KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DEPARTMENT OF INFORMATION & COMPUTER SCIENCE
ICS321: DATABASE SYSTEMS

Project Phase I

Group Number -6-

Submitted by:

Khalid Al-Baqami – 201934770

Saud Al-Faris – 201946230

Sultan Al-Moammar - 201928890

Friday 11/18/2022

Database Requirements

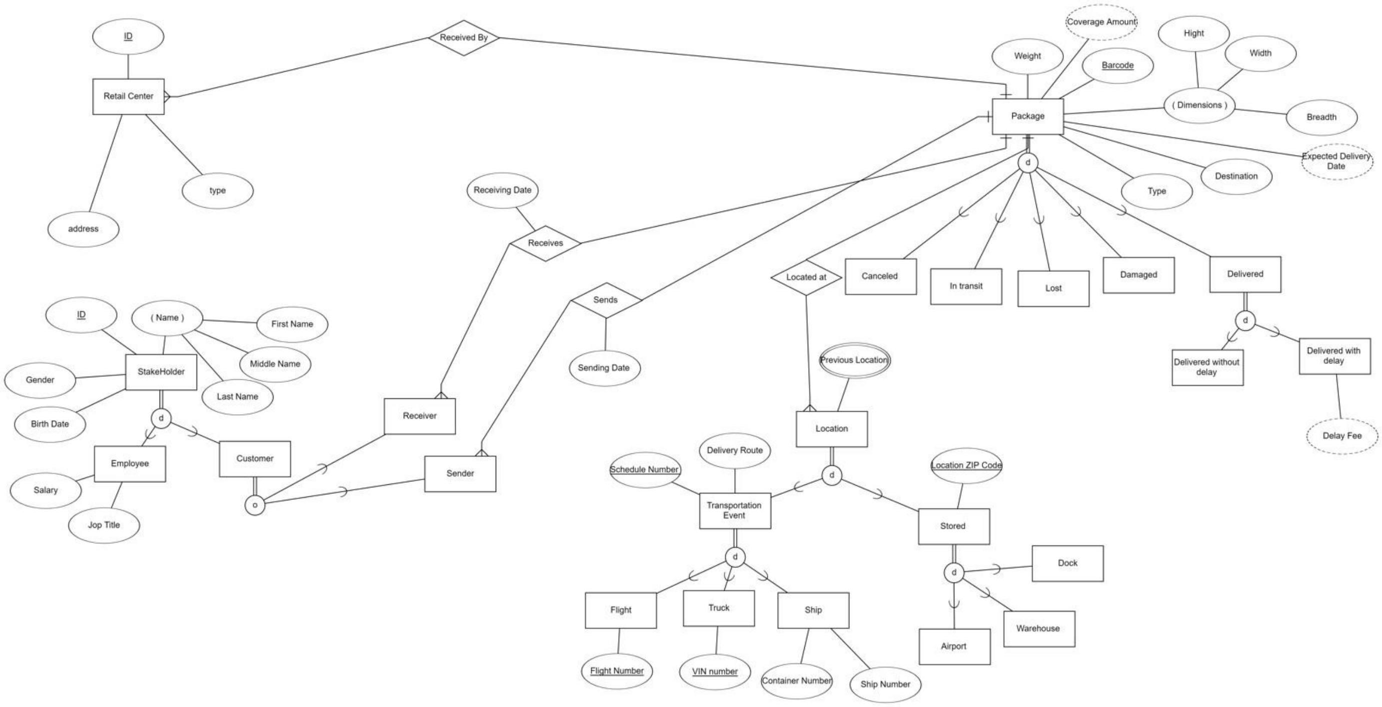
The database should implement the following entities:

- Stakeholder, which is the generalized concept of any person of relevance. It has a name, birthday, and gender, and a unique identifier (ID). It is recognized as either a customer, or an employee.
    - An employee has additional attributes, specifically salary and job title.
- Location, which tracks a package's full route. It has a unique identifier (barcode) and links to the previous location. It is either in store or in transport.
    - A transportation event is a dynamic location, and has schedule number as a unique identifier, it can be either:
        - Via ship, which stores container and ship number as attributes.
        - Via truck, which stores VIN number as attribute.
        - Via flight, which stores flight number as attribute.
    - Storage is identified with the ZIP code of the storage, it can be either:
        - In Dock
        - In Airport
        - In Warehouse
- Retail Center; which has a type, address, barcode, and a unique identifier (ID).
- Package, which is the core entity of the database. It is sent by a customer and received by another customer; it is also sent through a specific retail center. A package has height, width, breadth, weight, type, and destination as attributes. ○ A package is either delivered with delay or delivered without delay. ○ An undelivered package is either canceled, lost, in transit, or damaged.
    - A package is linked to a location, which keeps stores its current location and tracks its previous location to present a package route.

