SMART MARKET MONITORING SYSTEM

SAPM Project – MIS 6308

Group 6 – Team members

-Han Wang

-V B Praneeth Reddy Marreddy

-Roshan Tiwari

Executive Summary

In University of Texas at Dallas, the Compass Group operates a small market in every academic building. Cashiers working in these markets are UT Dallas students looking for part time income. These students report to a manager appointed by Compass Group. The checkout system designed in these markets is connected (through IOT) with a barcode reader to scan product codes, a printer to print payment receipts and a cash drawer that opens up when Cash payment is prompted. The UI for the system is designed on Oracle Micro OS and it is connected to Oracle SQL server. Cashiers sign in with their respective logins and do transactions in their shift and manager can look up the number of sales made by cashier as well as the respective time spent by them on the system.

There are many drawbacks in this system. Firstly, cashier has to do a manual inventory once a week which takes up lot of time and leads to unavailability of certain products during certain times of the week. This problem can easily be solved by feeding the information about every procurement in the system and then system can keep a track record of inventory. The system can check daily for items low in inventory and inform the manager to restock, this will save manual labour and also maintain a regular stock of all items. Secondly, cashiers have to manually look up expiry dates of items and then discuss discounts on items expiring soon as well as remove the items that have expired. This leads to wastage of food products and often times customers are at the risk of consuming expired items. The expiry dates of items can also be fed into the database at the time of procurement and the system can keep a track record of items expiring soon to inform cashiers on time to offer pre-approved discounts and remove expired items from the shelves. Finally, the transactional sales data that is stored in SQL servers is not analysed to provide any business insights. The seasonal sales pattern can provide insight into customer consumption during different times of a year. This can be utilized to procure different mix of products in different months and maximize profits. A basket analysis can be performed on customer transactions to provide insights on optimum product placements and designing combo offers.

PROBLEM STATEMENT

Project Name: UT Dallas Market Cashier Interface.

Project Sponsor: Compass Group (Company that manages markets in various buildings of the campus).

Business Need: The current cashier interface designed on the Oracle Micro OS and uses Oracle SQL database is outdated. New functionalities can be added to reduce manual labour by utilizing IOT concepts. Also, the data stored can be analysed using association rules and regression to provide better forecasting of sales and other for boosting sales

Current Functionalities:

- Current system relates to a barcode scanner to scan the product and access its pricing.
- It adds the pricing to total and calculates tax.
- Current system calculates tax and allows the cashier to choose mode of payments.
- If prompted with Card Payment, it activates card reader and waits for the response
- In Cash payment, it asks for Amount to calculates change amount and open cash drawer.
- Managers have the access to add, modify, and delete products from the database.
- Managers can check cashier summaries to find out the number of sales made by each cashier and monitor their login timings.

Problems:

- The system provides an option to manually enter discounts and does not store that info.
- Cashiers must manually count inventory despite the system having all the sales info.
- Cashiers must manually detect what products are about to be expired.
- Current system does not provide any forecasting or business insight using sales data.

SCOPE: 1. Total cost of setting up the system should not be more than 20,000 dollars. This cost will be used as two-month salary for a backend and a front-end developer.

2. It will take a week's time to train cashiers and managers according to the new system.

FUNCTIONAL SPECIFICATIONS

Proposed Functionalities (Functional):

1. Store discounts information in database and add them automatically when product is scanned.
2. Whenever new product is procured, its quantity will be stored in the database and every time a product is sold the system will deduct it from database. The system will then inform the manager if inventory falls below a threshold.
3. Once a product is procured the system will also take its expiration date as input. It will then inform the manager once that date approaches. Manager can also specify a discount that will be applied to products that are going to be expired soon.
4. Cashier system will do association rules analysis on transactions to decide optimum product placement. It will also create a sales graph to identify seasonal patterns in sales of different products.

NON FUNCTIONAL SPECIFICATIONS

Proposed Functionalities (Non-Functional):

Reliability: 1. System will check inventory two times in a day and expiration dates daily.

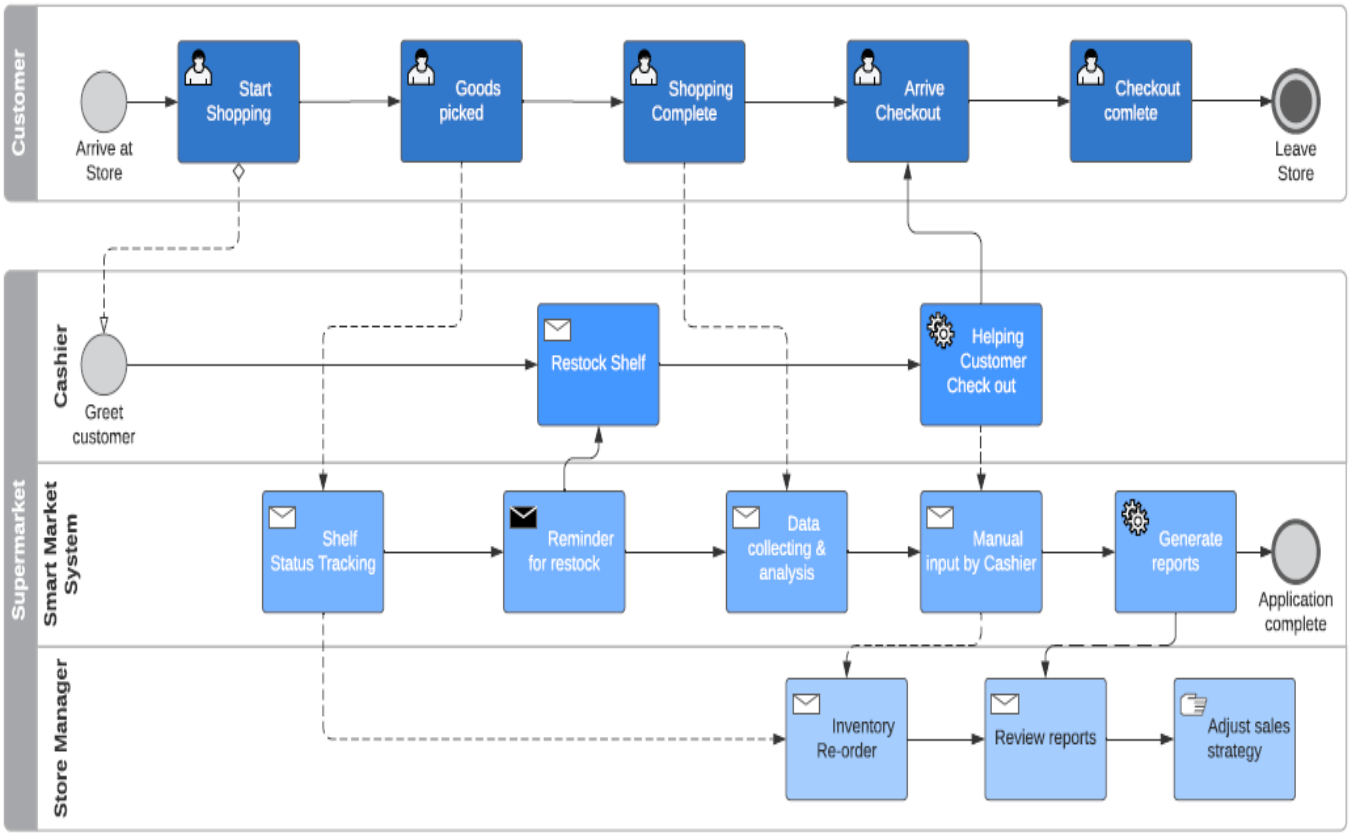
2. Before sending low inventory, expiring soon or item expired prompts the system will ask the cashier to confirm the prompt by checking inventory and items.

