

Sale Dock

Software Design Specification

#	Student ID	Full name
01	QE170148	Lê Minh Vương
02	QE170173	Hồ Trọng Nghĩa
03	QE170097	Đinh Quốc Chương
04	QE170033	Nguyễn Thị Thúy
05	QE170212	Nguyễn Thái Sơn
06	QE170023	Lâm Mỹ Linh

Table of contents

Table of contents.....	1
1. Overview.....	2
1.1. Code packages.....	2
1.2. Database design.....	2
1.2.1. Database schema.....	2
1.2.2. Table description.....	3
1.2.3. Table details description.....	3
1.3. Repository.....	5
1.3.1. Repository.....	5
1.3.2. CI/CD.....	5
1.4. Dependencies.....	6
2. Code designs.....	7
Sequence Diagram.....	7
2.1. UC1: UI for Login Screen with Employees Account.....	7
2.1.1. Class Diagram.....	7
2.1.2. Class Specifications: More precise and clear description of Class Diagram.....	8
2.1.3. Database Queries.....	8
2.2. UC2: UI for Forgot Password Screen.....	8
2.2.1. Class Diagram.....	8
2.2.2. Class Specifications: More precise and clear description of Class Diagram.....	9
2.2.3. Database Queries.....	9
2.3. UC3: Design UI for Dashboard Screen.....	9
2.3.1. Class Diagram.....	9
2.3.2. Class Specifications: More precise and clear description of Class Diagram.....	10
2.3.3. Database Queries.....	10
2.4. UC4: Design UI for Customer Management Screen.....	11
2.4.1. Class Diagram.....	11
2.4.2. Class Specifications: More precise and clear description of Class Diagram.....	11
2.4.3. Database Queries.....	11
2.5. UC5: Design UI for Product Management Screen.....	12
2.5.1. Class Diagram.....	12
2.5.2. Class Specification: More precise and clear description of Class Diagram.....	12
2.5.3. Database Queries.....	12
2.6. UC6: Design UI for Order Management Screen.....	13
2.6.1. Class Diagram.....	13
2.6.2. Class Specifications: More precise and clear description of Class Diagram.....	14
2.6.3. Database Queries.....	14