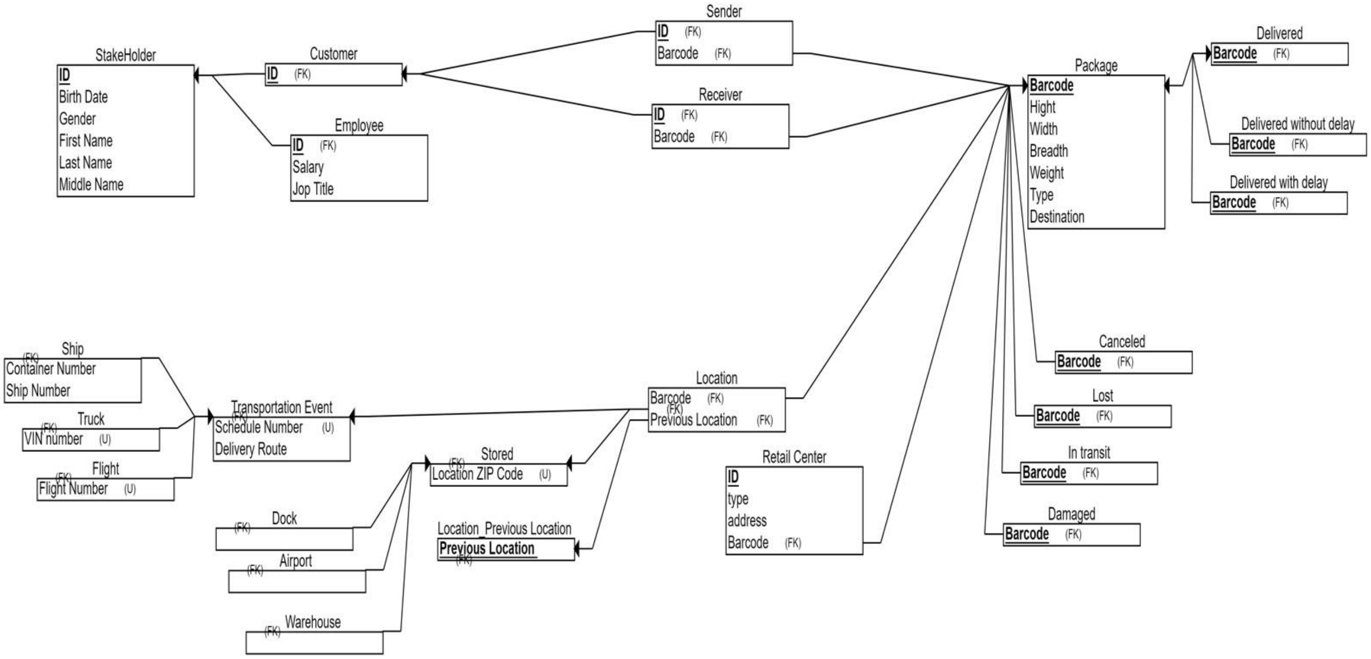
Database Constrains & Business Rules

- A package type specified its physical type, which is either regular, fragile, liquid, chemical, or other.
- The main identifier of any package or location is the barcode number.
- The value and insurance of any package is derived from the weight of package, according to company policy.
- The delay time is the difference between the actual and expected delivery date, which is derived from the source and destination.
- The delay fine is derived from the delay time, according to company policy.

EER Schema



Relational Schema



Work Distribution

Saud Al-Faris	Sultan Al-Moammar	Khalid Al-Baqami
Constrains (50%)	Constraints (25%)	Constraints (25%)
Business rules (50%)	Business rules (25%)	Business rules (25%)
Assumptions brainstorm (33%)	Assumptions brainstorm (33%)	Assumptions brainstorm (33%)
EER (25%)	EER (37.5%)	EER (37.5%)
Relational Schema (25%)	Relational Schema (37.5%)	Relational Schema (37.5%)

Note on Tasks:

Tasks done in this phase were coherently discussed and noted via Microsoft teams. Khalid shared his screen, Sultan and Saud requested control and started the work together. As soon as we finished the EER and the relational schema, we continued to list the assumptions/constraints/business rules. Even though there were many debates on what to include and not to include, we settled for what we have here.