Ease of Use: 1. Manager can assign threshold value for every product and then check the products categorized in – None, Very Few (threshold value), Sufficient (2 times of threshold) and Full categories.

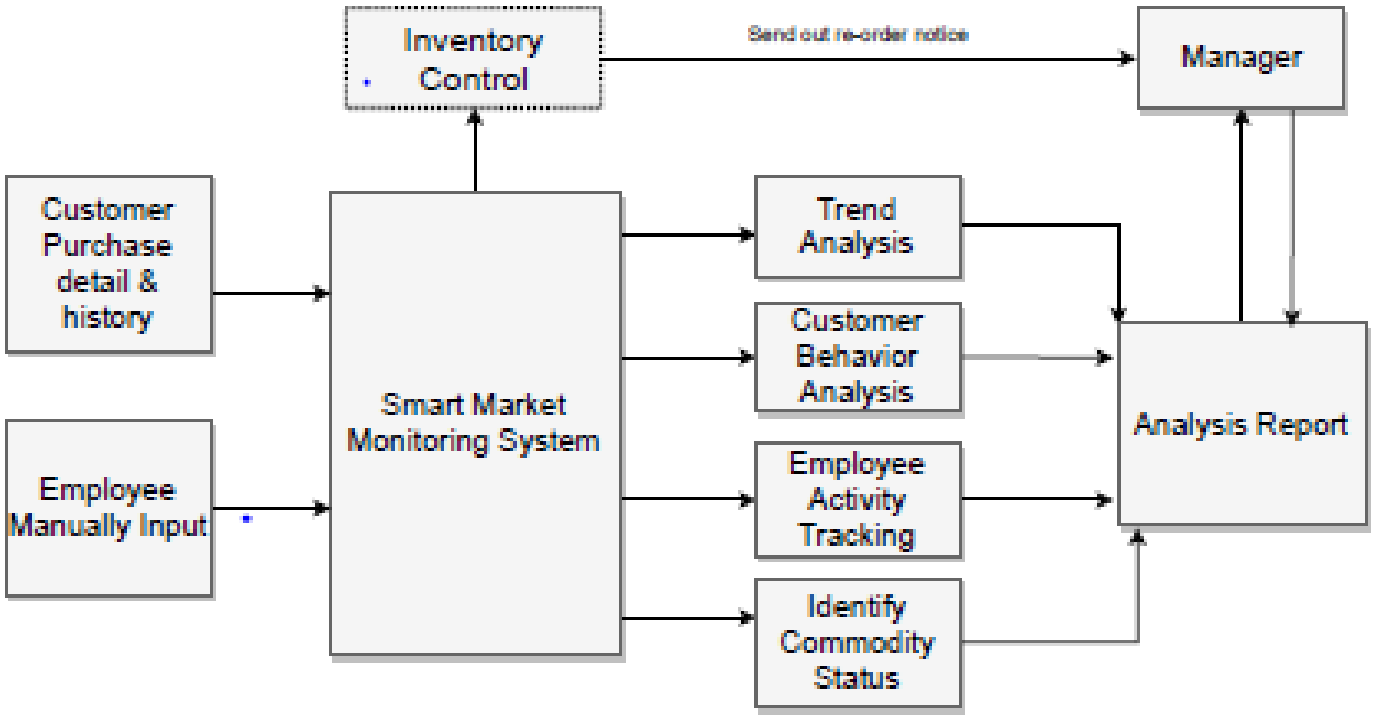
2. Manager can sort the cashiers according to Sale amount and time spent in both ascending and descending order.