1. Overview

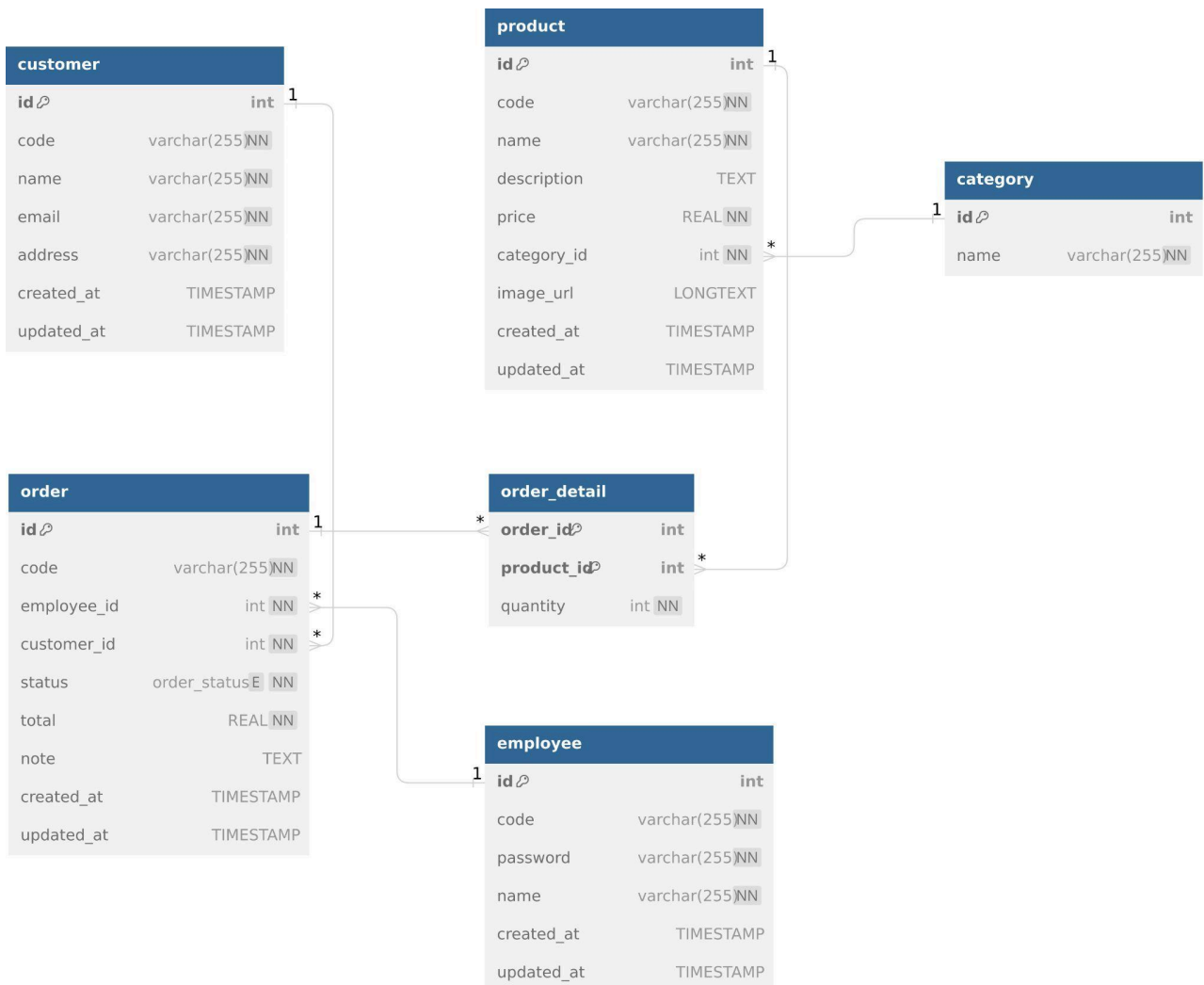
1.1. Code packages

- Java project:
 - Group Id: **io.hardingadonis**
 - Artifact Id: **saledock**
- Package description:

#	Package	Description
01	model	Contain all models in project
02	controller	Contain all Servlets in project
03	utils	Contain all utils in project

1.2. Database design

1.2.1. Database schema



1.2.2. Table description

#	Table	Description
01	customer	Customer information who purchased the product
02	product	Product information
03	category	Product's categories
04	order	Order information
05	order_detail	Details of the products are in the order
06	employee	Employee information

1.2.3. Table details description

- `customer` table

#	Field	Type	Constraint	Description
01	id	int	Primary key, increment	Primary key of table
02	code	varchar	Unique, not null	Customer's code
03	name	varchar	Not null	Customer's name
04	email	varchar	Not null	Customer's email
05	address	varchar	Not null	Customer's address
06	created_at	timestamp	Null	Time to create customer information
07	updated_at	timestamp	Null	Time to update customer information

- `product` table

#	Field	Type	Constraint	Description
01	id	int	Primary key, increment	Primary key of table
02	code	varchar	Unique, not null	Product's code
03	name	varchar	Not null	Product's name

04	description	text	Null	Product's description
05	price	real	Not null	Product's price
06	category_id	int	Foreign key, not null	Product's category
07	image_url	longtext	Null	Product's image
08	created_at	timestamp	Null	Time to create product information
09	updated_at	timestamp	Null	Time to update product information

- `category` table

#	Field	Type	Constraint	Description
01	id	int	Primary key, increment	Primary key of table
02	name	varchar	Not null	Category's name

- `order` table

#	Field	Type	Constraint	Description
01	id	int	Primary key	Primary key of table
02	code	varchar	Unique, not null	Product's code
03	employee_id	int	Foreign key, not null	Id of the employee created the order
04	customer_id	int	Foreign key, not null	Id of the customer placed the order
05	status	order_status	Not null	Current status of the order. Oder_status type has the following value: pending, shipping, done, cancelled
06	total	real	Not null	Total price of the order
07	note	text	Null	Order note
08	created_at	timestamp	Null	Time to create order information
09	updated_at	timestamp	Null	Time to update order information

- `order_detail` table

#	Field	Type	Constraint	Description
01	order_id	int	Primary key, foreign key	Primary key of table
02	roduct_id	int	Primary key, foreign key	Primary key of table
03	quantity	int	Not null	Quantity of the product in order

- `employee` table

#	Field	Type	Constraint	Description
01	id	int	Primary key	Primary key of table
02	code	varchar	Unique, not null	Employee's code
03	password	varchar	Not null	Employee account password
04	name	varchar	Not null	Employee's name
05	email	varchar	Unique, not null	Employee's email
06	created_at	timestamp	Null	Time to create employee information
07	updated_at	timestamp	Null	Time to update employee information

1.3. Repository

1.3.1. Repository

- GitHub repository: [GitHub](#)

1.3.2. CI/CD

- Auto build:
 - Name: Build with Maven
 - Trigger: push & pull_request
 - Branch: all branches
 - Code: [Link](#)
- Auto release:
 - Name: Release Sale Dock
 - Trigger: `Build with Maven` done with success result
 - Branch: main

