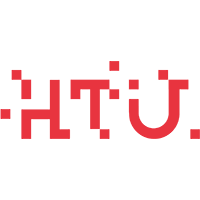
AL HUSSEIN TECHNICAL UNIVERSITY



Faculty of Information Technology

**HTU point of sale**

A graduation project submitted to college of information technology in partial fulfillment of the requirement for the PHP full stack course.

Project Programmer: Mr. Mohammad Bader

Project Supervisor: Mr. Bisharah Estephan

# **Introduction**

HTU point of sale is a web application for HTU university,

it manages the sales in the store, the products, the users and transactions,

this project is basically updating the traditional way of managing the store into web-based application so that the employees can do their job more efficiently and more productivity.

# **Motivation and Problem**

In this part we are going to show the motivation and problems for this project:

* Updating the traditional way of store management
* Reduce the time taken for the accountant to do his/her job
* The accountant doesn’t have to go to the store to do his/her job
* The admin can know the store status from anywhere by anytime without going physically to the store

• Many operations could be done without accessing the store computer

• In case there is an error in the store computer any device could be used to keep the store open

• Reduce the cost because no longer need to use papers

# **Contributions**

We propose in this project a solution for existing problem by updating the system to reduce the chance of errors happened, limiting the usage of paper to save some money, reduce the time and effort from the employees.

# **Project Schedule**

Table 1 shows the project schedule via 1 month.

**Table 1** Gantt chart for the proposed software project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task** | **Duration** | | | | |
| **December** | | | **January** | |
| **W3** | | **W4** | **W1** | **W2** |
| **Planning** |  |  |  |  |  |
| **Analysis** |  | |  |  |  |
| **Design** |  | |  |  |  |
| **Implementation** |  | |  |  |  |
| **Maintenance** |  | |  |  |  |

### **Functional Requirements and Constraints**

This section presents the functional requirements of the proposed software system as data were collected via meetings with the managers and employees of the store.

1. **Login**

* **Functionality name:** login
* **Description:** this page allows the users to access their main pages
* **User:** employees
* **Pre-condition:** the user must login to his/her account by entering username and password
* **exceptional flow:** wrong username or password
* **post-condition:** the user can access his/her main page

1. **Edit profile info**

* **Functionality name:** edit profile info
* **Description:** this page allows the users to edit his/her info
* **User:** employees
* **Pre-condition:** enter new information
* **exceptional flow:** wrong datatype of data
* **post-condition:** the new information has been updated

1. **Add new user**

* **Functionality name:** add new user
* **Description:** this page allows the admin to add new user
* **User:** admin
* **Pre-condition:** enter the new user information
* **exceptional flow:** wrong datatype of data
* **post-condition:** the new user has been added

1. **Add new item to the store**

* **Functionality name:** add new item
* **Description:** this page allows the users to add new item to the storage
* **User:** admin, procurement
* **Pre-condition:** enter new item information
* **exceptional flow:** wrong datatype of data
* **post-condition:** the new item has been added

1. **Sell items from the store**

* **Functionality name:** create new transaction
* **Description:** this page allows the seller to sell items from the store
* **User:** admin, seller
* **Pre-condition:** choose the item to sell
* **exceptional flow:** more items to sell than the store has
* **post-condition:** the new transaction has been added

1. **Non-Functional Requirements**

In this section we present the non-functional requirements of the proposed system.

**A. Performance**

Response time: system should be able to retrieve user requests within 5s. Scalability: system should be able of supporting no less than 100 users at a time when applied.

**B. Security**

System must guarantee that data about diverse types of user’s transactions must be treated in protected channel.

**C. Usability**

Users with diverse background knowledge can simply deal with the system.

**D. Support**

Application, database, and administrative support should be provided 24/7.

**E. Availability**

System should be capable to deliver services when requested 24/7.