3. System will give option to send SMS or e-mail prompt to manager for empty inventory and items expiring.

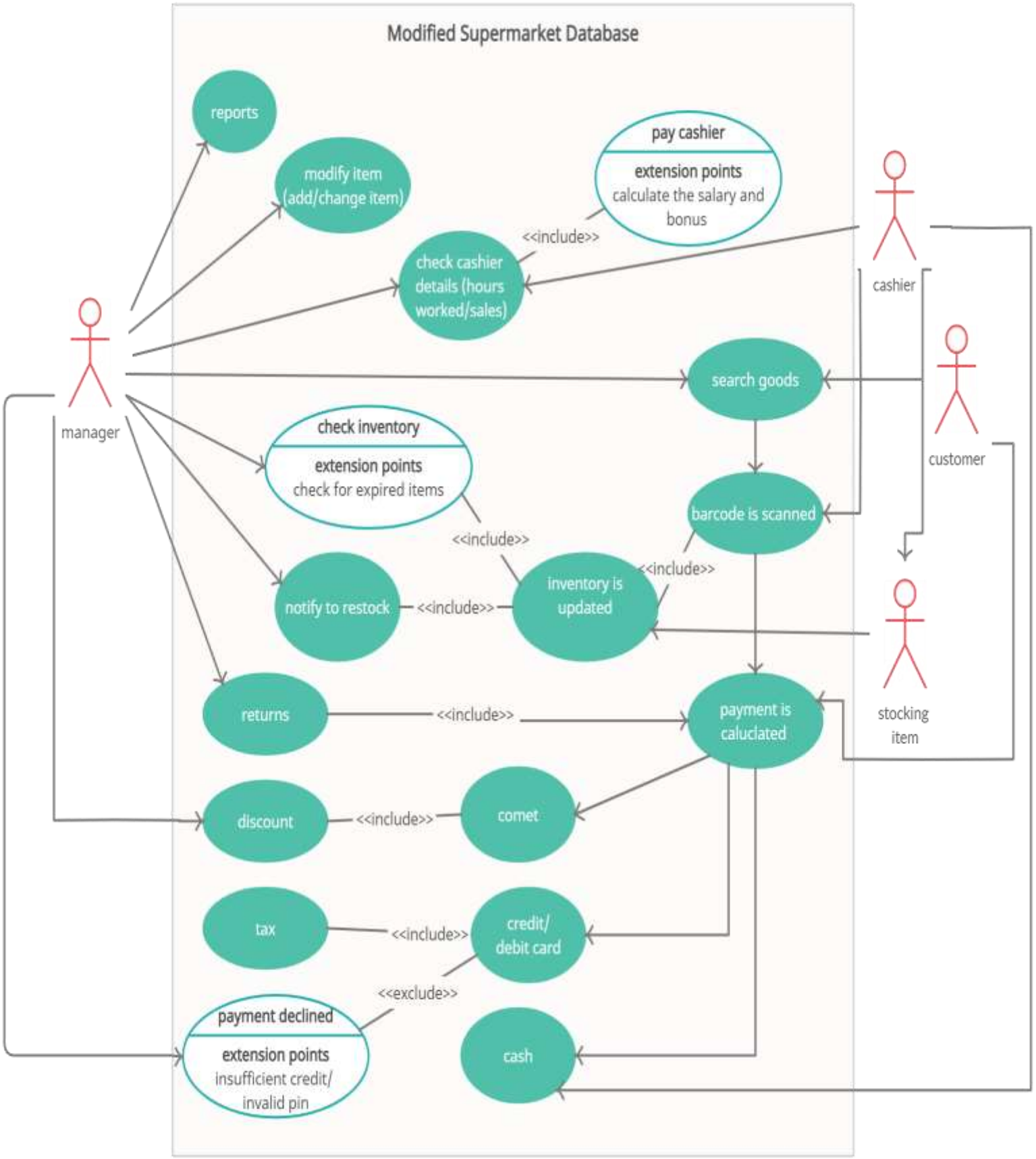
BUSINESS PROCESS MODEL USING BPMN



CONTEXT DIAGRAM FOR THE PROPOSED SYSTEM



USE CASE DIAGRAM



USE CASE DESCRIPTION

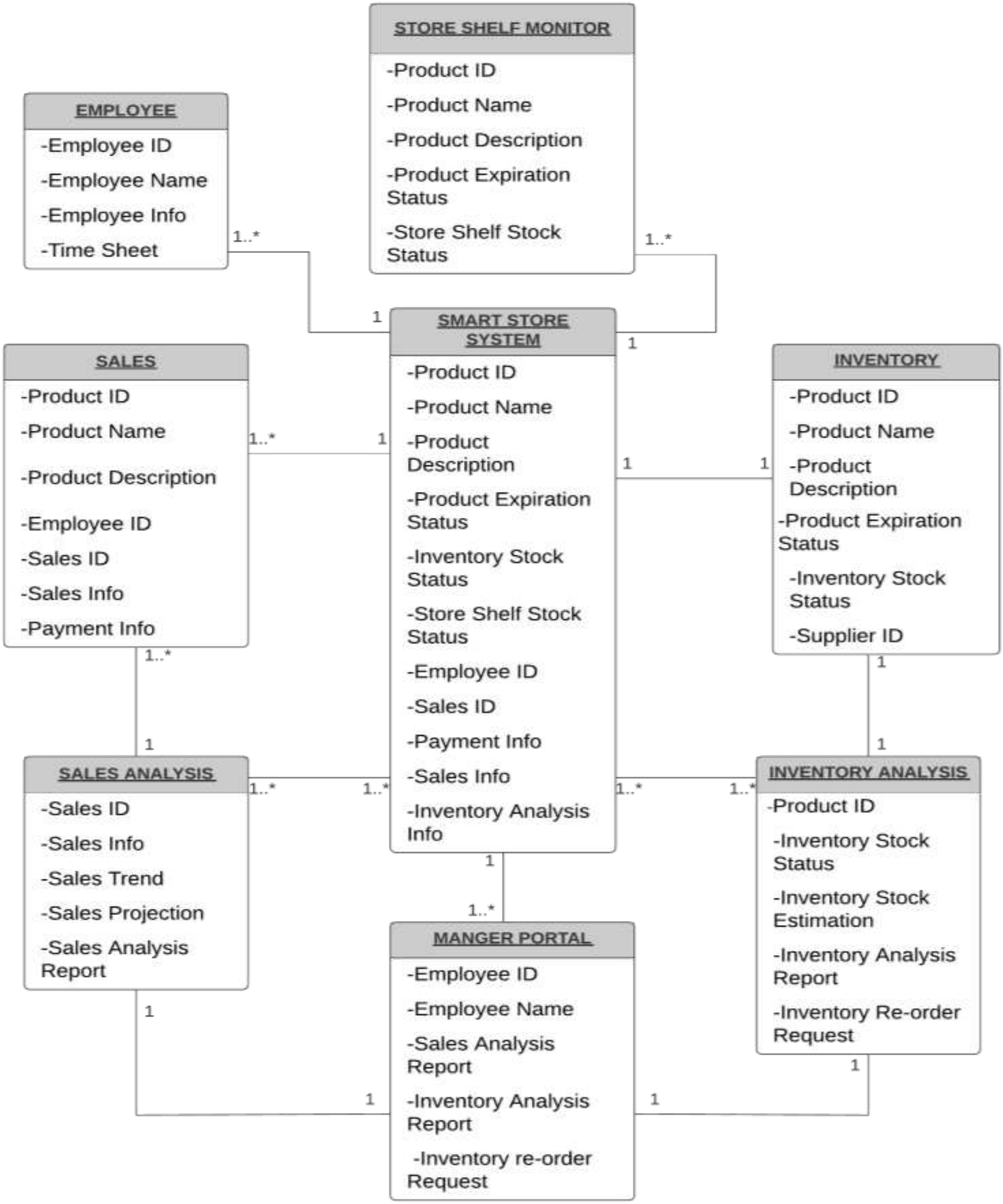
Use Case Name:	Check inventory
Primary Actor:	Customer
Brief Description:	While stocking employer updates the number of items, inventory is updated, Customer enters the market, searches for goods, brings to the counter for purchase, item is purchased, inventory is updated, if stock is less manager is notified.
Trigger:	Item barcode is scanned
Normal flow of events:	1. While stocking number of items are updated in inventory 2. Cashier scans item barcode 3. Inventory is updated (no. of items purchased are removed from inventory) 4. If the number of items is less than the threshold number 5. Manager is notified to restock the particular item
Alternate/Exception flow:	If any item is about to get expire manager is informed. (When the inventory is updated, system checks for any items about to be expired)

Use Case Name:	Pay cashier
Primary Actor:	Cashier
Brief Description:	cashier logins into system, scans the items, receives payments, restocks the inventory, cashier working hours are calculated, cashier receives the salary.
Trigger:	When manager request for cashier working hours.
Normal flow of events:	1. Cashier logins into the system 2. Cashier restocks and updates the inventory 3. Cashier scans the barcode 4. Cashier receives the payment 5. The hours of work and sales done are calculated (by manager) 6. Cashier receives the salary
Alternate/Exception flow:	When the number of sales are increased cashier may receive some bonus