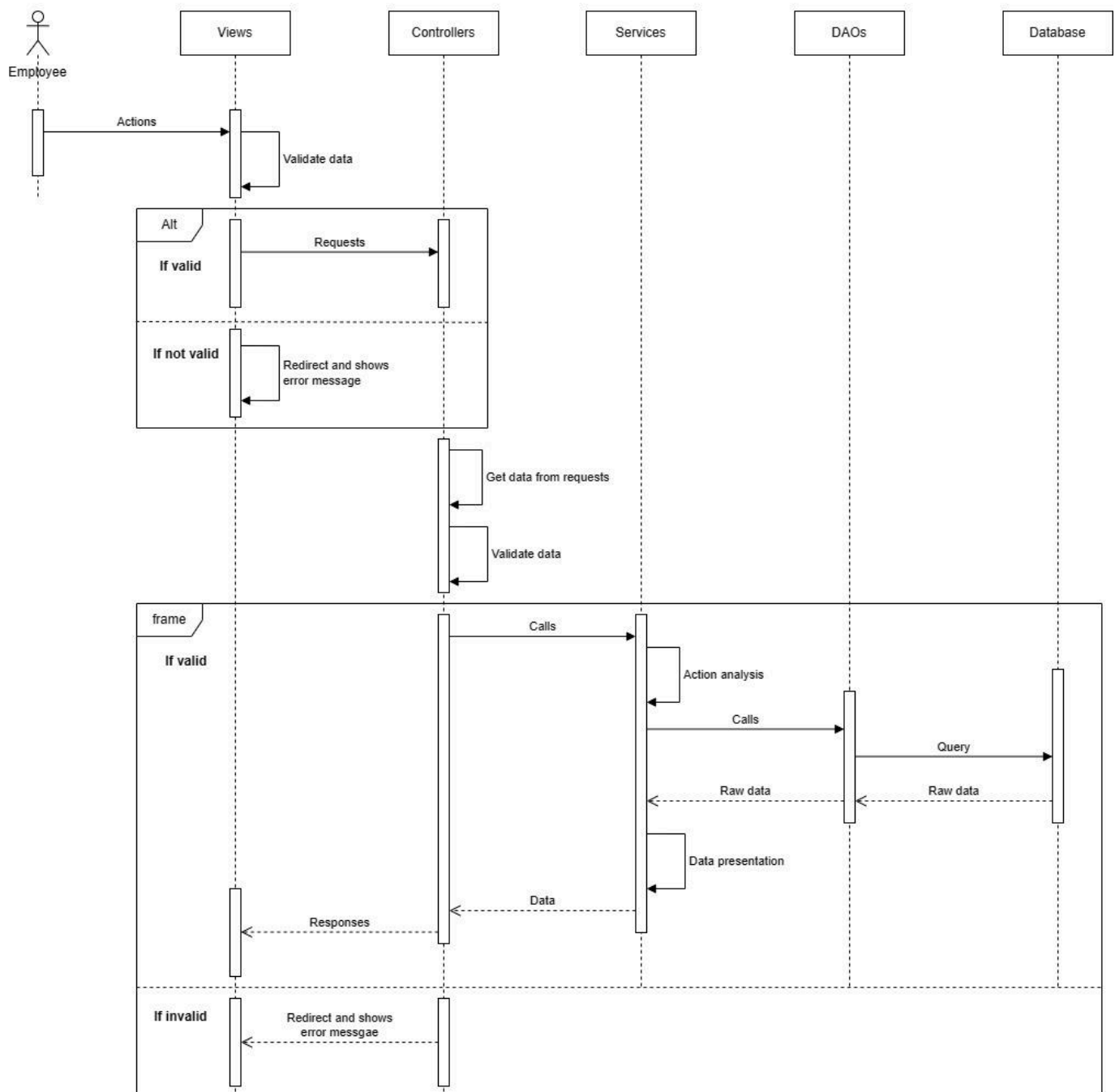
- Code: [Link](#)

1.4. Dependencies

#	Package	Version	Description
01	jakarta.platform.jakarta.jakartaee-api	10.0.0	Java(TM) EE 10 Specification APIs
02	com.mysql.mysql-connector-j	8.2.0	MySQL JDBC Driver
03	org.hibernate.hibernate-core	6.4.1.Finall	JPA implementer
04	org.projectlombok.lombok	1.18.30	Java code made concise with Lombok
05	org.glassfish.jersey.containers.jersey-container-servlet	3.1.5	Jersey Servlet container for GlassFish
06	org.glassfish.jersey.media.jersey-media-json-jackson	3.1.5	Jersey JSON support with Jackson
07	org.glassfish.jersey.inject.jersey-hk2	3.1.5	Jersey integration with HK2
08	commons-codec.commons-codec	1.16.0	Apache Commons Codec
09	com.googlecode.json-simple.json-simple	1.1.1	A simple Java toolkit for JSON
10	com.sun.mail.jakarta.mail	2.0.1	Jakarta Mail API
11	commons-fileupload.commons-fileupload	1.5	Apache Commons FileUpload
12	org.apache.poi.poi	5.0.0	Apache POI Common
13	org.apache.poi.poi-ooxml	5.0.0	Apache POI API Based On OPC and OOXML Schemas