**F. Safety**

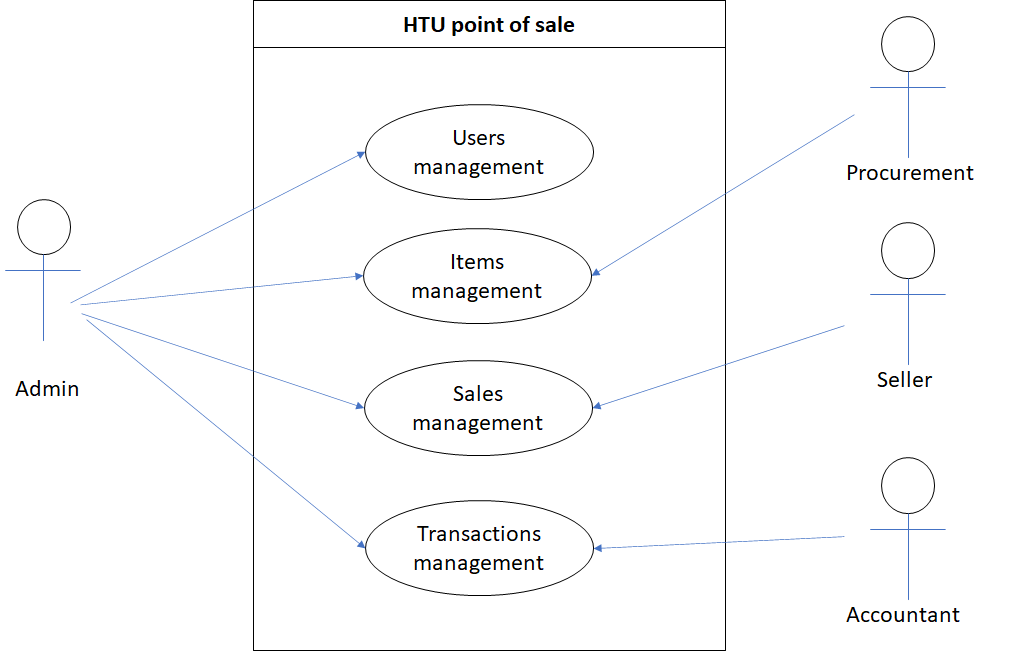
To avoid possible data damages or losses, the system must have a data recovery method.

**G. Reliability**

System should be capable to deliver services as specified

### **Use-case Diagram**

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well



**Figure 1.** Use-case diagram of the suggested system.

1. **Software Architectural Model**

An architectural model is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system or ecosystem.

The software architectural model shows the main software components and the relationships between those components.

Model-View-Controller (MVC) separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., key presses, mouse clicks, etc.) and passes these interactions to the View and the Model.

