GROUP PROJECT

COMP2411

Implementation of a database for vending machines

Authors:

Sollal Fouilland, 19086822D

Michael Daniel Bigler, 21020329X

Gaukhar Turgambekova 20065366D

Yang Ke 20100884D

Alibek Kuantkhan 20041645D

Lecturer: Dr. Kevin Yuen and Prof. Quing Li

1 Abstract

The specificities of vending machines in terms of diversity of products, scalability and low requirements in terms of maintenance and space have made them extremely popular in the crowded cityscape of Hong Kong and other global metropolises. Whether it is for lunchboxes in the business districts where the salaryman doesn't have the time to stop in a restaurant or for the facemasks that are so easy to misplace, the value of the vending machine as a form of retail tool is undeniable. However, their geographical spread and variety of product they offer makes them hard to maintain at scale as many factors have to be monitored and there cannot be a single person constantly monitoring their stock and state like a manager could do in a retail store. Our database system answers this need by providing an easy, scalable and comprehensive solution to track the major variables useful to a vending machine business

2 Main Content

2.1 Business problems

Vending machines are highly adapted to the fast-paced, diverse environment of busy cities. They can have offers specifically designed for their location, very low operation costs and provide the advantage that the stock can be centralised in other places than the location of the machine. Thanks to this, warehouses can hold the stock for many machines thus reducing the storage costs. However this versatility is also what makes handling a vending machine business hard. The fact that the storage point can be spread between hundreds of retail locations, the absence of on-site employees to repair or monitor the machine and the small storage capacity forcing regular resupply of the machines makes it a business that benefits greatly from accurate data. Not being able to accurately track the thousands of daily transactions can make it fiscally hard for business owners. Traceability of the products is also very complex as they undergo multiple transfers before getting to the end client. The frequent scenario where a client

is looking for a specific product but cannot find a location that has it in stock could be answered by having a record of the stocks accessible for them to check.

2.2 Solution Design

2.2.1 Intended users

In an effort to solve those problems, we have implemented a model to be used to support the resupply, maintenance and usage of vending machines at scale. This model aims to service four type of users:

- Customers that purchase the products in the machines.
- The administrator, who is in charge of the vending machine business and needs access to most of the database for legal and workflow purposes
- The technicians of the company that operates the vending machines, they are in charge of resupplying, maintaining and repairing every machine

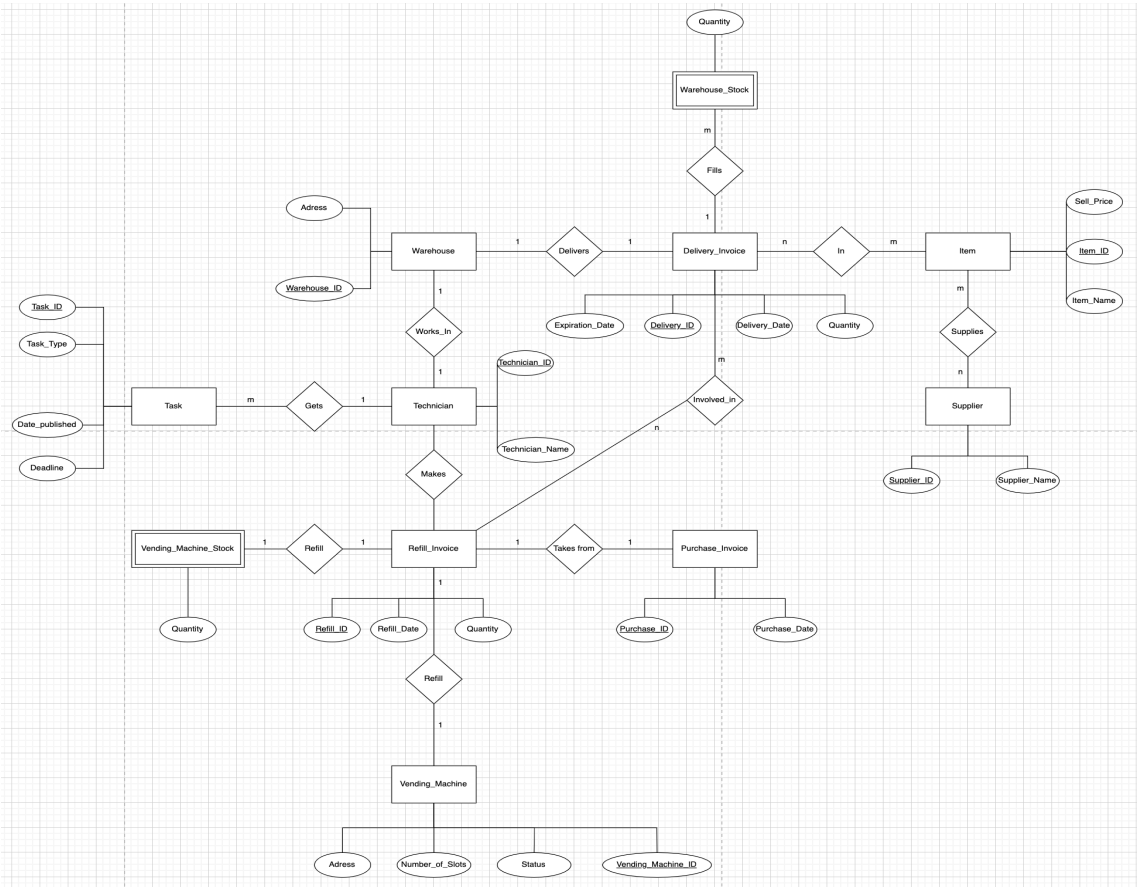
Because of those three target users, we define three different accesses and actions that they can perform on the database:

- Customer interface: The client has the most restricted access, he does not need a login, but can only view the vending machine information, their status and their stock. This would allow things like a website or an app that keeps track of every stock so that users looking for a product could check where to purchase it and whether it is in stock.
- Administrator interface: This interface offers total access over the database and is mainly intended for the owner of the vending machine business. Amongst the most important function available are:
    - Creating/Modifying/Deleting Job record to update the tasks to be done.
    - Creating/Modifying/Deleting Employee records, to keep track of current employees.
    - Creating/Modifying/Deleting Delivery invoices, to keep track of incoming supplies.

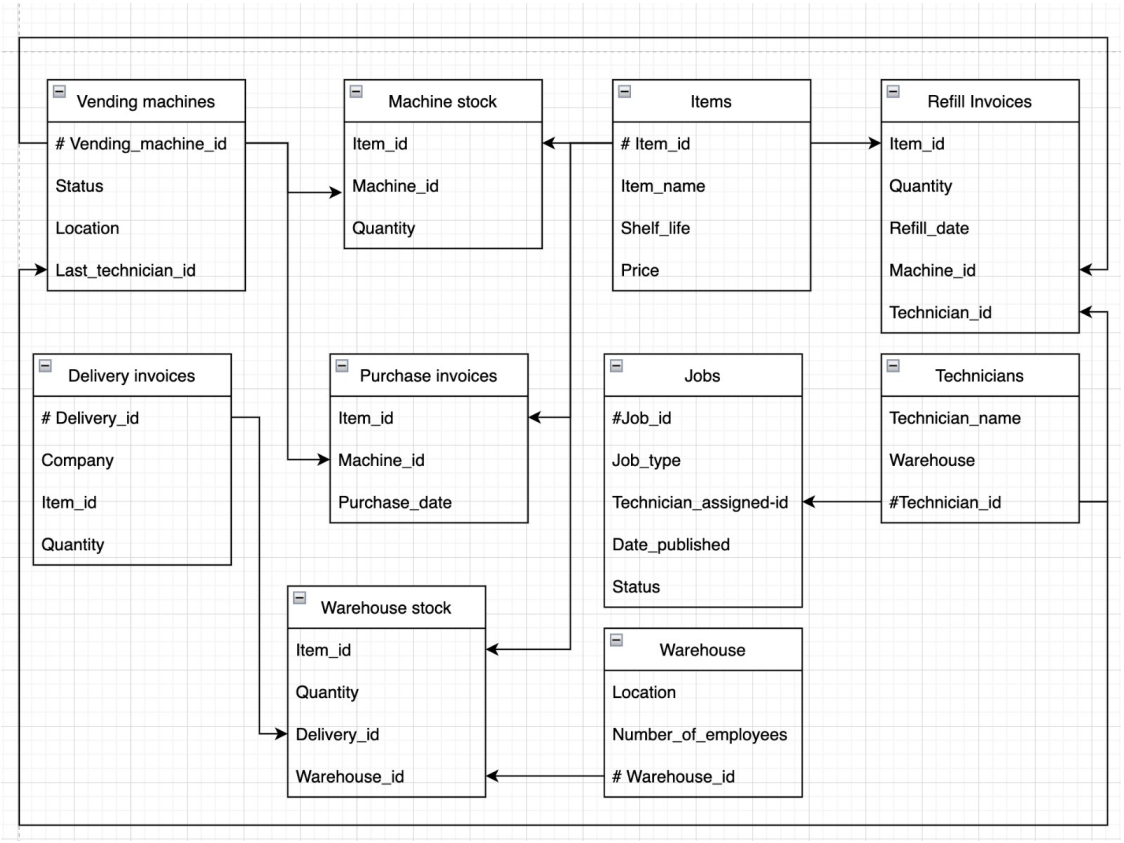
- Creating/Modifying/Deleting vending machine records, to keep track of the functioning vending machines and their location.
- Other functions are mostly designed to be updated automatically, like the stock in a vending machine.