2. Code designs

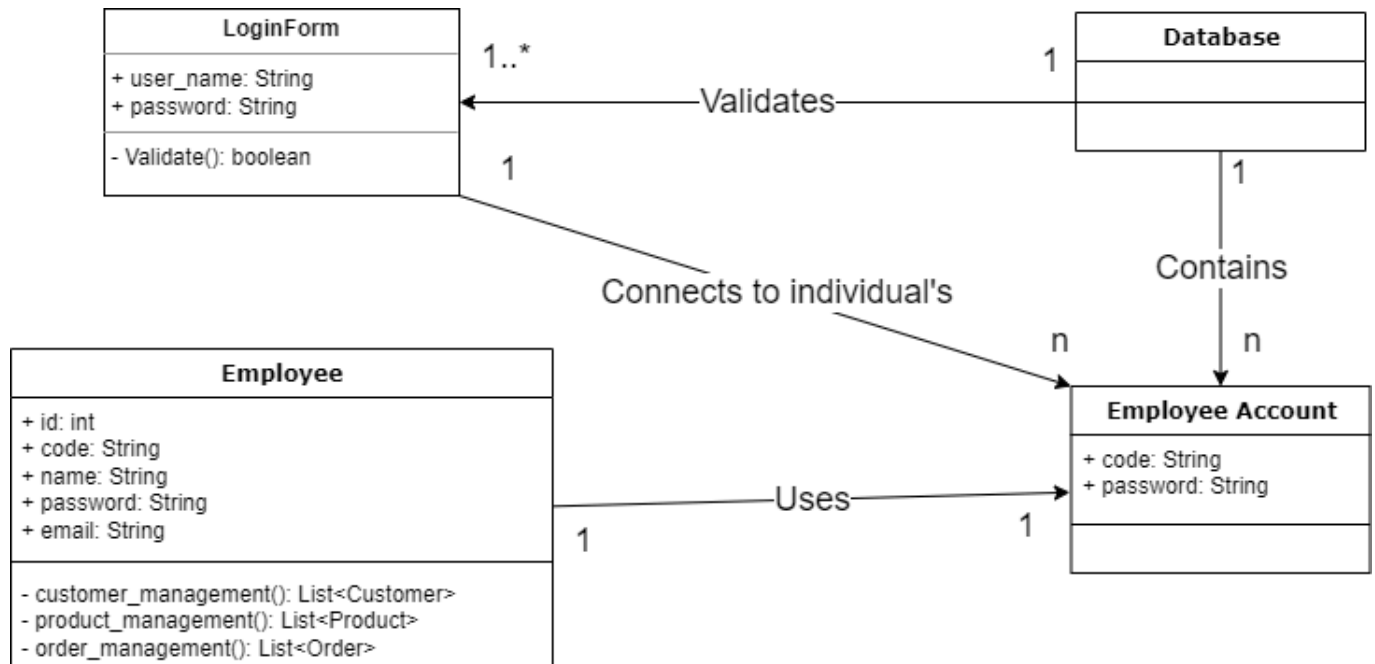
Sequence Diagram



2.1. UC1: UI for Login Screen with Employees Account

2.1.1. Class Diagram

- Below is a class diagram simulating UI for Login with Employees Account:



2.1.2. Class Specifications: More precise and clear description of Class Diagram

- **Employee**: After logging in, employees can perform the management methods below.
 - + Attributes: id, code, name, password, email, create_at, update_at.
 - + Methods : customer_management(), product_management(), order_management(),...
- **Login Form**: This is where users can enter their user_name and password to log in.
 - + Attributes: user_name, password.
 - + Methods : Validate().
- **Employee Account**: Allows employees to use attributes to log into the system.
 - + Attributes: code, password.
- **Database**: Contains information of all users in the system. Used to query data when needed.

