Miguel Sietereales
Jason Sioco

Threat Model

1. **Security Objectives**
    - To preserve the confidentiality of customer's information.
    - To retain the authenticity of the information regarding all the transactions made with the customer.
    - To ensure the right privileges and roles would be granted to the user.
    - To secure the availability of information for each roles.

2. **Application Overview**
    - **Roles**
        i. Unregistered Customer
            1. Can view products
        ii. Registered Customer
            1. Can buy products that are available in the inventory
            2. Can view personal purchase history
            3. Can leave reviews to the products they purchased
        iii. Administrator
            1. Can create a Product Manager and Accounting Manager account with temporary password that needs to be changed within 24 hours.
        iv. Product Manager
            1. Can edit product information
            2. Can add and edit delete products
        v. Accounting Manager
            1. Can view all the records regarding transactions or purchases of products.
    - **Key Scenarios**
        i. Unregistered user views all the products available.
        ii. Unregistered user uses the filters to view all the products that are under in a specific category.
        iii. Unregistered user searches for a product by using its name.
        iv. Unregistered user looks for the reviews of the selected product.
        v. Unregistered user registers a new account to the system.
        vi. Registered customer logs into a valid account.
        vii. Registered customer selects a product and add it to a cart.
        viii. Registered customer checks-out after adding products to the cart.
        ix. Registered customer completes the transaction.
        x. Registered customer has the choice to add a review to the product bought.
        xi. Administrator logs into a account with Administrator privileges.

xii. Administrator creates a product and a accounting manager account with a temporary password.
xiii. Product manager logs into a account with Product manager privileges.
xiv. Product manager fails to change the temporary password within the allotted time.
xv. Product manager changes temporary password with a new password.
xvi. Product manager add, edit, and delete a product from the inventory.
xvii. Accounting manager logs into a account with Account manager privileges.
xviii. Accounting manager fails to change the temporary password within the allotted time.
xix. Accounting manager changes temporary password with a new password.
xx. Accounting manager views all the transactions made since the beginning of time.

- **Technologies**
  i. Apache Tomcat ver ??
  ii. MySQL 5.7
  iii. PHP or JSP ??
- **Application Security Mechanisms**
  i. Using stored procedures when querying data from the database.
  ii. Performing encryption and decryption to sensitive data of the users.
  iii. Adding account lockouts after a number of incorrect password attempts.
  iv. Applying HTTPS protocol to have a better protection from a potential session hijacking attempt.
  v. Checking inputs for escape characters before inserting data to sensitive parameters.
  vi. Adding a long random number or string as session key.

3. **Application Decomposition**
   - **Trust Boundaries**
     i. Web Server Boundary
        1. Responses
        2. Requests
     ii. Database Boundary
        1. SQL Query
        2. Data
   - **Data Flows**
     i. The unregistered user browses the product catalog page. The first page of products would then be retrieved from the database and the data.
     ii. The unregistered user uses the search field and submits a search string. The string would be validated using regex, escaping characters that may affect the database. A stored procedure created within the database would be used and the string would be passed as its parameter.
     iii. The unregistered user registers a customer account by filling up the registration form. All the data from the form would be validated using a

regex to ensure that there are no characters that may affect the behaviour of the server. Details regarding how to contact the person would be encrypted along with the username and password to be used before storing it to the database.

iv. The registered user needs to login to a valid account using the username and password fields. The username and password would then be encrypted and checked if they exist in the database.

v. The registered user can buy products by adding them to the cart then checking out. During the process of checking out, the user should enter the credit card number to be used. The credit card number used in the transaction would be encrypted before storing it the the database along with the items purchased.

vi. The administrator creates an accounting or product manager account. Username and password given would be encrypted before storing it to the database. The date/time used for creating the temporary password should be based on the time of the server side to avoid possible issues.

- **Entry Points**
  - **i. Login**
    1. Login can be accessed by any users.
  - **ii. Registration**
    1. Registration page can be accessed by any users without logging in to the system.
  - **iii. Checkout Cart / Payment**
    1. Checkout Cart/Payment can be accessed by logged in users with a Registered Customer account.
    2. Checkout Cart/Payment can be opened after adding a product to the card.

- **Exit Points**
  - **i. Product catalog**
  - **ii. Product page**
  - **iii. Purchase history**
    1. The user will be able to see only his/her own purchase history only.
  - **iv. Product reviews**
  - **v. Personal Information**
    1. The user will be able to see only his/her own personal information only.