- Technician interface: This interface is intended to be used by the technicians, it requires a log-in. It allows them to see what jobs have been assigned to them, update their status. Accessing the information on the vending machines and their stock, but also to update them. And to create/ modify/ delete refill and purchase records if needed.

Database ER-diagram:



Database Relational Schema diagram:



2.2.2 Entities

This implementation was made keeping in mind the three different types of users, practicality, scalability and the specifics of the vending machine industry.

Central to the implementation were the vending machines and the warehouses. We designed this model to make sure that restocking and servicing the machines was as streamlined as possible and that all products could be traced from beginning to end. For this purpose, we created 3 types of invoices:

- Delivery invoices: Those invoices are here to keep track of supplies provided by external suppliers, they help the administrator keep track of the source of the merchandise as well as all the payments that have been made over a time period. Those delivery invoices update the number and type of available products in each warehouse. Because they are associated with a single bulk delivery of a product, they also store the expiration date of the product and the warehouse they were delivered to.
- Refill invoices: Those are used to keep track of items moving from the warehouses to the machines. They remove the stock from the warehouse stock and refill the machine

stock. To keep track of the expiration date and from which delivery they are coming from, the Delivery_ID linking to the invoice is also stored in the Refill_Invoice. In addition, the refilled machine is also recorded.

- Purchase invoices: Those are used to record the purchases at each machine to the customers. They are the sales record and are mostly used to keep track of the sales of the company. They also store the Refill_ID that they came from, thus ensuring end-to-end traceability of the products in case any issues arise.

To support this system, we created a few more entities.

- Warehouses: Those keep track of the location working of each warehouse.
- Warehouse stock: This keeps track of the stock at each warehouse so that resources can be allocated accordingly, for example if a product is out of stock or running out, the refilling of the machines can be adapted accordingly.
- Vending machines: Those are the actual vending machines with their location and status (In service/ Out of order/ Empty/ In Repair) as well as the warehouse they are associated with.
- Vending machine stock: This keeps track of the actual stock inside each machine and the items and quantity available. This way products that are out of stock can be quickly signaled
- Item: This is used to keep track of all the possible types of items that may be either in the warehouse or in the machines. This way key information like name, supplier, and selling price can be recorded, modified, and accessed at any time and fast.
- Suppliers: This is used to keep track of the suppliers, because multiple suppliers may supply the same product (i.e cola drink for PepsiCo and The Coca-Cola Company), it is important to keep track of the suppliers.

Having implemented that much, we decided to expand the system to facilitate the administration of the business and the allocation of jobs to on-site technicians. The following two entities were created:

- Technicians: This records every on-site employee so that they may be easily identifiable, and also records their assigned warehouse. While not constricted to that, we believe that warehouses define “areas” by the geographical area that they are in. This way, a machine is serviced by the closest warehouse and it’s technicians. This avoids unnecessary transit to resupply a machine that is closer to another warehouse but also serves as a delimitation of the reach of the technicians. While not absolutely

constrained by it, it makes more sense for technicians to service the machines in the same area as their warehouse.