2.1.3. Database Queries

- **Query**: This is a database query statement when an employee logs in.

```

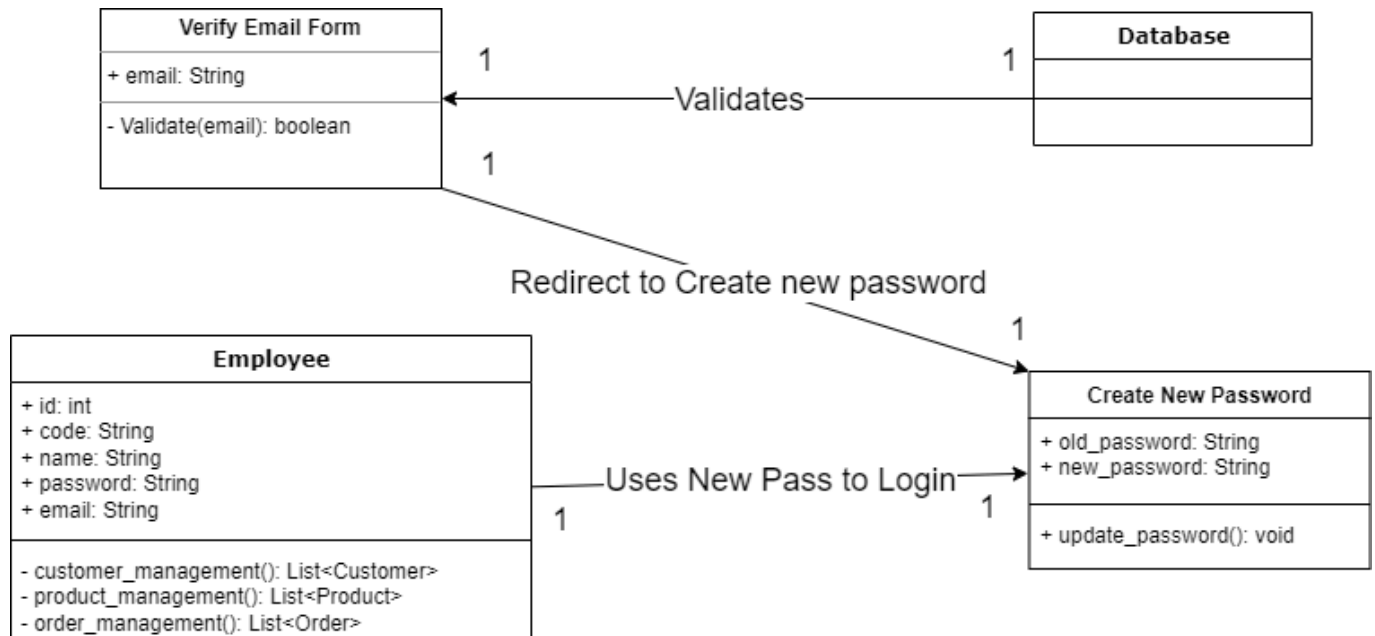
SELECT e.code, e.password
FROM saledock.employee AS e
WHERE e.code LIKE '?'
  
```

- Note: '?' is the employee_code

2.2. UC2: UI for Forgot Password Screen

2.2.1. Class Diagram

- Below is a class diagram simulating UI for Forgot Password Screen:



2.2.2. Class Specifications: More precise and clear description of Class Diagram

- **Employee**: After forgetting the password, employees can receive verified code in their email.
 - + Attributes: id, code, name, password, email, create_at, update_at.
 - + Methods : customer_management(), product_management(), order_management(),...
- **Verify Email Form**: This is where users can enter their email to receive verify code.
 - + Attributes: email.
 - + Methods : Validate(email).
- **Create New Password**: Allows employees to use attributes to log into the system.
 - + Attributes: old_password, new_password.
 - + Methods: update_password().
- **Database**: Contains information of all users in the system. Used to query data when needed.

2.2.3. Database Queries

- **Query**: This is a database query statement to get an account by email.

```

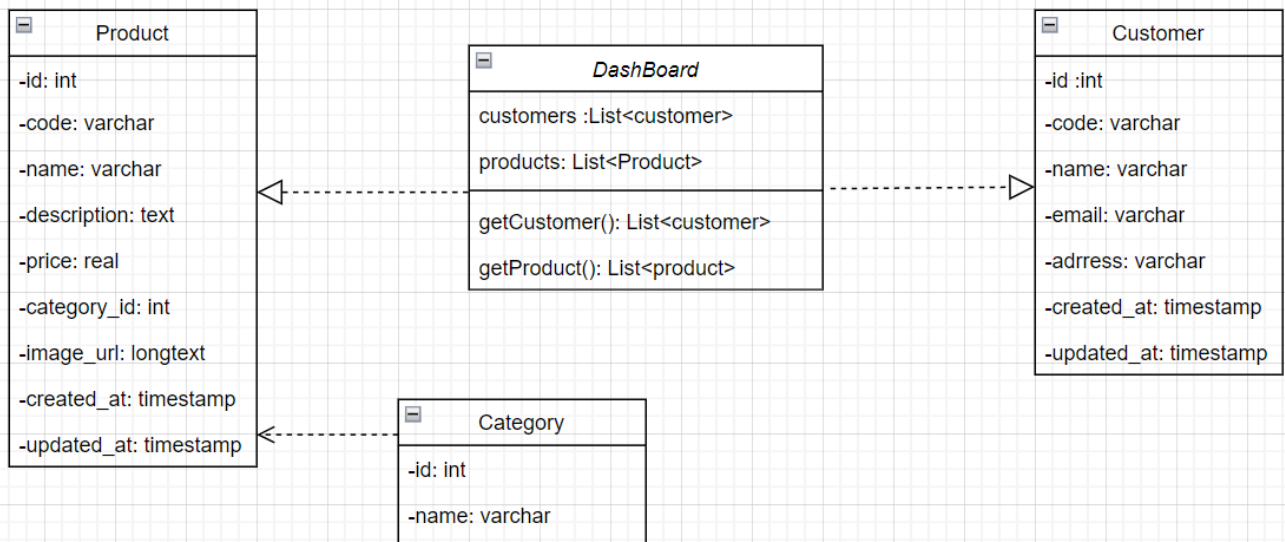
SELECT e.code, e.password
FROM saledock.employee AS e
WHERE e.email LIKE '?';
  
```

- Note: '?' is an email.

