An Online Corporate Operational Management Platform Based on Web Database*

Xingjie He[a], Mengqi Su[a], Mengjie Chen[b], Jinrui Lin[b], Yan Tong[b], Yangzhixin Luo[b], Haoxuan Xie[b], Wenke Zhong[b], Jie Pan[b] and Shuting Jiang[b]

[a]School of Science and Engineering, Chinese University of Hong Kong, Shenzhen
[b]School of Data Science, Chinese University of Hong Kong, Shenzhen

ARTICLE INFO

Keywords:
Web Database
Data Mining
Data Dashboard
Greedy Algorithm
Key Performance Indicator

ABSTRACT

In this project, we intend to build a web-based database that synthesizes various functions of an online operational management platform. The platform, serving a chip-manufacture corporation, allows order release from customers and the order management of staff. The analytical functions of the platform provide assistance for product managers in deciding the optimal production line based on the "shortest processing time first" (SPTF) policy, including visualizing logistics situations and computing the cheapest path using the greedy algorithm.

1. Background and Introduction

Our project aims for developing a web platform based on consumer database for the **Chipanda Semiconductor Manufacturing Company**. The purpose of this project was to create a system that would allow customers to place orders, as well as enable staff to manage those orders. The platform was designed with scalability in mind, allowing it to be easily adapted for future growth and changes in customer demand. In addition, the system has been designed with usability in mind so that customers can quickly and easily place their orders without any confusion or frustration. Finally, the system will provide detailed reporting capabilities for both staff and management which will allow them greater insight into order trends over time.

The target user of our platform is a chip-manufacture corporation that needs a secure, efficient, and easy-to-use system for managing customer orders. Our platform provides several advantages over traditional methods such as

- Increased security measures to protect confidential data during transmission and storage, scalability to accommodate future growth in demand;
- Detailed reporting capabilities that provide greater insight into order trends;
- Take availability and transaction cost into consideration when designing the system, so plant managers can quickly place their orders in the most convenient and economic way.

By providing these features and benefits we believe our platform will offer the best possible experience to both staff and customers alike.

2. Database Structure

To simulate the real situation of a company, we constructed a relational database and generated some data based on real-world situations. This section will elaborate on the relational schema, construction process, and data generation process of our web database.

*This work is a final project of CSC3170: Database System.
ORCID(s):

2.1. Data Collection and Generation

The data are generated using the random package of Python, combining certain real information of sample companies

User The name, country, province, phone number, and street addresses of 1062 users are randomly generated. There are two types of users: customers and staff. Staff is the employee of any plants belonging to his/her province. The province of a user is his/her shipping address, on default.

Plants We downloaded the plant locations of Intel China Inc., and randomly generated the location, phone number, and street addresses for them.

Machine We scratched the machine information from a foreign chip production website, including machines' names, versions, and prices. Each type of machine may have several copies located in different plants, and their corresponding plants are random.

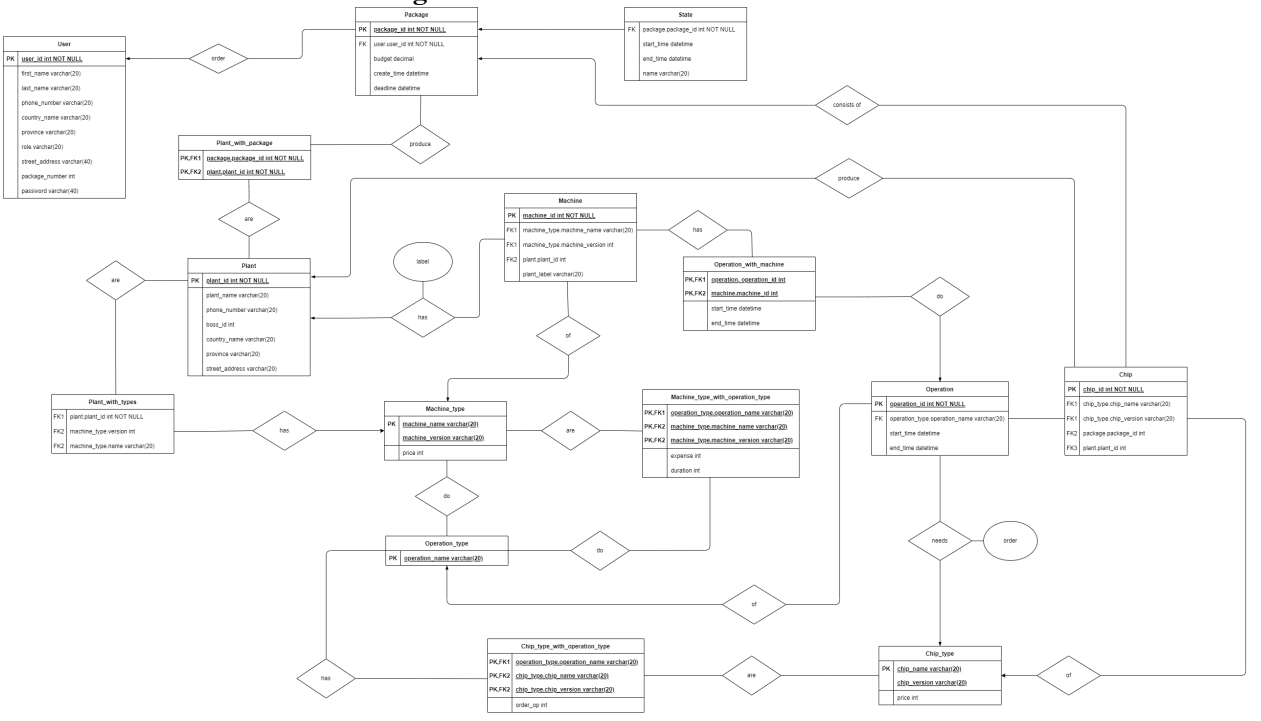
Operation The **operation type** table records the names of each operation necessary for producing different chips. For simplicity, the name of an operation is composed of one letter and one number, for example, "A1" or "D10". A type of operation can be performed multiple times, and the **Operation** table records the starting time and ending time of each operation.