- Jobs: Because the refills and servicing of the vending machines are not always regular (refill might be more regular during the weekends or festivity periods and servicing is dependent on how often the machines break down) we have created the job entity. A job consists of a description, a machine id, a technician id and a status. The description and the machine id refer to the nature and location of the job, the technician id is the technician that is in charge of this repair and the status records whether the job is done, in progress or not done. This system allows the administrator to assign jobs to its technicians based on their route, number of jobs that they have to do or other factors. It could also be used to implement quotas where technicians choose their own tasks and are paid per task. The flexibility of this system makes it highly versatile.

2.2.3 Design choices

Because this system is meant to be implemented for a real life situation , there are a few design decisions that were made specifically for practicality reasons:

- Duplicate information between the invoices and the stocks. Our original decision was to keep track of the machine stock and warehouse stock exclusively through the invoices. While this would have staved us from having to create the “stock” entities, in practicality, the volume of invoices for a big scale vending machine business would have made the repetitive calculation of stock for each machine extremely costly, especially if done every time a technician or a customer wants to access it. In this way, while the stock does need to be updated with every delivery, refill and purchase, we believe it is still more practical than having to add up the hundreds of thousands of receipts every time a stock has to be calculated. This does incur a cost to the design of the database which was slightly ‘denormalized’, however the savings in computational work incurred by this change justify it.

2.3 Development and implementation

In this part of the report the focus lays on how the group developed and implemented the database. Mainly techniques out of the lectures were used for this.

2.3.1 Database tier

After having a final ER-diagram and relational schema the implementation of the database tier is rather easy. The tables needed can be read out of the schemas as well as the primary keys. In total ten tables were created in SQL and filled up with random data for testing purposes. Most of the work consisted of reconciling the real life constraints of SQL with the theoretical concepts of the ER-diagram. This forced multiple back and forth adaptation of the model to get something truly functional. In addition, restricting the relation attributes to avoid null or non-atomic values where there shouldn't be was helpful in rendering the database more resilient.

2.3.2 Client tier

For the client tier, we developed a java application with four classes. One is the entrance of the database, and the rest correspond to different clients.

- Class Clients: This is the main class of the application where the connection to the Oracle database is made and the user's identification is declared. Three cases are considered in terms of the identity of users by switch structure. The identity cannot be changed as long as the program is executing while for each identity, it keeps receiving different commands. Technicians need to provide their id number to proceed while the Customer and Administrator have no such procedure.
- Class Customer: This class is used to cater the needs of the customer, which enables users to check the addresses of all machines that contain a desired item and check the stock of a vending machine with a certain address.
- Class Technician: Once a technician has entered the id number, it is passed into an object of this class with multiple functions. In this class, technicians can check their refill invoice and get a list of jobs assigned to them. Besides, technicians can also set a job as finished.
- Class Administrator: In this class, administrators can both make all-rounded queries or updates to the vending machine database. For queries, they are divided into three main parts: vending machines, warehouse, and Task & Technicians. Besides, it also has an option for administrators to enter their own query. For making updates to the vending machine database, administrators are allowed to add more machines, tasks to be done, suppliers, and technicians… They can also delete any tuples from any tables.

2.4 Testing and Validation

2.4.1 Database

To test the SQL database, we added made up data by generating a few of the tuples manually as well as using randomisers to initialise the variables. By properly designing the randomiser and constricting the possible values for the random variables, we are able to generate a believable set of data. In addition, setting the seed before running the data generation makes the generation repeatable. All of those operations are done in the TestingValuesGenerator.sql file. Once the data was created, we were able to test out the SQL queries to ensure that everything worked as intended.

2.4.2 Client tier

For validation, the client tier runs on apollo2 server. Here's the screenshot of the starting interface of the Application Clients.

```
20100884d-apollo2:/home/20100884d/src$ java -classpath ojdbc7.jar:. Clients
Enter your username: 20100884d
Enter your password:
1 -->> Customer
2 -->> Technician
3 -->> Administrator
-1 -->> Exit
Please enter the number of your position:1
1 -->> Get addresses of vending machine that has a certain item
2 -->> Get a list of items in a vending machine with a certain address
-1 -->> Exit
Please select the type of query you like to make:
```

For testing, we use TestingValuesGenerator.sql to implement test data and then run the Clients application.

3 Learning experiences

This project has been very lengthy as it required the conceptualisation and implementation of a functional product that was very unfamiliar to us. It required learning many new things to be functional and caused us to spend a lot of time figuring out how to implement the project rather than on the conceptualisation of the project itself.