2.3. UC3: Design UI for Dashboard Screen

2.3.1. Class Diagram

- Below is a class diagram that simulates the user interface for Dashboard Screen:



2.3.2. Class Specifications: More precise and clear description of Class Diagram

- Dashboard:
 - + Attributes: customers, products.
 - + Methods: getCustomer(), getProduct().
- Product:
 - + Attributes: id, code, name, description, price, category_id, image_url, created_at, updated_at.
- Customer:
 - + Attributes: id, code, name, email, address, created_at, updated_at.
- Category:
 - + Attributes: id, name.

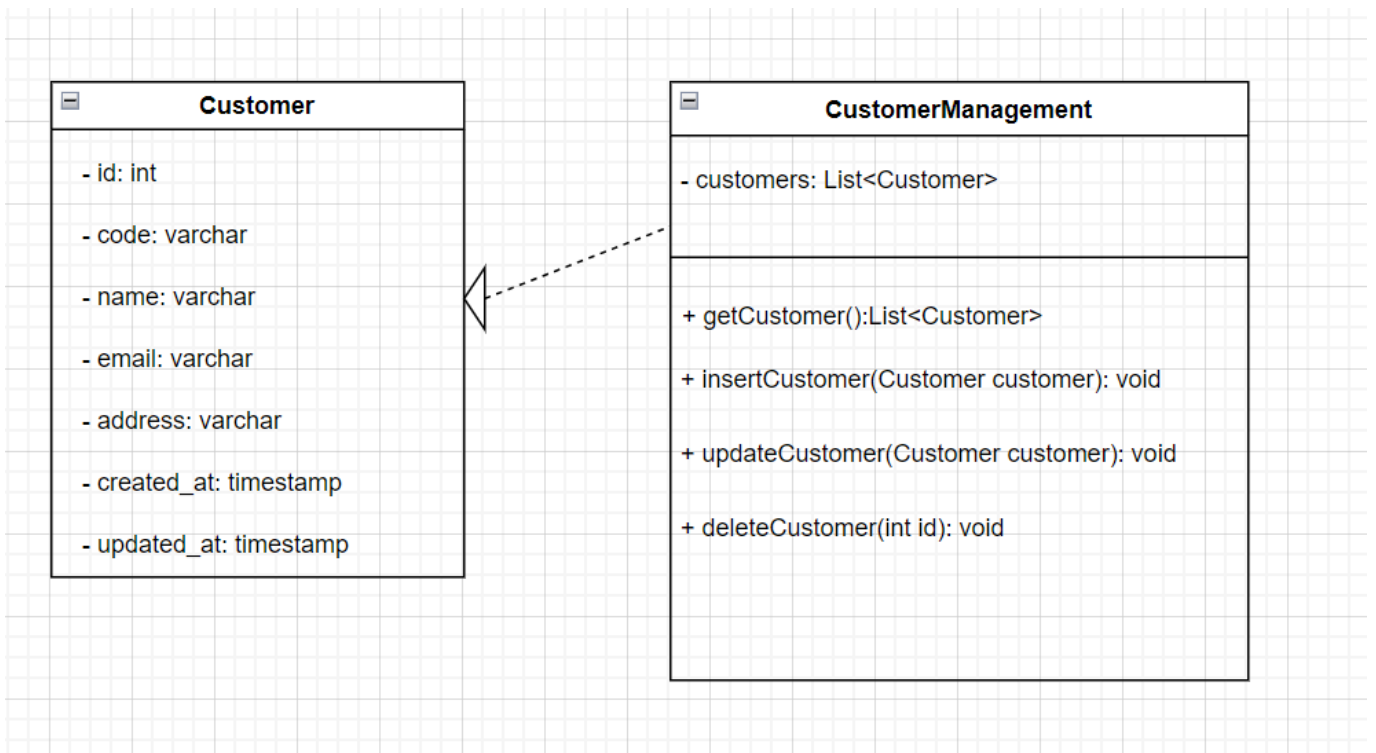
2.3.3. Database Queries

- Query: This is a database query statement show all customers:
 SELECT*
 FROM customer
- Query: This is a database query statement show product information:
 SELECT *
 FROM Product

2.4. UC4: Design UI for Customer Management Screen

2.4.1. Class Diagram

- Below is a class diagram simulating UI for Customer Management Screen:



2.4.2. Class Specifications: More precise and clear description of Class Diagram

- Customer:
 - + Attributes: id, code, name, email, address, created_at, updated_at.
- CustomerManagement:
 - + Attributes: customers.
 - + Methods: getCustomer(), insertCustomer(), updateCustomer(), deleteCustomer().

