

ILLINOIS INSTITUTE OF TECHNOLOGY

CS 425

DATABASE ORGANIZATION

CS425 Project: Online Distribution Center

Authors

Hector HERNANDEZ

Ramir AGUILOS

Rafael ZAVALA

Hauyu WANG

Professor

Dr. Omar ALDAWUD

February 28, 2018

Introduction

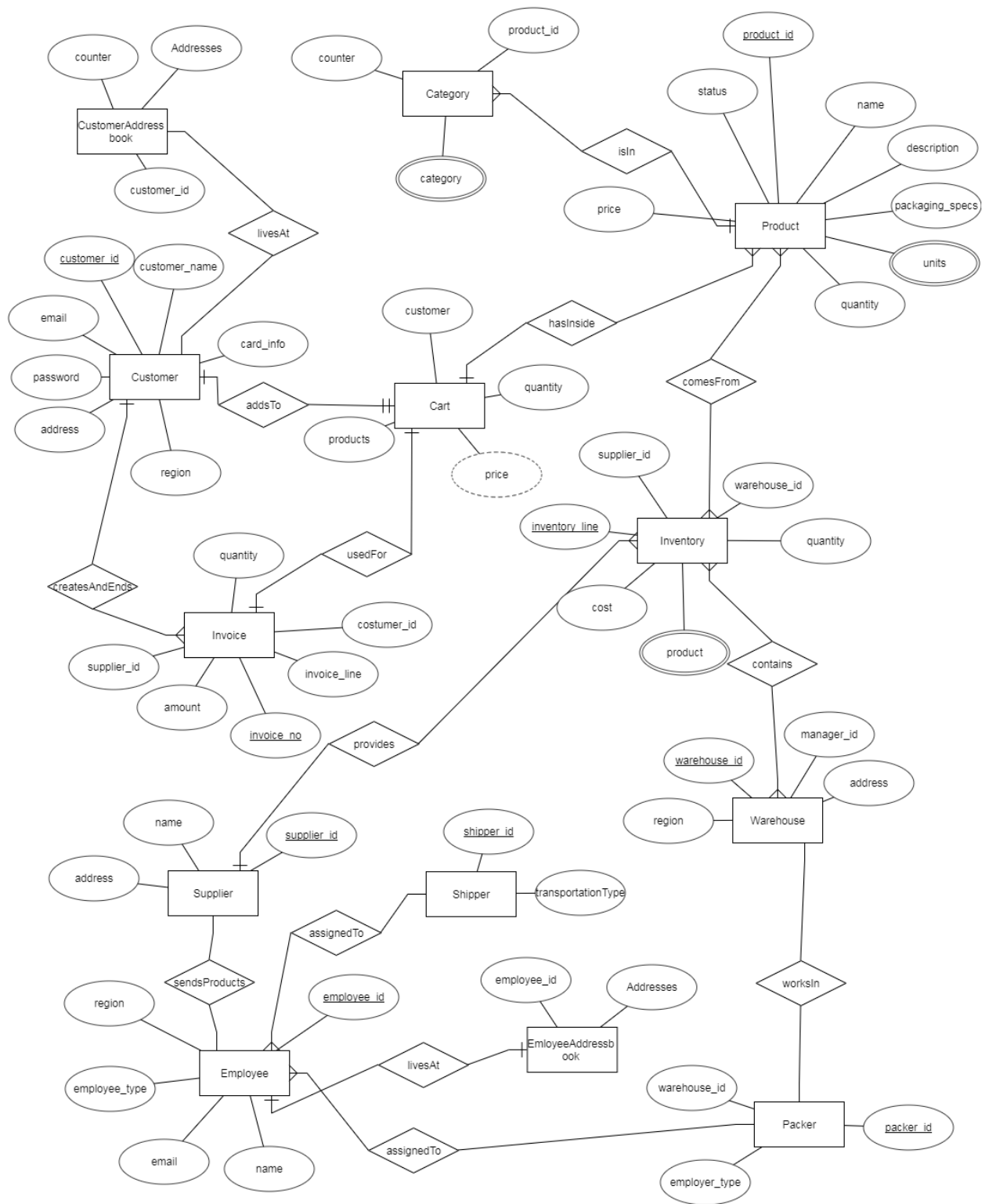
This project consist of a web solution for an e-commerce web application regarding product distribution. From the web application, customers will have access to querying products from different suppliers. Employees of the web application will have specific views tailored to their tasks. The project itself will be conducted using a waterfall model [1] for development. The waterfall model consists of these ordered steps: Requirements, Design, Implementation, Verification, and Maintenance. The requirements that are set out for the development of a software project are constraints, usages, views, functions, and any other interaction between users and the software's interface. For this project, the requirements are listed as a list of functionalities that different users have with the product distribution system. The design of a software project is the development of blueprints or plans for how the development will be implemented. Design also consists of determining what medium and environment the project will be developed in. For the purposes of this project, the design consists of using Github repository as a functioning interaction between the project developers to create files necessary for the database driven software. The implementation of a software project will be the actual creation of the design plans. The implementation of the project will be a web-based application using a DBMS (database management software) as the back-end of the system and HTML, CSS, and PHP, with NodeJS as the front-end of the application. The verification of a software project is the running of tests to determine its readiness to be deployed as a product or service. For this project, testing will be driven by the methods of unit testing each aspect of the web application and having test users use the application. The maintenance of a web application occurs after a software project has been launched or deployed. Ideally, the maintenance of a web application occurs autonomously as the database and server structure will be able to maintain a state of software homeostasis where users can manage their access to the service without losing the integrity of data stored in the database. When there are cases such as data integrity loss or concurrency issues, the physical maintenance and restructuring software are necessary, which would require project developers to address. For this project, maintenance of the web application can be accomplished through many methods such as having more than one physical drive to store user data or creating 'watchdog' programs to monitor the state and integrity of the databases that do not rely on the main server.