Chip A chip represents an actual integrated circuit to be produced, whose information generally contains its name, version, and price. The price of a chip are mainly below 300 USD, and it is usually increasingly expensive in the more advanced version. This information is included in the table **chip type**. Since there could be many copies of a type of chip, all chips produced by the company are recorded in the table **chip**. A chip can be processed by one plant, but it usually requires the cooperation of different machines.

Other relational tables The relational tables are generated according to some assumptions and constraints, listed as follows:

- One chip can only have one chip type. A chip type can be made by a single or multiple machines. A chip type is made from single or multiple types of operations.
- A machine has only one machine type, and one machine type corresponds to only one type of operation.
- Each operation will be done all at once. And the operations will be done based on the order. For a specific chip type, the type and order of operations are fixed. An operation may be done by multiple machines working together.
- Each user can place multiple orders and users are classified as consumers and staff. The staff can also place order as consumer but the consumers don't have the rights of staff.
- Each package will contain many chips with multiple chip types. Each package is assigned to a factory capable of fulfillment.
- Each plant has single or multiple machine types and each plant has multiple machines.

2.2. Relational Schema and E-R Diagram



User(User_Id, First_Name, Last_name, Phone_number, Country_name, Province, Role, Package_done, Street_address, Password)

Package(User_id, Package_id, Budget, Create_time, Deadline)

State(Package_id, Start_time, End_time, State_name)

Machine_type(Machine_name, Machine_version, Price)

Plant(Plant_id, Plant_name, Phone_number, Boss_id, Country_name, Province, Street_address)

Machine(Machine_name, Machine_version, Price, Plant_id, Machine_id, Plant_label)

Chip_type(Chip_name, Chip_version, Price)

Chip(Chip_name, Chip_version, Price, Package_id, Chip_id, Plant_id)

Operation_type(Operation_name)

Operation(Operation_name, Operation_id, Start_time, End_time)

Operation_with_machine(Operation_id, Machine_id, Start_time, End_time)

The ER-diagram of our system contains 15 entities. The entities are *user, package, sate, plant_with_package, machine, chip, plant, operation_with_machine, machine_type, machine_type_with_operation_type, operation, operation_type, chip_type and chip_type_with_operation_type.*

User Anyone who wants to order chips from our company will have to sign up an account. So the user entity consists of information of the user's account, such as user's name, phone number, region, country, province, address and role.

Package In our system, a package consists of multiple chips, so the relation between package is chips are one to many. A package must be ordered by a customer, foreign key *"user.user_id"* gives the id of the customer. *"Package_id"* identifies the package and determines other attributes. *"budget"* describes the budget for a package. *"Create_time"* is the date when the package was created and *"Deadline"* is the date when the package was supposed to be finished.

State In our system, a package has multiple processing states, so the relation between state and package is many to one. Attribute *"Package.package_id"* is a foreign key connected to Package that identifies the package, *"Start_time"* describes the date when the package was started to be processed, *"End_time"* describes

the date when the processing of the package was finished, *"Name"* describe the processing state of packages, i.e. processing; cancelled; finished.