2.4.3. Database Queries

- Query: This is a database query statement when employees view all customers:
`SELECT*`
`FROM customer`
- Query: This is a database query statement when employees add a customer:
`INSERT INTO customer(id, code, name, email, address, created_at, updated_at)`
`VALUES (?, ?, ?, ?, ?, ?, ?);`
 - + Note: '?' is data read from the customer input value
- Query: This is a database query statement when employees update a customer:

```

UPDATE customer
SET id = ?, code = '?', name = '?', email = '?', address = '?', created_at = '?',
updated_at = '?'
WHERE id = ?
+ Note: '?' is data read from the customer input value

```

- Query: This is a database query statement when employees delete a customer:

```

DELETE FROM customer
WHERE id = ?

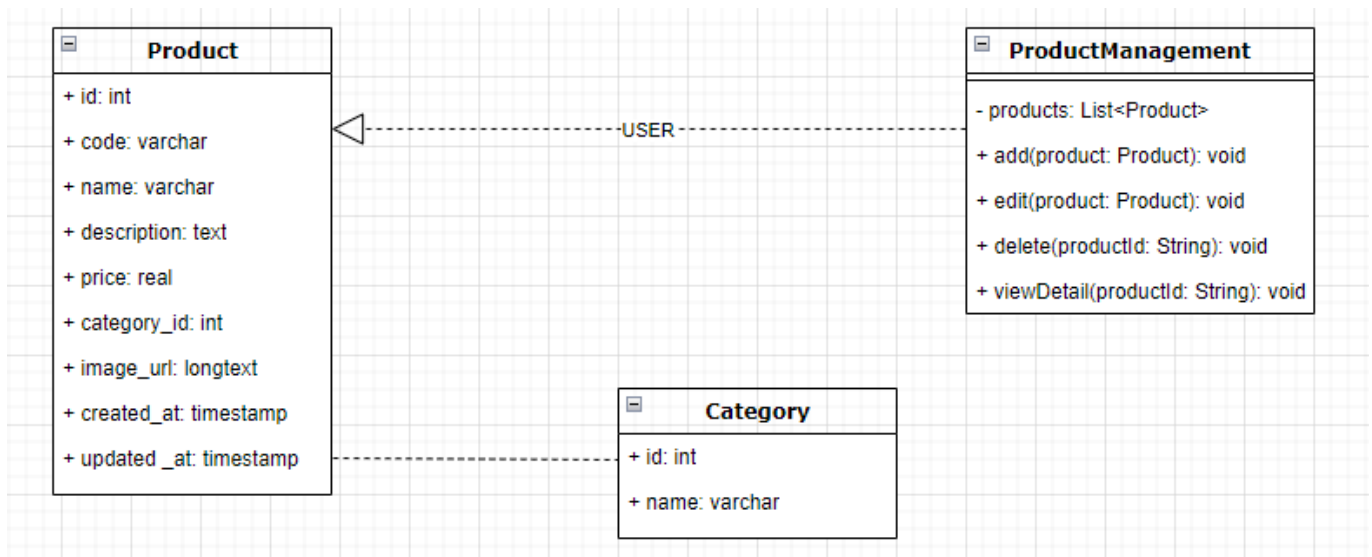
```

- + Note: '?' is data read from the customer input value

2.5. UC5: Design UI for Product Management Screen

2.5.1. Class Diagram

- Below is a class diagram that simulates the user interface for Product Management Screen:



2.5.2. Class Specification: More precise and clear description of Class Diagram

- Product:
 - + Attributes: id, code, name, description, price, category_id, image_url, created_at, updated_at.
- ProductManagement:
 - + Methods: add_product(), edit_product(), delete_product(), viewDetail_product().
- Category:
 - + Attributes: id, name.