3.1 Cooperation and management

Since the project was carried out by a small team, it wasn't necessary to give leadership roles to anyone. We conceptualise the idea together and split up the task depending on

everyone's expertise. Some of us were not familiar with coding and as such were asked to deal with the design of the database and the writing of the report. For others, the work was split in the classical back-end, front-end division of labour. This decision was made because the initial implementation of the front-end did not require a final version of the database design to be ready and the back end had to meet a few requirements to fit the front end, but was overall very flexible. This was also made because the front-end was handled through java and the back-end through SQL.

3.2 Best practices

Regular updates from all members are very important. A centralised access to all the code files through github was also very practical in keeping all the members updated on the latest version of the code and the report. Setting a report up in the beginning of the project was also very helpful as continuous work is less effort than writing the whole report at the end of the project. As soon as the report was in its final form the team could quickly develop the slides needed as all the information was present.

3.3 Challenges and Solution

The main challenges that arose from this project were:

- Bridging the gap from java to SQL: This was initially very complex as most of us were unfamiliar with projects that connected multiple coding languages together. This was solved by gaining knowledge of the JDBC API that helped us bridge that gap.
- Arriving at a satisfying database design: Because of the open-endedness of the project, and the lack of practice most of us had in designing a database from the ground up, we had to iterate many times over the design of the database to arrive at the current one. This was solved by repeatedly analysing the database design through the lenses of best practices and normalisation forms, but also from the perspective of a real-life implementation.
- Condensing information to fit into the page limits: As there was much to be learnt by this project a lot of information got accumulated. Fitting all this into a report meant a lot of rewriting and focusing on the most important statements and facts.

4 Discussion and Conclusion

In conclusion, this project was challenging and pushed us to learn many new things about the actual implementation of databases. We believe we provided a functional implementation that could bring many benefits to a vending machine business. While not absolute, this design satisfies the vital requirements that are needed for it to be robust and helpful. Having the customers being able to access storage data of the vending machines will ensure satisfied customers whenever they decide to buy goods from a vending machine.

When looking back at the project the used approach seems to be the best one possible. Finalising a database design and staying with it is vital as everything depends on the design itself and changing it would mean a lot of work to be redone. For a future project it would be advisable to push the deadline for a final database design as early as possible as it would create more time to develop the presentation tier.

Further improvements could include:

- Implementation of a supplier interface: For the supplier to have access to the records of their sales to the company.
- Development of an interactive app for the customers to access the storage data and even reserve custom products for themself.

Appendix

Running Database Michael Daniel Bigler:

```
21020329x-apollo:/home/21020329x$ source /compsoft/app/oracle/dbms.bashrc
21020329x-apollo:/home/21020329x$ sqlplus

SQL*Plus: Release 12.1.0.1.0 Production on Mon Nov 29 00:40:19 2021

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter user-name: "21020329X"@dbms
Enter password:
Last Successful login time: Mon Nov 29 2021 00:36:56 +08:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL*PLUS:21020329X>DESCRIBE ITEM;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_ID                                   NOT NULL NUMBER
 ITEM_NAME                                 NOT NULL VARCHAR2(50)
 SUPPLIER                                           NUMBER
 SELL_PRICE                                NOT NULL NUMBER

SQL*PLUS:21020329X>SELECT ITEM_ID FROM ITEM;

   ITEM_ID
----------
         1
         2
         3
         4
         5
         6
         7
         8
         9
        10
        11
```

Running Database Sollal Fouilland:

```
Enter user-name: "19086822D"@dbms
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied


Enter user-name: "19086833D"@dbms
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied


Enter user-name: "19086822D"@dbms
Enter password:
Last Successful login time: Mon Nov 29 2021 01:05:54 +08:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL*PLUS:19086822D>describe item;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_ID                                   NOT NULL NUMBER
 ITEM_NAME                                 NOT NULL VARCHAR2(50)
 SUPPLIER                                           NUMBER
 SELL_PRICE                                NOT NULL NUMBER

SQL*PLUS:19086822D>select * from item;

   ITEM_ID ITEM_NAME                                          SUPPLIER
---------- -------------------------------------------------- ----------
SELL_PRICE
----------
         1 Cola Drink                                                  1
        10

         2 Cola Drink                                                  5
       9.8

         3 Oreo                                                        3
       6.4


   ITEM_ID ITEM_NAME                                          SUPPLIER
---------- -------------------------------------------------- ----------
SELL_PRICE
----------
         4 Lemon Tea                                                   4
         6
```