Plant In our system, a plant consists of multiple machines so the relation between plant and machine is many to one. Attribute *"Plant_id"* identifies the plant, *"Country_name", "Province", "Street_address"* give the location of the plant, *"Phone_number", "Boss_id"* gives the information of the manager, *"Plant_name"* gives the name of the plant.

Plant_with_package Since a plant can produce multiple packages, and a package can be produced by many plants, we designed this relationship table to describe the relationship between plant and package. *"Plant.plant_id"* refers to the plant and *"Package.package_id"* refers to the package.

Machine In our system, a machine belongs to only one plant. *"Machine_type.machine_name"* describes the name of the machine, *"Machine_type.machine_version"* descries the version of the machine, *"Plant.plant_id"* and *"Plant_label"* describes what plant the machine belongs to.

Operation_with_machine This is a relationship table that describes the relationship between operation and machine. *"Operation.operation_id"* and *"Machine.machine_id"* refer to the corresponding operation and machine, *"State_time"* describes when the machine starts to process the operation, and *"End_time"* describes when the machine finishes the operation.

Operation Many operations are needed to produce one chip and each operation belongs to one operation type. In our system, one operation can be processed by more than one machine.*"Operation_id"* is designed to store the id of each operation. *"Operation_type.operation_name"* is used to determine the operation type corresponding to this operation. *"Start_time"* and *"end_time"* are designed to store the start time and end time of each specific operation.

Operation_with_chip A combination of chips and operations. Many operations will be applied when it comes to producing one chip, and one operation can be applied to many chips, the the relation is many to many. *chip.chip_id* gives the id of the chip and *operation.operation_id* gives the id of the operation.

Operation_type In our system, an operation type is determined, if and only if for any machine, the time and expense to process it is determined. Typical operation types are “design-import”, “etch”, “drill” etc. The attribute *“operation_name”* is designed to store the name of each operation.

Machine_type Each machine belongs to one specific machine type. One machine type has corresponding specific operation types. *“Machine_type.machine_name”* and *“machine_type.machine_version”* are designed to store the name and version identifier of each machine and the latter one cannot be empty. *“Price”* is designed to store the price of each machine.

Machine_type_with_operation_type In our system, each type of machine can only process specific operation types. This entity has five attributes. *“Operation_type.operation_name”* is designed to store the name of each operation. *“Machine_type.machine_name”* and *“machine_type.machine_version”* are designed to store the name and version of each machine. “Expense” is designed to store the expense that a certain type of machine needs to process a certain type of operation. “Duration” is designed to store the time a certain type of machine needs to process a certain type of operation.

Chip *“Chip_id”* is designed to store the id of each chip. This entity has four foreign keys of other entities. *“Chip_type.chip_name”* and *“Chip_type.chip_version”* are designed to link the chip with a corresponding chip. *“Package.package_id”* is designed to determine which package the chip is from. *“Plant.plant_id”* is designed to determine which plant the chip is to be produced in.

Chip_type Each chip belongs to one specific chip type. *“Chip_name”* is designed to store the name of each chip and it cannot be empty. *“Chip_version”* is designed to store the production version identifier of each chip and it cannot e empty. *“Price”* is designed to store the sale price of each chip type and the chip price is determined by this attribute.

Chip_type_with_operation_type If the chip type is given, the operation type should be determined. One chip type needs more than one operation. In our system, there aren’t repeated operation types when

processing one chip type. *"Operation_type.operation_name"* is designed to store the name of each operation. *"Chip_type.chip_name"* and *"chip_type.chip_version"* is designed to store the name and production version of each chip type. *"Order_op"* is designed to store the order of the operation in chip manufacturing.

2.3. Cloud Database

We utilize the Huawei cloud database RDS for MySQL to be the back-end database of our project. The version of the database engine is MySQL 5.7.38. The RDS instance is equipped with 2 vCPUs and 4GB of main memory.

After the initialization of the cloud database, we create the schema and tables using *source/db_table.sql*, then insert the data by directly loading the CSV datasets, since the column names in the CSV datasets conform to those in the creating-table SQL respectively. *source/db_insert.sql* is for locally testing whether the insertion of data works as expected. For debugging purposes, we also provide *source/db_clean.sql* to drop all the database tables.