2.5.3. Database Queries

- Query: This is a database query statement when employees add new products

```

INSERT INTO product (ID, code, name, description, price, category_id, image_url,
created_at, updated_at)
VALUES (?, '?', '?', '?', ?, ?, ?, '?', '?', '?');

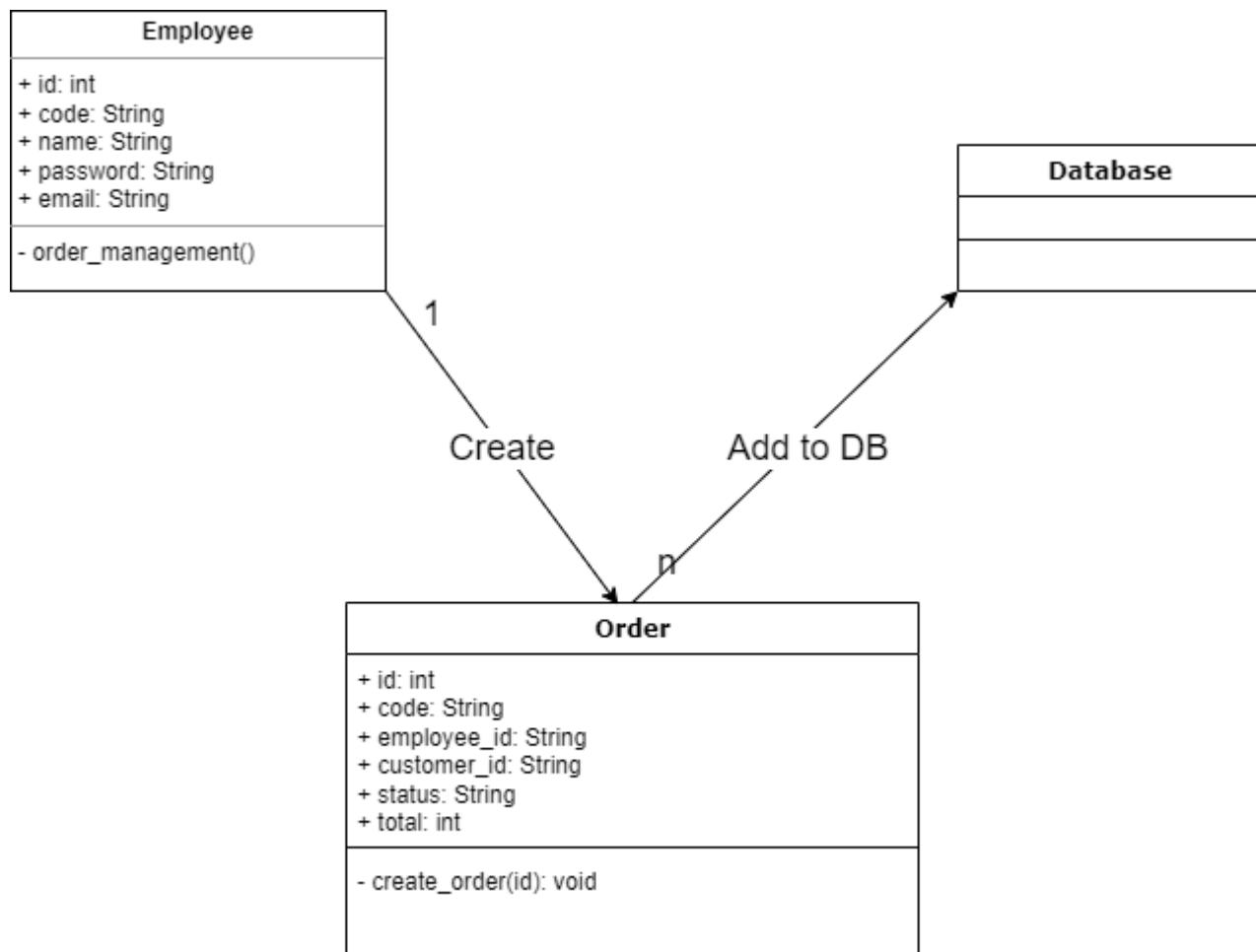
```

- + Note: '?' is data read from the employee input value
- Query: This is a database query statement when employees edit product information
 - UPDATE Product
 - SET ID = ?, code = '?', name = '?', description = '?', price = ?, category_id = ?, image_url = '?',
created_at = '?', updated_at = '?'
 - WHERE ID = ?
 - + Note: '?' is data read from the employee input value
- Query: This is a statement that queries the database when an employee deletes a product
 - DELETE FROM Product
 - FROM ID = ?
 - + Note: '?' is data read from the employee input value
- Query: This is a database query statement when employees want to view detailed product information
 - SELECT *
 - FROM Product
 - WHERE ID = ?
 - + Note: '?' is data read from the employee input value

2.6. UC6: Design UI for Order Management Screen

2.6.1. Class Diagram

- Below is a class diagram that simulates the user interface for Order Management Screen:



2.6.2. Class Specifications: More precise and clear description of Class Diagram

- Employee: After forgetting the password, employees can receive verified code in their email.
 - + Attributes: id, code, name, password, email, create_at, update_at.
 - + Methods : customer_management(), product_management(), order_management(),...
- Order: This is where employees can create orders.
 - + Attributes: id, code, employee_id, customer_id, status, total, note, create_at, update_at.
 - + Methods : create_order().
- Database: Contains information of all users in the system. Used to query data when needed.

2.6.3. Database Queries

- Query: This is a database query statement when employees add new order


```

INSERT INTO product (ID, code, employee_id, customer_id, status, total, note,
created_at, updated_at)
VALUES (?, ?, ?, ?, '?', ?, '?', '?', '?');
      
```

 - + Note: '?' is data read from the employee input value
- Query: This is a database query statement when employees edit order information


```

UPDATE order
SET ID = ?, code = '?', employee_id = ?, customer_id = '?', status = '?', total= ?,
note= '?', created_at = '?', updated_at = '?'
      
```

WHERE code = '?'

+ Note: '?' is data read from the employee input value

- Query: This is a statement that queries the database when an employee deletes a order

DELETE FROM order

FROM code = '?'

+ Note: '?' is data read from the employee input value

- Query: This is a database query statement when employees want to view detailed order information

SELECT *

FROM order

WHERE customer_id = ?

+ Note: '?' is data read from the employee input value