Running Database Alibek Kuantkhan:

```
[
[Enter user-name: 20041645D@dbms
Enter password:
Last Successful login time: Mon Nov 29 2021 02:19:20 +08:00

Connected to:
[Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
[
SQL*PLUS:20041645D>select * FROM item;

   ITEM_ID ITEM_NAME                                          SUPPLIER
---------- -------------------------------------------------- ----------
SELL_PRICE
[----------
         1 Cola Drink                                                1
        10

         2 Cola Drink                                                5
       9.8

         3 Oreo                                                      3
       6.4


   ITEM_ID ITEM_NAME                                          SUPPLIER
---------- -------------------------------------------------- ----------
SELL_PRICE
----------
         4 Lemon Tea                                                 4
         6

         5 Soy Milk                                                  4
         8

         6 Barbecue Flavoured Chips                                  5
      10.4


   ITEM_ID ITEM_NAME                                          SUPPLIER
```

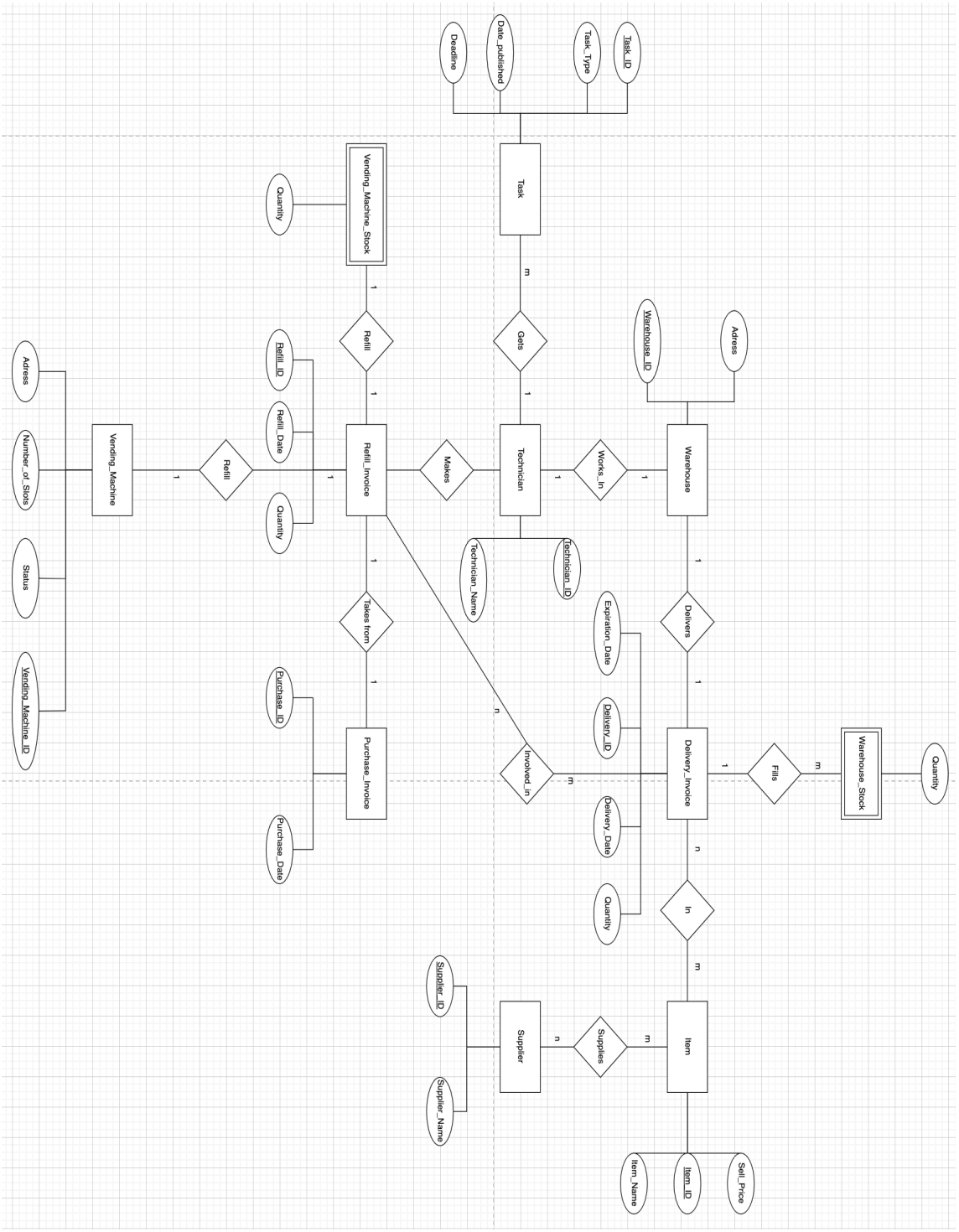
Running Database Yang Ke:

```
SQL*Plus: Release 12.1.0.1.0 Production on Mon Nov 29 02:27:53 2021

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter user-name: 20100884d@dbms
Enter password:
Last Successful login time: Mon Nov 29 2021 02:27:04 +08:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL*PLUS:20100884D>describe Item
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_ID                                   NOT NULL NUMBER
 ITEM_NAME                                 NOT NULL VARCHAR2(50)
 SUPPLIER                                           NUMBER
 SELL_PRICE                                NOT NULL NUMBER

SQL*PLUS:20100884D>Select ITEM_ID,ITEM_NAME From Item;

   ITEM_ID ITEM_NAME
---------- --------------------------------------------------
         2 Cola Drink
         3 Oreo
         4 Lemon Tea
         5 Soy Milk
         6 Barbecue Flavoured Chips
         7 Cream and Onions Flavoured Chips
        13 Sports Drink
        14 Cheese Biscuits
         1 Cola Drink
         8 Orange Juice
         9 Peach Flavoured Iced Tea

   ITEM_ID ITEM_NAME
---------- --------------------------------------------------
        10 Iced Coffee
        11 Iced Coffee Full Roast
        12 Iced Coffee Rich
        15 Water

15 rows selected.

SQL*PLUS:20100884D>
```