The public IP address of our RDS instance is 123.60.157.95, with the port being 3306. We have granted access right from all IP addresses, thus the front end can retrieve the database structure and data using the following code:

```
cnx = mysql.connector.connect(
    host="123.60.157.95",
    port=3306,
    user="root",
    password="xxx", # replace the password here
    database="project")
```

3. Operational Functions

Our company is named as Chipanda Semiconductor Manufacturing Company and we have developed a web platform to facilitate communication and collaboration with both staff and consumers. The platform consists of three main sections: the login page, the staff portal, and the consumer portal. Here comes a brief introduction of those three parts.

- **LOGIN PAGE** Upon accessing the platform, users are directed to the login page where they can enter their registered credentials. Based on their identity, they are then redirected to the appropriate portal. New users can register for an account in this page as well.
- **CONSUMER PORTAL** The consumer portal allows consumers to order chips and offers information about their current packages.
- **STAFF PORTAL** The staff portal is designed for company employees, on which they can access the information about plants and assign operations to differnt packages.

3.1. Log-in Page

The log-in interface is the gateway to our web platform. It allows users to access the STAFF portal or the CONSUMER portal based on their registered identity. To log in, users simply need to enter their first name, last name, and password. Once logged in, they will be directed to the appropriate portal where they can access the resources and tools they need. New users can register for an account in this page as well. The log-in interface is designed to be user-friendly and easy to use, ensuring that users can quickly and easily access the information they need. It is an essential part of our platform, helping to streamline communication and workflow within staffs, as well as providing a convenient interface for our clients to access information and resources.

3.2. 2C: Consumer Interface

Through the CONSUMER portal, users can access a wide range of features and functions, including the ability to purchase any of the over 20 different types of chips offered by our company. The CONSUMER portal is constantly updated with the latest chips as they are released, making it easy for users to stay up to date with the latest products. In addition to purchasing chips, users can also specify which factory they would like to use for processing, with shipping costs calculated based on the distance between the factory and the user. The CONSUMER portal also provides a convenient way for users to track and supplement their orders.

3.3. 2B: Staff management System

The STAFF portal on Chipanda Semiconductor Manufacturing Company Limited's web platform is a valuable resource for our company's employees. Through the STAFF portal, employees can access a range of tools and features to manage and streamline their work. For example, STAFF can use the portal to assign machines and use the operation function to process packages. The STAFF portal also provides a convenient way for employees to view performance data for individual factories and track the progress of their work. Overall, the STAFF portal is a powerful tool that helps our company's employees stay organized and productive.

- **PLANT MANAGEMENT PAGE** This page contains two main functions. First, the staff can manage the chips. Staff can select any number of chips that are in the state of **"Waiting"**, and the system will assign them machines according to the **Cost-Minimization Algorithm**. After clicking the submit button, the database will be updated during this process, so there should be a page refresh to observe the results after management. Eventually, these selected chips will be in the state of **"Processing"** and therefore will not appear in the selection box of the first function. The second function, on the other hand, displays the uncompleted chips in that plant in a table. Whenever an employee completes managing a chip under the first function, the chip will enter the state of "Processing" and will be displayed in the table.

- **PLANT PERFORMANCE PAGE** This page is used to show the company's performance for a year. It has a total of three data displays. The first display is the **top six chips by REVENUE** in each province. Users can select any number of provinces in a multi-select box, and the six highest-selling chips in the selected provinces will be displayed in a table. The second display is the change in **the company's REVENUE** for each month, displayed in a line chart. Staff can use this table to view the company's monthly revenue status. The third display is the change in the number of **orders accepted** and the number of **orders completed** by the company for each month, displayed in an area chart. Staff can use this chart to check the number of orders received by the company for each month and the status of their completion.

4. Analytical Functions

Our platform provides various analytical functions for chip production companies to maximize their efficiency. Data visualization capabilities offer a comprehensive overview of sales, customer relations, and delivery status throughout the country, helping with identifying areas of improvement. as well as a cost minimization algorithm that can be used to reduce costs and improve profitability. With our intuitive interface and powerful analytics tools, businesses can quickly make informed decisions about their operations in order to increase productivity and reduce overhead expenses.

5. Conclusion

The project intends to build a web-based database that synthesizes various functions of an online operational management platform. with this goal, on the back end, the project members finished building the database in the cloud and created the underlying schemas. In addition, the data was generated. On the front end, the web page allows customers to log in with an account or register an account. After logging in, the webpage provides customers with an introduction page, an order placement interface, and an order management interface.The site also provides Staff with pages to check the company's performance and manage packages. In this project, the team members have accomplished achievements in database generation, database access, and database update.

6. Contribution

Every team member contributed to the project equally.

References