Requirements

#	Requirement Description	Testing Criteria
R1	The application shall allow customers to register with the application. The application shall present the user with a registration form with the following fields: Customer Name, Contact Name, Email, phone, region, address, major product line category (research a list that fits most customers).	User enters registration information and the data is stored in the Customer database.
R2	Once a customer registers with the application the application shall generate a unique customer id.	A unique customer is stored in the database.
R3	The system shall allow the customer to enter a password for her account. The password shall be 8 characters long and shall have at least one distinctive character and one uppercase character and a mixed alphanumeric value.	Customers password shall be encrypted and stored in the database.
R4	The system shall allow the customer to enter a password for her account. The password shall be 8 characters long and shall have at least one distinctive character and one uppercase character and a mixed alphanumeric value.	Customers password shall be encrypted and stored in the database.

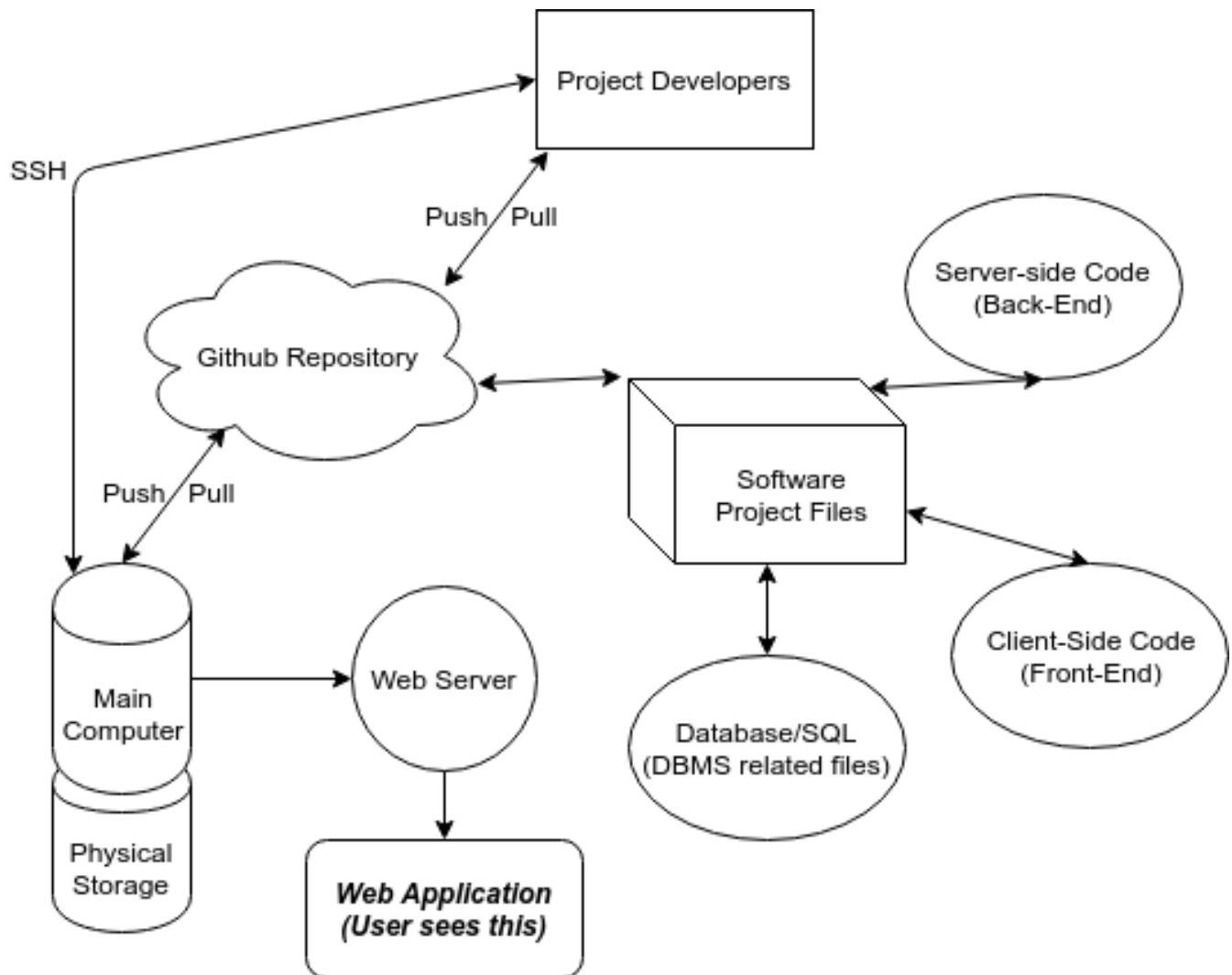
Design

Entity Relationship Diagram



Implementation

The implementation of the project will be a web-based application using JDBC program files as a back-end and Node.js, HTML, CSS, and PHP program files as a front-end. The development environment that the project developers will use is the UNIX operating system environment and tools connected with Github serving as a functioning repository for back-end and front-end code to be updated as the project progresses. The web application will be hosted on an apache web server linked to a main computer with Ubuntu Linux as its operating system. This main computer can also be accessed via SSH for the project developers. Below is a diagram for the method of implementation:



Verification