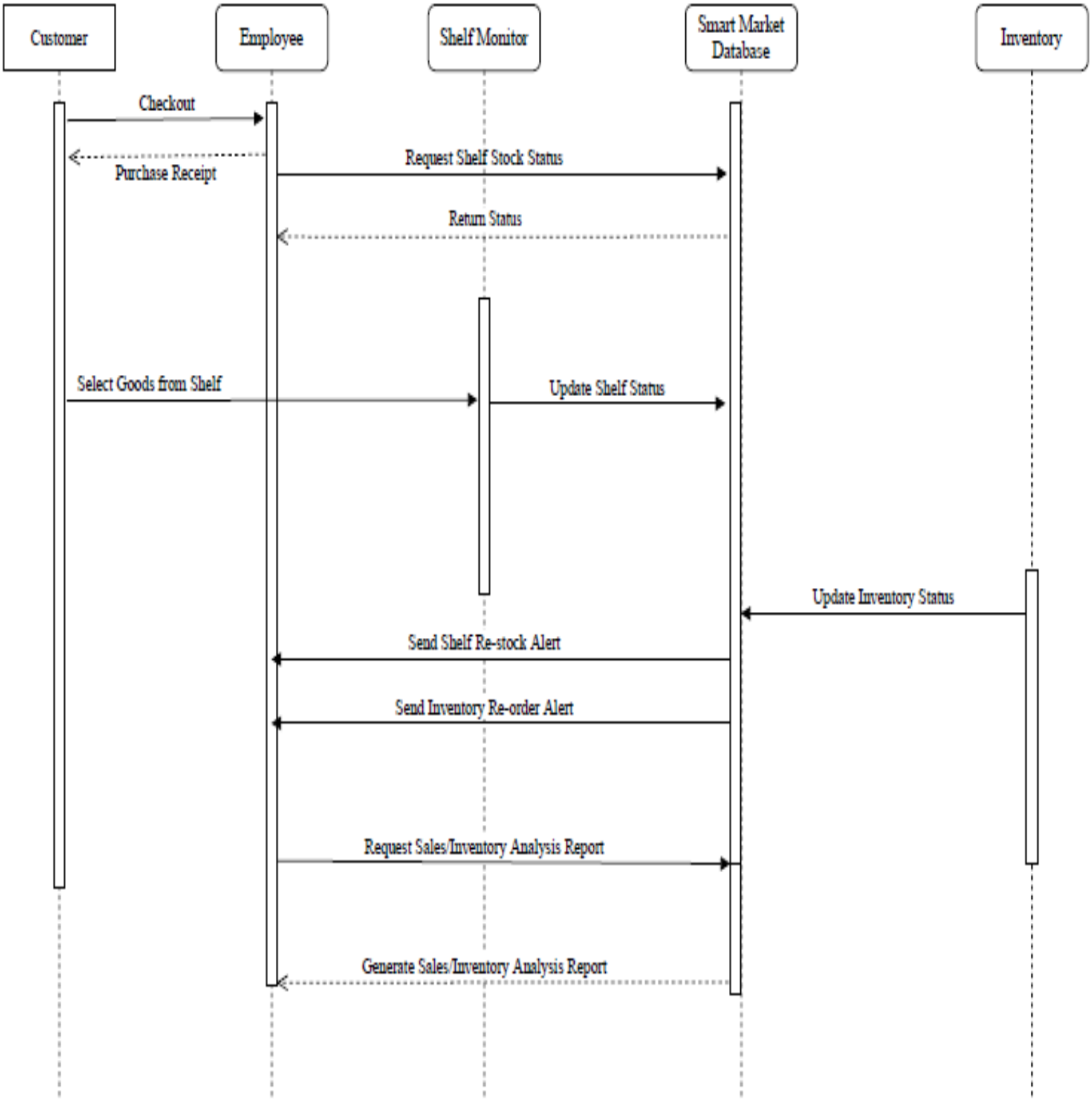
DATA DICTIONARY

- Product Description = product ID + product Name
- Store Shelf Stock Status = Product Description + Product Expiration Status
- Employee Info = Employee ID + Employee Name + Time Sheet
- Payment Info = {cash | credit | comet}
- Sales Info = {item} + Sales ID + Employee ID
- Sales Analysis Report = Sales Info + Sales Trend + Sales Projection
- Inventory Stock Status = Supplier ID + Product Description
- Inventory Re-order request = Inventory Stock Status + Inventory Analysis Report

CLASS DIAGRAM



SEQUENCE DIAGRAM



FUNCTIONAL SPECIFICATION DOCUMENT FOR THE PROPOSED SYSTEM

Proposed System Monitoring Module:

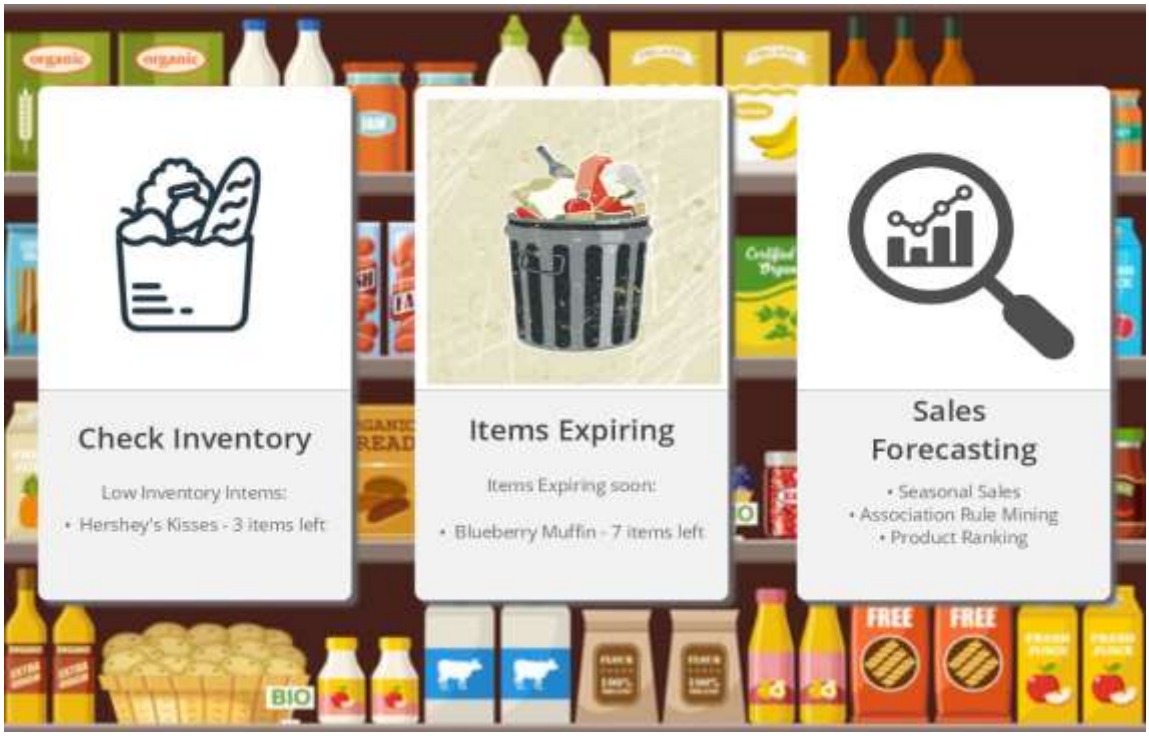
- Will track the real-time stock status of both in-store merchandise shelfs and store inventory
- Will track the expiration time of merchandise
- Will allow users to inspect the current or self-selected time period status of both in-store merchandise shelfs and store inventory
- Will send out alerts for merchandise shelf re-stock, close to expire reminder, and low inventory balance for re-ordering

