**Original proposal**

**Capstone Proposa**l: RFID Shopping

**Concept:**

The RFID Shopping was an application which uses RFID to verify IDs and purchase groceries. The application shall provide the functions, including what a list of products, prices, and total price which are token by a customer. What are ingredients of each product? What receipt history look like.

**The system shall have these screens:**

* A portal for login customer account.
* A dashboard for users to decide the functions, look up history, purchase groceries, and sign out.
* A screen to show a list of products, quantities, prices and total cost which customers want to purchase.
* A page for showing a product information including name, price, category. If it’s a food, then shows nutrition facts, and ingredients.

**Database:**

Primary tables:

* Customer : Primary key: Customer\_Id
* Receipts : Primary key: Receipt\_Id ; Foreign key: Customer\_Id, payment\_way
* Products : Primary key: Product\_Id ; Foreign key: category\_name
* NutritionFacts : Primary key: NutritionFacts\_Id

The Customers table will contain name, customer id, RFID ID, password, e-mail, and phone number, receipt(s). The Products tables will have, Product\_Id, product name,category, price, and ingredients. The Receipts table would have the Receipt\_Id , Customer\_Id , sum of prices, payment\_way, date, and product(s). NutritionFacts will include serving size,serving per container, calories, saturated fat, trans fat, sodium, potassium, total carbohydrate, dietary fiber, sugars, and protein.

The following support/lookup tables:

* Categories : Primary key: category\_name
* Ways\_to\_pay : Primary key: payment\_way

Category identify what kind of categories (i.e. Grocery, Household Essential, School & Office Supplies, etc.) is it. Ways to pay will identify the payment method types, such as credit/debit cards or cash.

Linking tables:

* Customers\_to\_Receipts : Primary key: CToR\_Id ; Foreign key: Customer\_Id , Receipt\_Id
* Receipts\_to\_Products : Primary key: RToP\_Id ; Foreign key: Receipt\_Id, Product\_Id

Since a customer can have multiple receipts, and a receipt can have multiple products, intermediary tables are needed.

**Ideas/concerns:**

A vendor will categorize all the products into specific groups, but some products belong to multiple categories. For example, adhesive tapes are not only office supplies but also paint supplies. If a vendor wants to give a discount on tapes via using “office supplies” to query products, the vendor would miss altering the tapes whose categories are “Paint supplies”. To simplify first phase, I will postpone updating nutrition facts for each product. It will only include default values. In the future, I propose to memory the lowest prices of each product.

**Capstone Proposa**l: RFID Shopping

**Concept:**

The RFID Shopping was an application which uses RFID to verify IDs and purchase groceries. The application shall provide the functions, including what a list of products, prices, and total price which are token by a customer. What are ingredients of each product? What receipt history look like.

**The system shall have these screens:**

* A portal for login customer account.
* A dashboard for users to decide the functions, look up history, purchase groceries, and sign out.
* A screen to show a list of products, quantities, prices and total cost.
* A page for showing a product information including name, price, category. If it’s a food, then shows nutrition facts, and ingredients.
* The page for user to modify and view their privacy information.

**Database:**

Primary tables:

* Customer : Primary key: Customer\_Id
* Receipts : Primary key: Receipt\_Id ; Foreign key: Customer\_Id, payment\_way
* Products : Primary key: Product\_Id ; Foreign key: category\_name
* NutritionFacts : Primary key: NutritionFacts\_Id

The Customers table will contain name, customer id, rfid id, password, e-mail, and phone number, receipt(s). The password will be stored after encrypted. The Products tables will have, Product Id, product name, category, price, and ingredients. The Receipts table would have the Receipt Id, Customer Id, sum of prices, payment way, date, and product(s). NutritionFacts will include serving size, serving per container, calories, saturated fat, trans fat, sodium, potassium, total carbohydrate, dietary fiber, sugars, and protein.

The following support/lookup tables:

* Categories : Primary key: category\_name
* Ways\_to\_pay : Primary key: payment\_way

Category identify what kind of categories (i.e. Grocery, Household Essential, School & Office Supplies, etc.) is it. Ways to pay will identify the payment method types, such as credit/debit cards or cash.

Linking tables:

* Customers\_to\_Receipts : Primary key: CToR\_Id ; Foreign key: Customer\_Id , Receipt\_Id
* Receipts\_to\_Products : Primary key: RToP\_Id ; Foreign key: Receipt\_Id, Product\_Id

Since a customer can have multiple receipts and a receipt can have multiple products, intermediary tables are needed.

**Ideas/concerns:**

A vendor will categorize all the products into specific groups, but some products belong to multiple categories. For example, adhesive tapes are not only office supplies but also paint supplies. If a vendor wants to give a discount on tapes via using “office supplies” to query products, the vendor would miss altering the tapes whose categories are “Paint supplies”.

The privacy information, such as e-mail, only allow customers themselves or a high-level administrator to view or modify. The password will need more step to modify and cannot see it directly. Passwords will be hidden when entering. E-mail used to reset a password when a user forgot the password. Phone number is used to send a text to users after they paid. It can prevent frauds.

To simplify first phase, I will postpone updating nutrition facts for each product. It will only include default values. In the future, I propose to memory the lowest prices of each product.

Revision notes:

1. Add a new page in screen of system.
2. Customer table, password will be encrypted.
3. Ideas/concerns adds more concern about privacy information.
4. Ideas/concerns adds reasons to explain why the system requires some privacy information.