
* PHP can encrypt data
* With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.MySQL database.

### Mysql database

MySQL is a very popular open-source relational database management system (RDBMS).

What is MySQL?

* MySQL is a relational database management system
* MySQL is open-source
* MySQL is free
* MySQL is ideal for both small and large applications
* MySQL is very fast, reliable, scalable, and easy to use
* MySQL is cross-platform
* MySQL is compliant with the ANSI SQL standard
* MySQL was first released in 1995
* MySQL is developed, distributed, and supported by Oracle Corporation
* MySQL is named after co-founder Monty Widenius's daughter: My

Who Uses MySQL?

* Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
* Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
* A very large number of web developers around the world

Show Data On Your Web Site

To build a web site that shows data from a database, you will need:

* An RDBMS database program (like MySQL)
* A server-side scripting language, like PHP
* To use SQL to get the data you want
* To use HTML / CSS to style the page

### JavaScript

Javascript is a scripting language, which is one of the core technologies of the World Wide Web. It is a lightweight, interpreted, and just-in-time compiled programming language with first-class functions.

### HTML5

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

### CSS Modal

* CSS is the acronym of “Cascading Style Sheets”.
* CSS is a computer language for laying out and structuring web pages ([HTML](https://atinternet.com/en/glossary/html-2) or XML).
* This language contains coding elements and is composed of these “cascading style sheets” which are equally called CSS files (.css).

### Ajax

Ajax is a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behaviour of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page. In practice, modern implementations commonly utilize JSON instead of XML.

### Bootstrap

* Bootstrap is a free front-end framework for faster and easier web development
* Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
* Bootstrap also gives you the ability to easily create responsive designs

## System Architecture

Basically, both the pages are implemented based on the following architecture diagram.

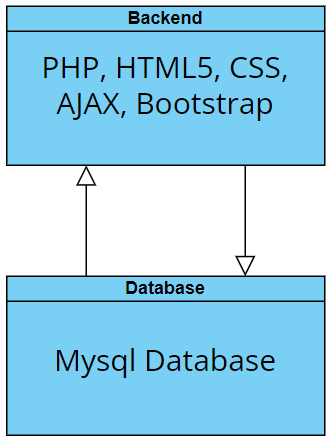


Figure 2: System Architecture Diagram