Proposed System Analysis Module:

- Will track in-store merchandise shelf activities, customer purchase behavior, and inventory flow
- Will generate analysis reports:
    - Based on inventory related data, generate inventory trending & forecasting reports
    - Based on sales and customer related data, generate sales trending & forecasting reports

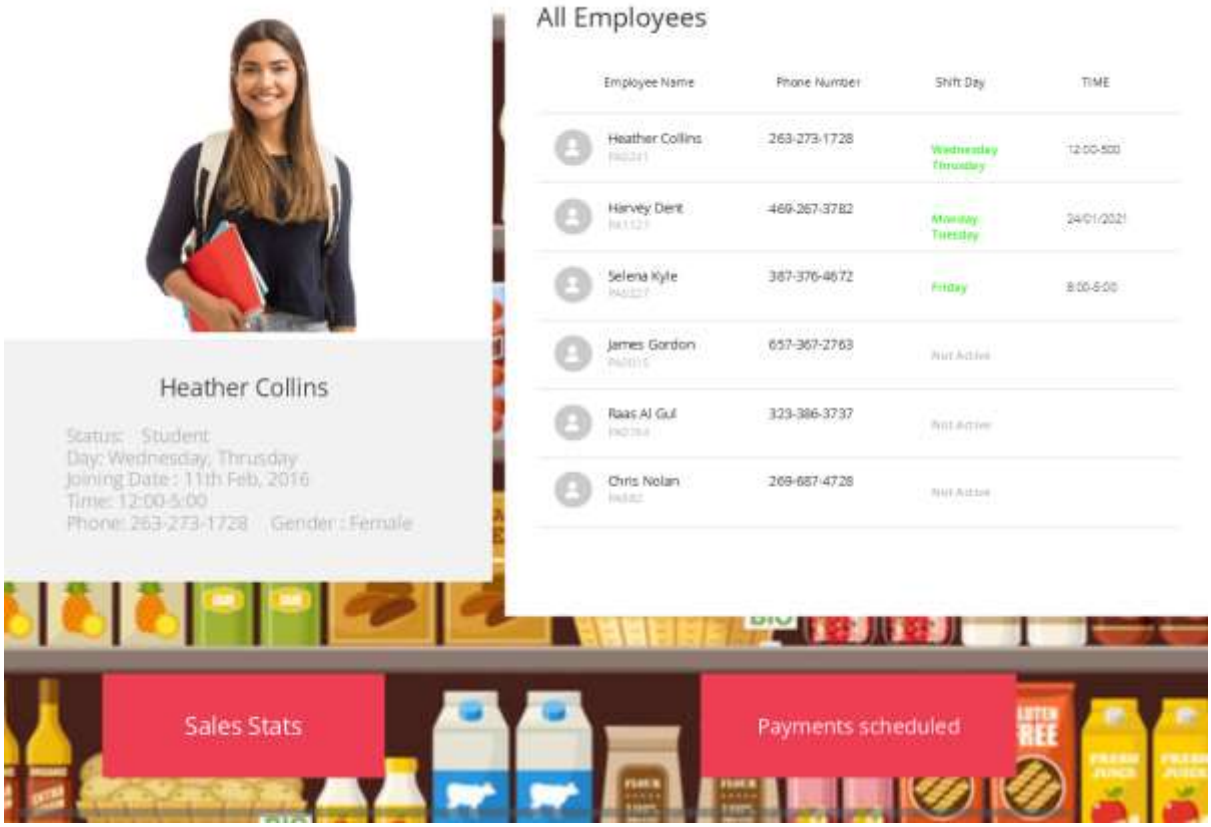
INTERFFACE DESIGN

First Page after login provides three options: Check Inventory, Check Expiring items and Analyze sales data.