The verification of the project will be performed by implementing different testing methods [2] with the web application. Firstly, unit testing will be implemented by testing individual aspects of the web application. For example, a customer is able to complete transaction to his or her request. So the implementation of unit testing here would exist as making sure each individual part of the transaction occurs correctly. In accordance to the ERD, the 'Invoice' is an entity with data that is updated atomically; specifically, this is the record of a transaction a customer may request. To unit test the entire transaction, code will be debugged with each iteration of the program until the transaction results in a feasible output (i.e. the invoice). By implementing unit testing during the development of the web application, each individual function of the web application will have higher success of functioning correctly because each function would have been tested to work individually. System testing is also a necessity for the project because system testing captures the users' perspectives and views, which is important in keeping integrative yet private data. System testing is best described as having a box with an input and output where specific users have specific views. In this project for example, if a customer orders a product, a customer would like to know what warehouse it is coming from but may not need to know who the supplier of the product is. Integration testing is the methodology of testing where developers test to see if their software works with different hardwares and interfaces. To actually have users of the application, users must be able to use their own devices and hardwares to access the application, which makes integral testing necessary. Performance testing is a method of testing to see the efficiency of the application. For this project, performance testing will be implemented by having the product distribution service have relevant information readily available for different users.

Maintenance

References

[1] *Waterfall Model*

<http://testingfreak.com/waterfall-model-software-testing-advantages-disadvantages-waterf>

[2] *Testing Methodology*

<https://www.guru99.com/testing-methodology.html>