4. **Threats**
   - **Session Hijacking**
     i. Intercepting a session may be possible
   - **Cross Site Scripting (XSS)**
     i. Scripts can possible be injected on the page

- **SQL Injection**
    i. SQL injection can be used to access administrative accounts
    ii. SQL injection can be used to view data that is not intended to be viewed
    iii. SQL injection can remove the integrity of the information stored in the database
- **Brute force**
    i. Credentials for login may be compromised by brute force.
    ii. Denial of service may happen when too many requests are made.
- **Distributed Denial of Service Attacks (DDoS)**
    i. DDoS attacks forces the website to go offline.
    ii. DDoS attack can bring down the e-Commerce website by sending overwhelming number of data and request to the system.
- **Insecure Direct Object Reference**
    i. Predictable account IDs and object names can possibly be guessed if visible in the URL.
- **Misconfigured security**
    i. Misconfigured firewalls, databases and OS can lead to exploitable security vulnerabilities.

5. **Vulnerabilities**
    - Lack of distinction between user types.
    - Failure to sanitize data read from the database.
    - Failure to use either stored procedures or prepared statements when querying.
    - No restriction or any rule in to force the user in creating a complex password.
    - No encryption and decryption of credentials.
6. **Class Diagram**

7. **ERD**

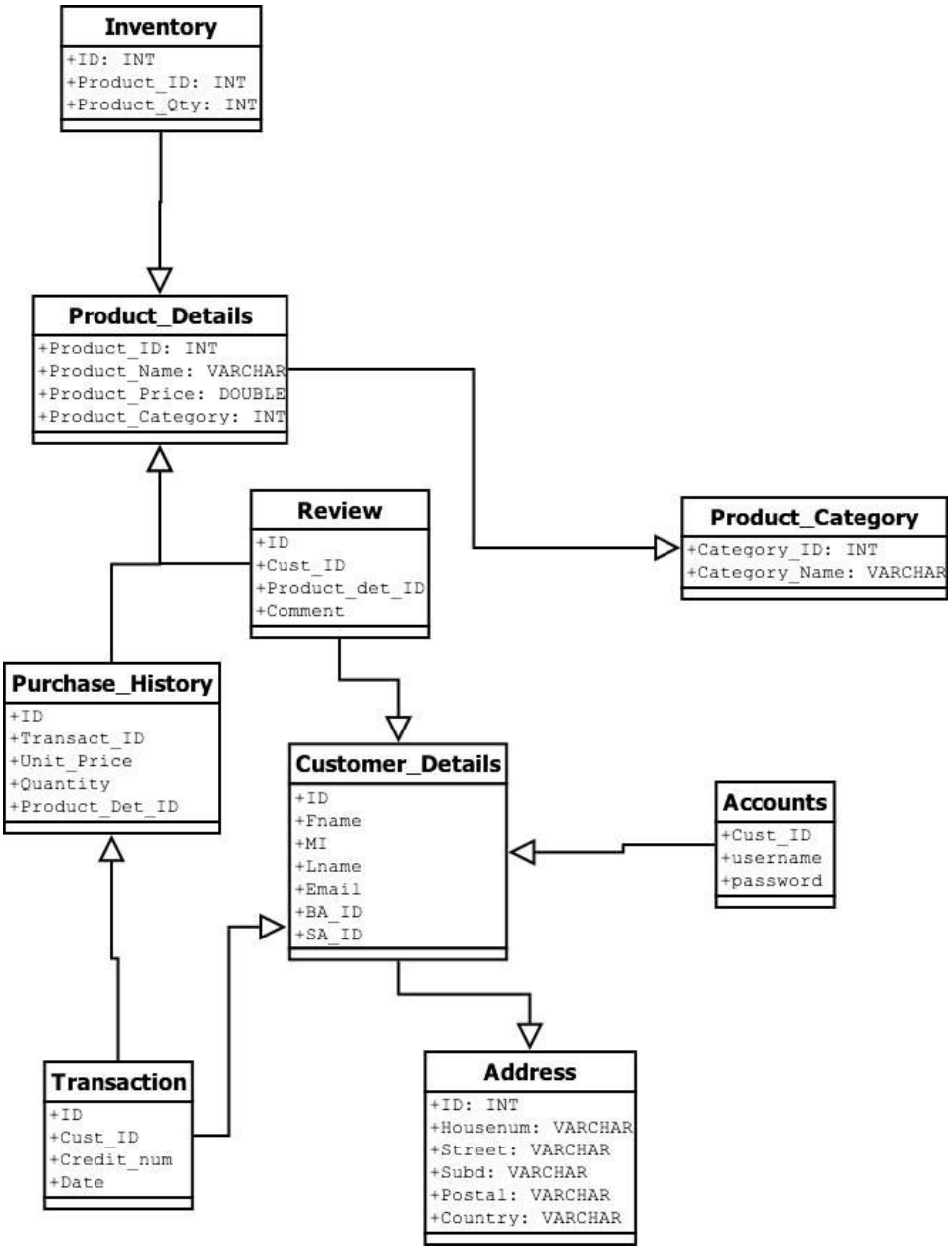

Figure 1. e-Commerce Website ERD