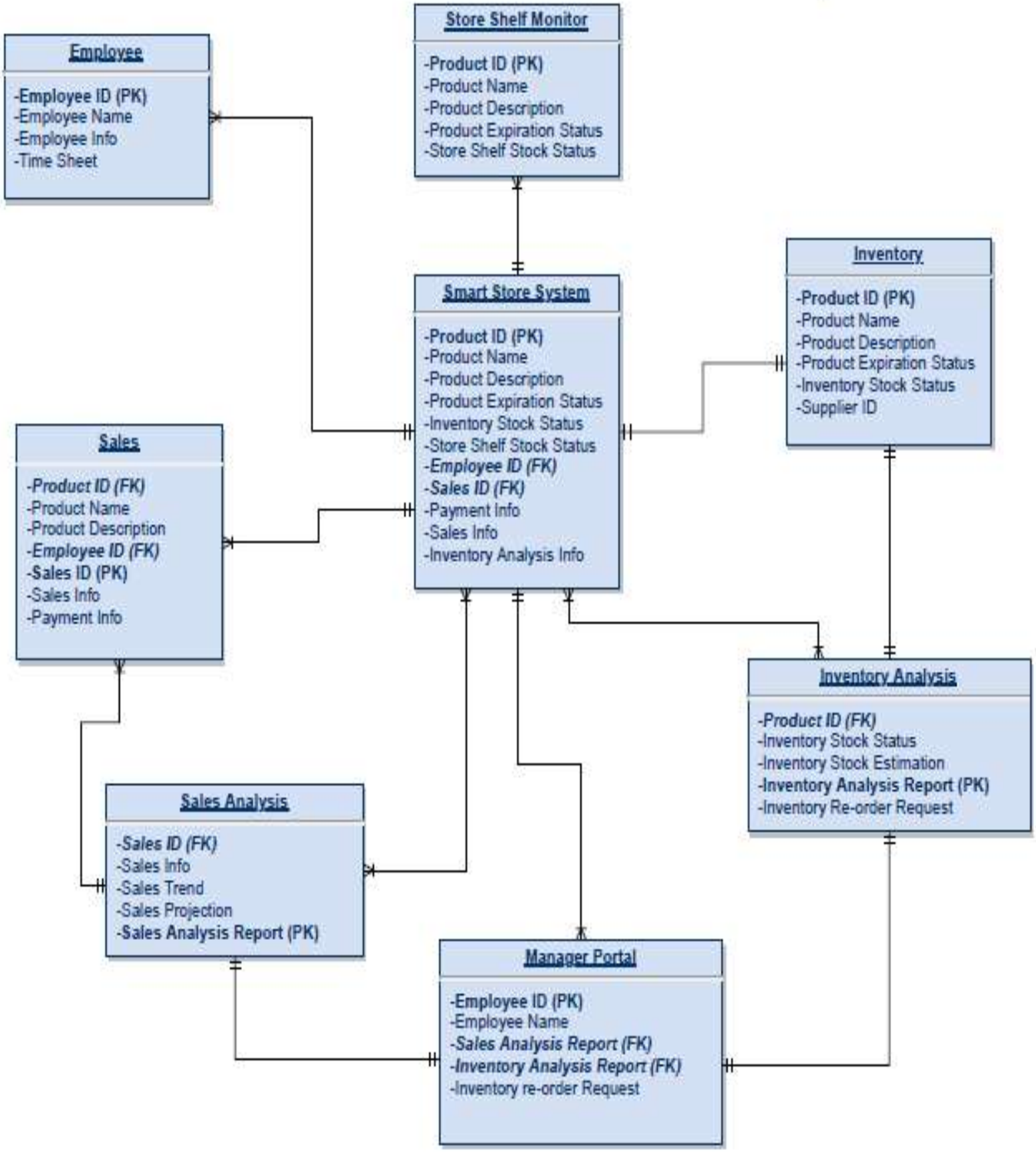


Analysing sales data provides options to analyse month-wise and employee wise:

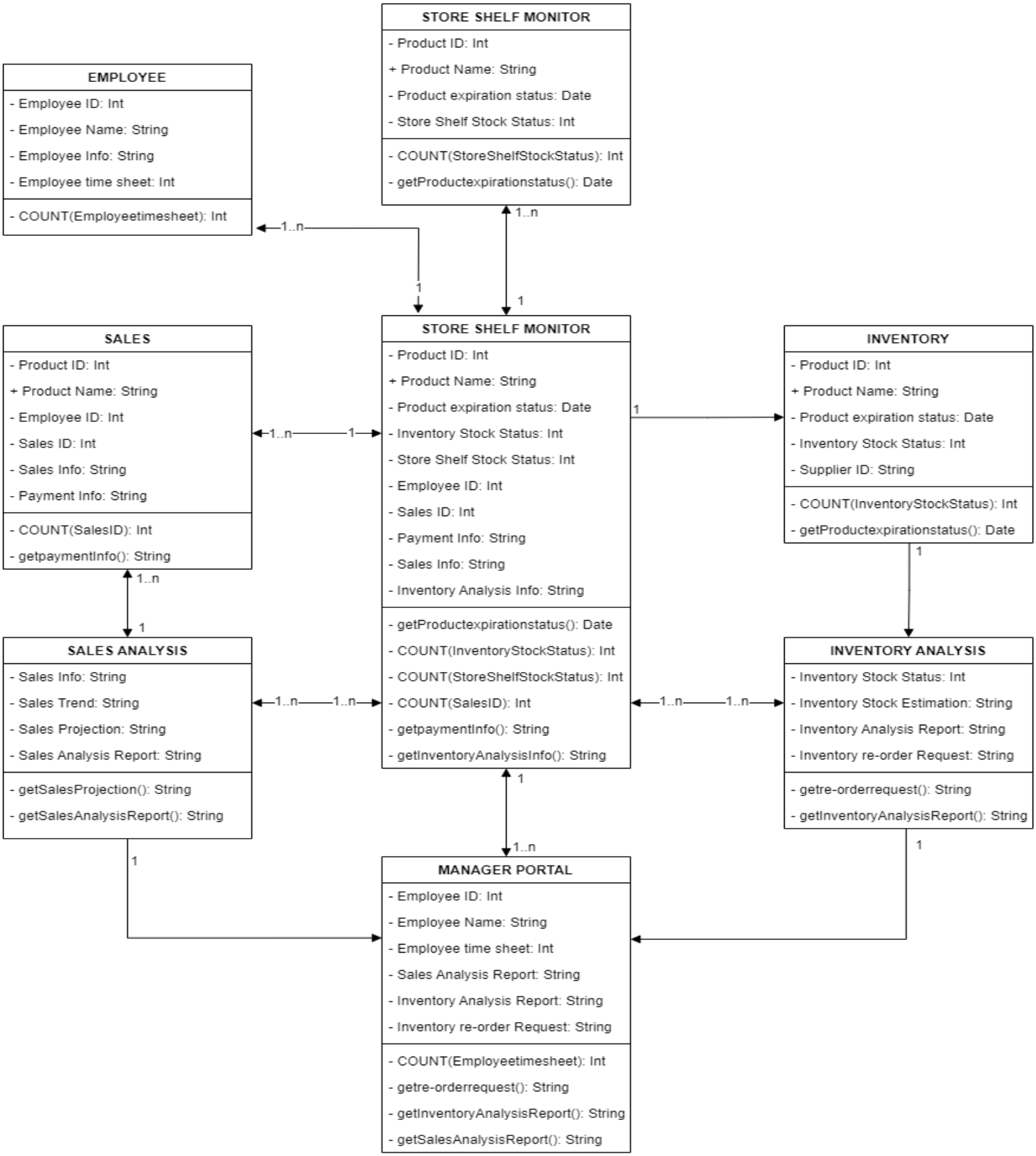
All employee details can be monitored in more user friendly way.