Running Database Gaukhar Turgambekova:

```
Enter user-name: "20065366D"@dmbs
Enter password:
ERROR:
ORA-12154: TNS:could not resolve the connect identifier specified


Enter user-name: "20065366D"@dbms
Enter password:
Last Successful login time: Mon Nov 29 2021 04:26:47 +08:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL*PLUS:20065366D>describe item;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_ID                                   NOT NULL NUMBER
 ITEM_NAME                                 NOT NULL VARCHAR2(50)
 SUPPLIER                                           NUMBER
 SELL_PRICE                                NOT NULL NUMBER

SQL*PLUS:20065366D>select * from item;

   ITEM_ID ITEM_NAME                                          SUPPLIER
---------- -------------------------------------------------- ----------
SELL_PRICE
----------
         1 Cola Drink                                                 1
        10

         2 Cola Drink                                                 5
       9.8

         3 Oreo                                                       3
       6.4


   ITEM_ID ITEM_NAME                                          SUPPLIER
---------- -------------------------------------------------- ----------
SELL_PRICE
----------
         4 Lemon Tea                                                  4
         6

         5 Soy Milk                                                   4
```

ER Diagram


Deadline
Date_published
Task_Type
Task_ID
Task
m
Gets
1
Technician
Technician_Name
Technician_ID
Works_In
1
Warehouse
Warehouse_ID
Adress
1
Delivers
1
Delivery_Invoice
Expiration_Date
Delivery_ID
Delivery_Date
Quantity
n
In
m
Item
Item_Name
Item_ID
Sell_Price
Fills
m
Warehouse_Stock
Quantity
Involved_in
m
n
Supplies
m
n
Supplier
Supplier_ID
Supplier_Name
Vending_Machine_Stock
Quantity
1
Refill
1
Refill_Invoice
Refill_ID
Refill_Date
Quantity
1
Takes from
1
Purchase_Invoice
Purchase_ID
Purchase_Date
Makes
Refill
1
Vending_Machine
Adress
Number_of_Slots
Status
Vending_Machine_ID

SQL database deployment file.

Included in the code file: vendingmachine.sql

SQL database data generator

Included in the code file: TestingValuesGenerator.sql

Java files for the different interfaces for the users:

Included in the code file:

Technician.java

Clients.java

Administrator.java

Customer.java