Tools & Resources

Sources used include the course slides and textbook. Erdplus.com was used for drawing and mapping the EER and relational schema.

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DEPARTMENT OF INFORMATION & COMPUTER SCIENCE

ICS321: DATABASE SYSTEMS

Project Phase II

Group Number -6-

Submitted by:

Khalid Al-Baqami – 201934770

Saud Al-Faris – 201946230

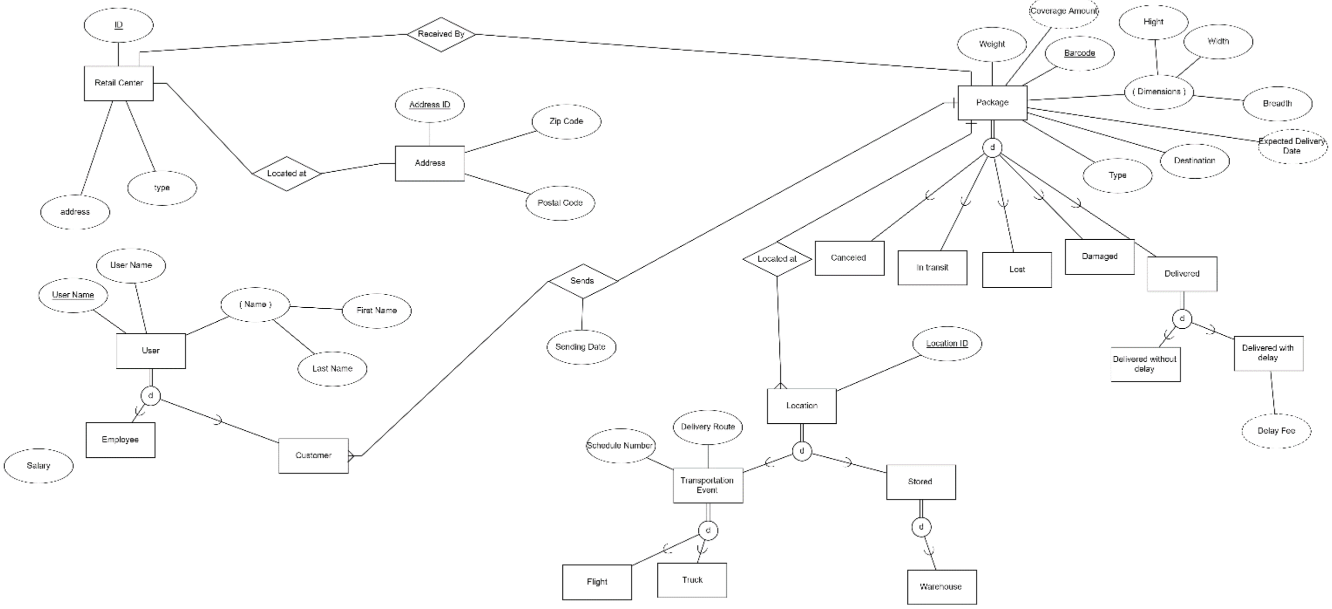
Sultan Al-Moammar - 201928890

Thursday 12/15/2022

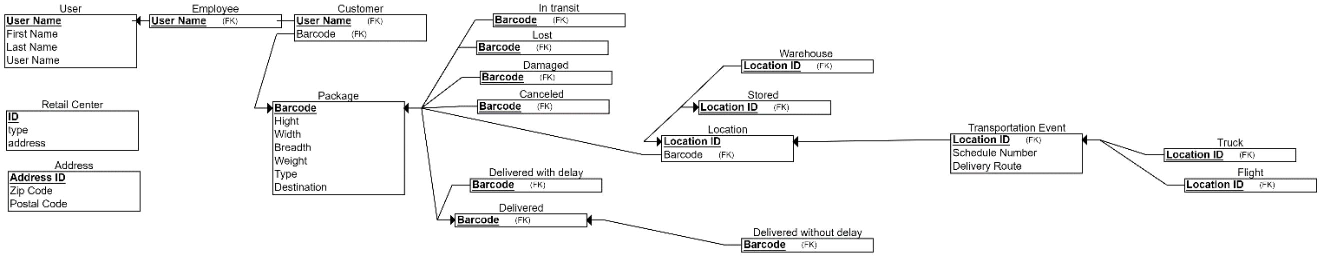
▪ How you implemented Phase 2.

We first modified the previous EER schema and relation schema to what fits the implementation (we realized it was too much to implement) and it became like this.