DATA BASE DESIGN



COMPLETE CLASS DIAGRAM



SOFTWARE DESIGN

1. Function to check expiring items:

Sample Data:

	A	B	C	D	E
1	Product Code	Product Name	Price	Quantity	Expiry
2	A001	Starbucks Triples	4	15	2022-01-15
3	A002	Starbucks Triples	4	10	2022-01-15
4	A003	Bang Caffine	3	10	2022-01-10
5	A004	Monster Energy	3	8	2022-01-20
6	A005	Smart Water	2	15	2022-02-10
7	A006	Oreo Double stu	3	20	2022-02-25
8	A007	Lays Kettle cook	2	15	2022-02-25
9	A008	Lays Cream and	2	10	2022-02-25
10	A009	Cheetos cheddar	2	15	2022-01-20
11	A010	Snickers	1	30	2022-03-10
12	A011	Trolli Gummy be	3	15	2022-02-10
13	A012	Hersheys kisses	3	3	2022-02-25
14	A013	Bubbles sparkling	2	8	2022-01-10
15	A014	AHA sparkling wa	2	7	2022-01-15
16	A015	Coke classic	2	10	2022-01-10
17	A016	Coke diet	2	10	2022-01-05
18	A017	Pepsi	2	15	2022-01-15
19	A018	Blueberry muffin	5	7	2022-01-02
20	A019	Crossaint	4	5	2022-01-05
21	A020	Cupcakes	4	5	2022-01-07
22	A021	Pistachios	7	10	2022-01-15
23	A022	Roasted Almond	7	10	2022-01-20
24	A023	Yogurt Pretzels	5	5	2022-01-10
25	A024	Mini Donuts	3	5	2022-01-10
26	A025	Assorted Truffles	5	5	2022-01-15

Code (In Python):

```
#Code to check expiration date of items and send alert.
import pandas as pd
import datetime as d
#After implementation data will be extracted from SQL server
product = pd.read_csv('Project_Dataset.csv')
expiry = product['Expiry']
#Extract today's date
today = d.date.today()
count=0

for i in expiry:
    i=i.strip(' ')
    s=i.split('-')
    exp=d.date(int(s[0]),int(s[1]),int(s[2]))
```

```
    days_left = exp-today
    s= str(days_left).split(',')
    t = s[0].split(' ')
    days_left = int(t[0])

    if days_left > 0:
      if days_left <= 30:
        name = product.at[count,'Product Name']
        amt = product.at[count,'Quantity']
        print('Alert! Product {0} is expiring in {1} days. There are {2} items in
inventory'.format(name,days_left,amt))

    else:
      name = product.at[count,'Product Name']
      amt = product.at[count,'Quantity']
      print('Warning!! Product {0} has expired. There are {1} items in inventory
Expired!'.format(name,amt))
    count =count+1
```

Result on sample data:

```
Alert! Product Blueberry muffin is expiring in 28 days. There are 6 items in inventory
```

2. Function to check inventory and send alert.

(Working on previous sample data)

Code (In Python):

```
#Code to check inventory and send alerts.

import pandas as pd

#After implementation data will be extracted from SQL server

product = pd.read_csv('Project_Dataset.csv')

count = 0

qty = product['Quantity']

for i in qty:

  if i < 5:

    name= product.at[count,'Product Name']

    print('Alert! Product {0} is low on inventory with {1} pieces left.'.format(name,i))

  count=count+1
```

Result on sample data:

```
Alert! Product Hersheys kisses is low on inventory with 3 pieces left.
```

3. Function to run association rules and extract insight.

Sample data:

	A	B	C	D	E	F	G
1	Transaction	Diet Coke	Lays Chips	Starbucks Triple Shot	Monster Energy Drink	Snickers	Blueberry Muffin
2	1	1	1	0	0	1	0
3	2	0	1	0	1	0	0
4	3	0	1	1	0	0	0
5	4	1	1	0	1	0	0
6	5	1	0	1	0	0	0
7	6	0	1	1	0	0	0
8	7	1	0	1	0	0	0
9	8	1	1	1	0	1	0
10	9	1	1	1	0	0	0
11	10	0	0	0	0	0	1
12							

Code (In R):

```
library(arules)

fp.df <- read.csv("Transaction.csv")

fp.df1<-fp.df[,-1]

as(fp.df1, "transactions")

# remove first column and convert to matrix

fp.mat <- as.matrix(fp.df1)

# convert the binary incidence matrix into a transactions database

fp.trans <- as(fp.mat, "transactions")

inspect(fp.trans)

#Apriori Algorithm

rules <- apriori(fp.trans, parameter = list(supp = 0.3, conf = 0.5, target = "rules",minlen=2))

inspect(rules)

# inspect the first three rules, sorted by their lift

inspect(head(sort(rules, by = "lift"), n = 3))
```

Results on sample data:

```
> inspect(head(sort(rules, by = "lift"), n = 3))
    lhs                          rhs                    support confidence coverage lift     count
[1] {Starbucks.Triple.Shot} => {Oreo}                  0.4     0.6666667  0.6      1.111111 4
[2] {Oreo}                  => {Starbucks.Triple.Shot} 0.4     0.6666667  0.6      1.111111 4
[3] {Starbucks.Triple.Shot} => {Lays.Chips}            0.4     0.6666667  0.6      0.952381 4
>
```

4. Function to deduct purchased products from product database:
   Code (In Python):

```
#Code to update inventory after every transaction
#Product codes taken from barcode reader
product_codes=['A004','A007','A018']

product = pd.read_csv('Project_Dataset.csv', index_col = 'Product Code')

for i in product_codes:
   product.at[i,'Quantity']=product.at[i,'Quantity']-1

product.to_csv('Project_Dataset.csv', index = False)
```

5. Function to make pie chart from sales data of products.

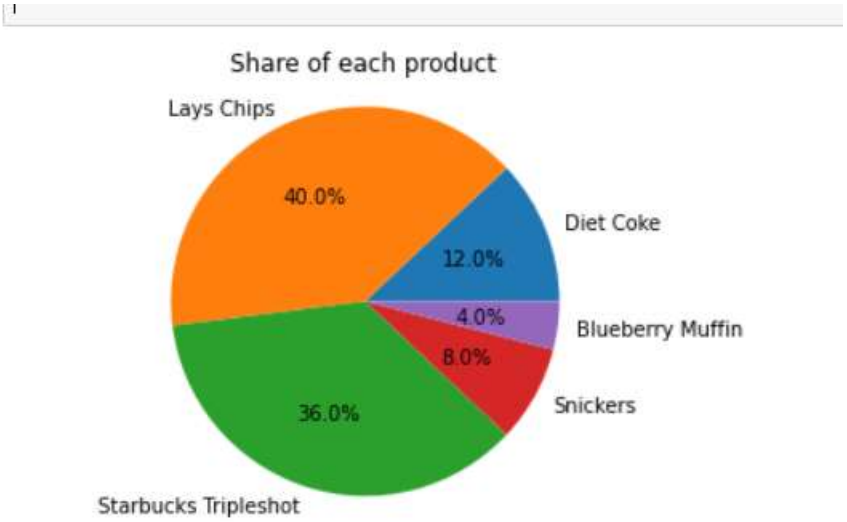
   Code (In Python):