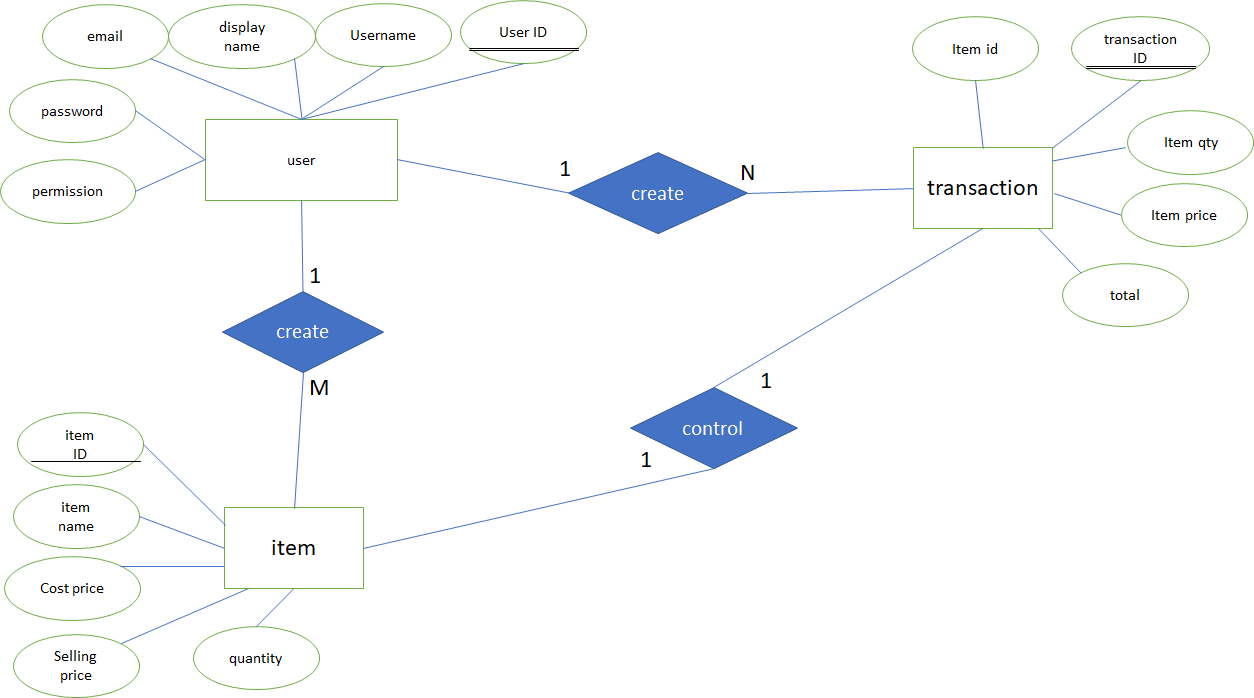
Diagram

Description automatically generated

**Figure 2.** the organization of the suggested software system as Model-View-Controller

1. **Entity Relationship Diagram (ERD)**

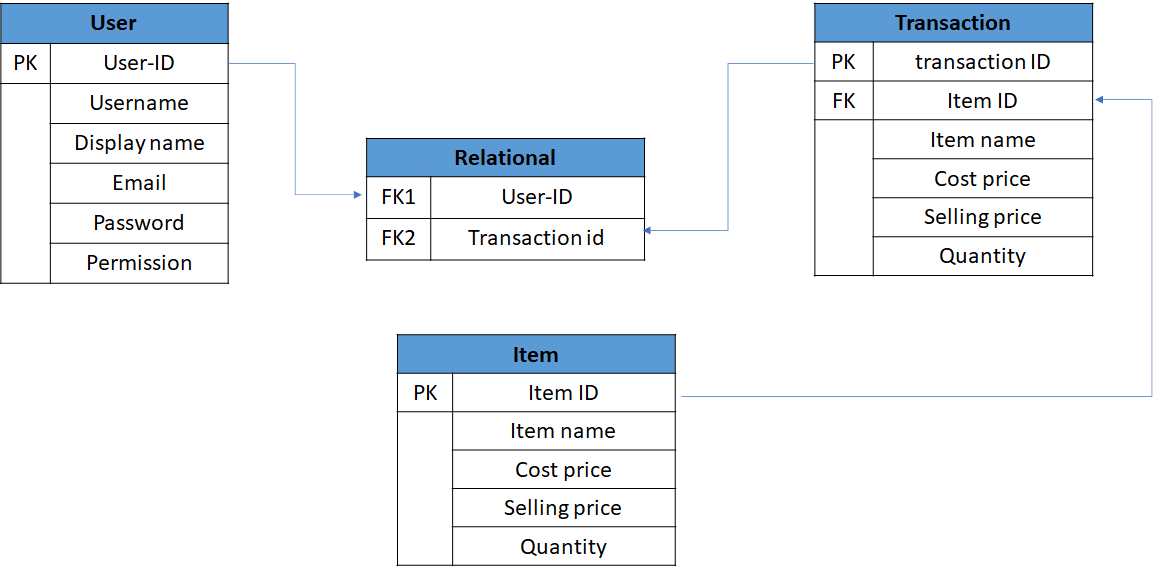
entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts, or events within an information technology (IT) system.

****

**Figure 7. ERD for the proposed project**

1. **Relational Database Model**

The purpose of the relational model is to provide a declarative method for specifying data and queries



**Figure 8.** relational database model for the proposed project