EER Schema



Relational Schema



We implemented the functional version of the project (phase 2) by developing an MVC-based web application. We implemented the design into a database, linked this database to a back-end web system, and linked this system to a front-end user interface. We used this approach instead of a GUI-based desktop approach to focus on the database implementation instead of the GUI beauty, as this is a course where we present our skills in database systems, not GUI interfaces.

▪ All the tools and resources that you used.

We used HTML and some CSS to develop the user interface views, python as the backend system using the Flask web framework, SQLite as the database, and SQLite3 as the python module to access it. We also used DB Browser to create the database. The selection of these tools and technologies was solely based on the factor of our experience with them.

▪ All the problems you faced.

1) Requirements were not clear, we had to put inefficient effort into understanding them. However, this is part of the course as we cannot expect people to be clear about their needs, we should expect the opposite actually.

2) Another problem was single threading in SQLite3, which could only be solved by combining all our code in a single file, which some inefficiency inefficiencies

3) Finally, we had a problem in integrating Flask with SQLite3, as it could only be integrated with SQLAlchemy. We couldn't use SQLAlchemy because it is an Object Relational Mapper and lacked the ability to enter SQL queries, we feared this could affect our grade so we hardcoded SQLite3 into our Flask application. This was time-consuming but it worked.

▪ All the things that you learned from the project.

Things learned in this project are uncountable, we tried to list the most significant lessons learned.

This project taught us that communication is key, every task of this project required brainstorming and an evaluation. This project also taught us that every project is not ideal, there are firm rules that restrict the completion in such a way that it shapes the process into a certain way. We also learned how to take integrity, referential and domain constrains into consideration throughout the project lifetime. Other than the constraints regarding SQL, there are logical constraints called business rules. Business rules set basis to any assumptions done in completing the project phases. This project made us create an EER Schema referencing the requirements that are listed, then use what we learned throughout the course to transform it into a relational schema. We also learned in phase 2 how to create a functional database mapping the relational schema and develop an interface that is suitable to use what we have worked on. The most important lesson learned is the integration of every part of the project is an important factor to complete the puzzle.

- A table which list the percentage completion of each required operation.

Required operation	
Describe constraints, including business related, and technical constraints, such as primary keys, foreign keys, check constraints, and not null constraints, for the tables and attributes, etc	100%
Design the database, following an EER approach; then go through the normalization process to come up with a collection of tables that are in Third normal forms.	85%
Use any SQL database to create the normalized tables	100%
Find out how the most recent version of your database implements the concept of triggers and then create at least one trigger for your database. For example, one such trigger could be that someone wants to send a package; your trigger must create a package id automatically (optionally).	100%
Populate the database by using SQL insert statements or with some GUI interface such as PhpMySQLAdmin in case you are using MySQL.	90%
Functions of an Administration/Employee	100%
Functions of a Customer	100%
Reports	100%
General Function	100%

- A table of all the extra things done. (For bonus points)

No extra things have been done in phase II.

▪ A table which lists the tasks done by each group member.

Saud Al-Faris	Sultan Al-Moammar	Khalid Al-Baqami
Constrains (50%)	Constraints (25%)	Constraints (25%)
Business rules (50%)	Business rules (25%)	Business rules (25%)
Assumptions brainstorm (33%)	Assumptions brainstorm (33%)	Assumptions brainstorm (33%)
EER (25%)	EER (37.5%)	EER (37.5%)
Relational Schema (25%)	Relational Schema (37.5%)	Relational Schema (37.5%)
Python code and Integration (50%)	Python code and Integration (25%)	Python code and Integration (25%)
Normalization (37.5%)	Normalization (25%)	Normalization (37.5%)
Triggers brainstorming (33.3%)	Triggers brainstorming (33.3%)	Triggers brainstorming (33.3%)
GUI interface (45%)	GUI interface (27.5%)	GUI interface (27.5%)
SQL Table population (33.3%)	SQL Table population (33.3%)	SQL Table population (33.3%)
Report (25%)	Report (37.5%)	Report (37.5%)

Note on Tasks:
Tasks done in this phase were coherently discussed and noted via Microsoft teams. Khalid shared his screen, Sultan and Saud requested control and started the work together. Even though there were many debates on what to include and not to include, we settled for what we have here.

▪ Suggestions to improve ICS 321 future projects.

- The only suggestion we have is that the requirements were vague which causes a huge confusion that has been reflected on the implementation therefore we would suggest a clearer and less requirements which yet still help with delivering the objectives of the course without making as much confusion.