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
labels = ['Diet Coke', 'Lays Chips', 'Starbucks Tripleshot', 'Snickers','Blueberry Muffin']
sizes = [15, 50, 45, 10, 5]

fig, ax = plt.subplots()
ax.pie(sizes, labels=labels, autopct='%1.1f%%')
ax.axis('equal')
ax.set_title('Share of each product')

plt.show()
```

   Result:



Project Management Deliverables

1. **Project Activities:**

   a) Problem Statement
   b) BPMN Diagram
   c) Context Diagram
   d) User Interface Design
   e) Data Model
   f) Use Case Diagram
   g) Code functions
   h) Database Design
   i) Use Case description
   j) Executive Summary
   k) Project Compilation

2. **Distribution of project activities:**

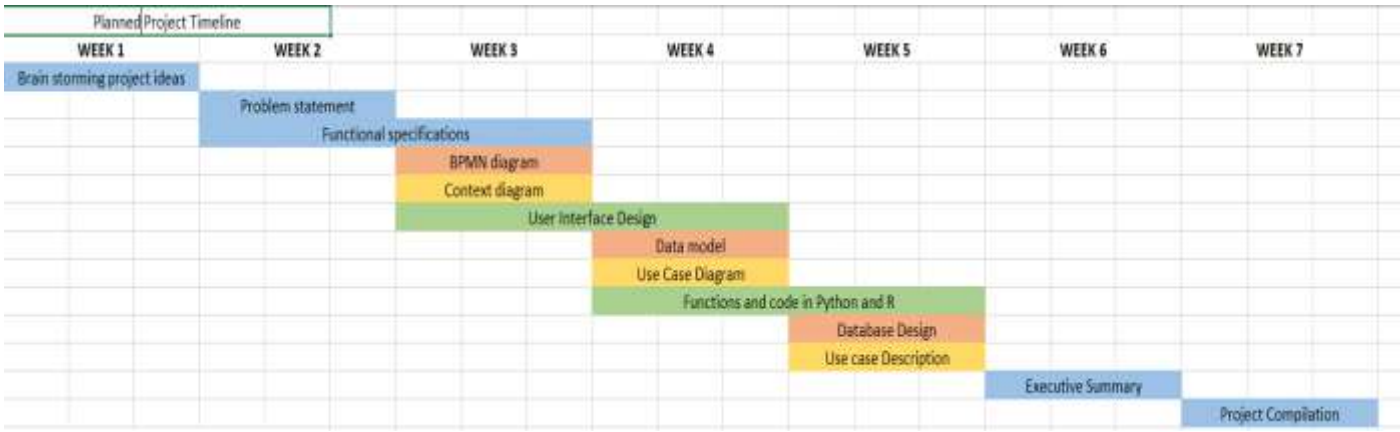
To be done in group: Problem Statement, Functional Specifications, Executive summary, Project compilation

Han Wang: BPMN diagram, Data Model, Database Design

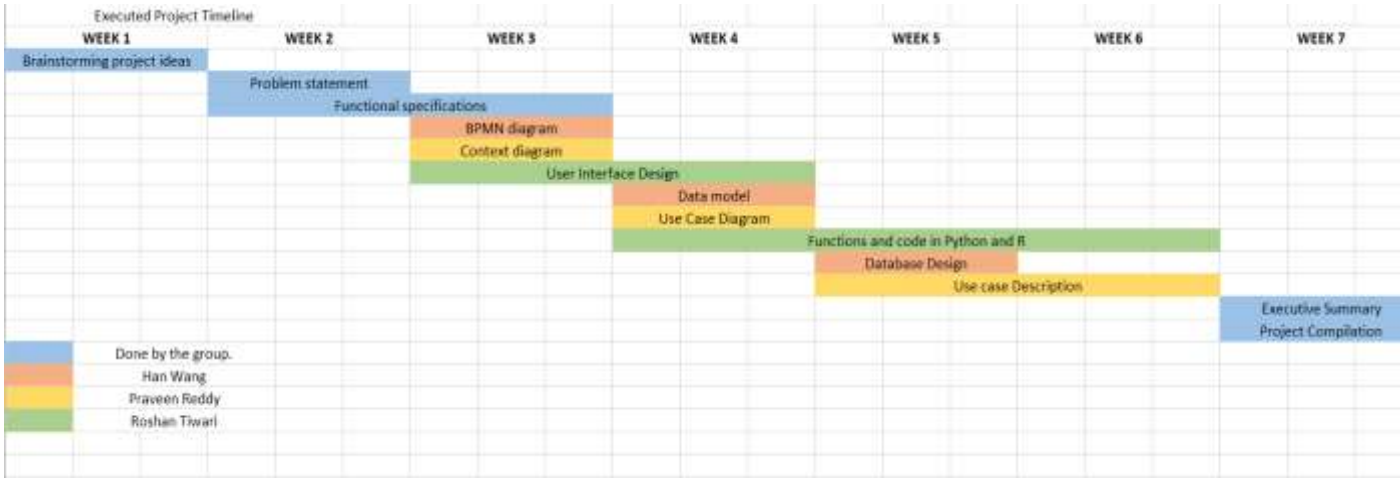
Praveen Reddy: Context Diagram, Use Case diagram, Use Case description

Roshan Tiwari: User Interface design, Coding functions

3. **Planned Timeline**



4. Executed Timeline



5. Minutes of Meeting

Meeting 1: Date-10/22/2021 Duration- 20 mins [All 3 members present]

a) Discussing project ideas and decided on the current checkout system in UTD markets.

b) Brainstorming flaws in the current model.

Meeting 2: Date-11/1/2021 Duration- 30 mins [All 3 members present]

a) Discussed and finalized functional specifications for the new model.

b) Performed sword analysis of team members and distributed project activities accordingly.

c) Created estimated project timeline.

Meeting 3: Date-11/12/2021 Duration- 30 mins [All 3 members present]

a) Created problem statement.

b) set timeline to finish first three deliverables by next meeting

-BPMN and diagram by Han

-Context diagram by Praveen

-Mock user interface by Roshan

Meeting 4: Date-11/20/2021 Duration- 40 mins [All 3 members present]

a) Discussion and feedback on context diagram BPMN and diagram and Mock UI

b) Deadline set for:

-Data model and database design by Han

-Use case diagram and use case description by Praveen

-Software algorithms and functions by Roshan

Meeting 5: Date-11/27/2021 Duration- 70 mins [All 3 members present]

a) Discussion on use case description and functions. Decision made to modify some use case descriptions and add more analytical functions.

b) Execution timeline deviated from estimated timeline more time about it to Praveen and Roshan for their deliverables

Meeting 6: Date-12/1/2021 Duration- 40 mins [All 3 members present]

a) Executive summary briefly discussed and assigned to Roshan for final completion.

b) Deadline decided for:

-Object behavior model by Han

-Complete class diagram by Praveen

Meeting 7: Date-12/7/2021 Duration- 30 mins [All 3 members present]

a) Project compiled and final project report created.