B22DCVT013 - Đỗ Tiến Anh - Costume Rental Management - Design

Nhập môn công nghệ phần mềm (Học viện Công nghệ Bưu chính Viễn thông)

POSTS AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY 1

BIG ASSIGNMENT

INTRODUCTION TO SOFTWARE ENGINEERING

TOPIC: COSTUME RENTAL MANAGEMENT

Class: E22CQCN01 - B

Group: 04

Members:
1. Đỗ Thùy Anh (B22DCCN015)
2. **Đỗ Tiến Anh (B22DCVT013)**
3. Nguyễn Hoàng Phan Anh (B22DCCN028)
4. Trần Anh Dũng (B22DCDT059)

Module: Returns And Pays

Weekly requirements: Design

Hanoi, 2/2025

A. Requirement

I. Concept explanation

1. Raising concepts:

- Manager
- Staff
- Customer
- Costumes
- Personalities
- Accessories

2. Glossary list:

	Vietnamese meaning	English meaning	Explain
People-related terminology			
A.warehouse manager-related terminology			
1	Nhân viên bán hàng	Sales Agent	Person in charge of handling rental procedures for customers
2	Nhân viên tư vấn	Consulting staff	Person in charge of consulting and assisting customers in choosing clothes

	Vietnamese meaning	English meaning	Explain
3	Nhân viên vận chuyển	Delivery Staff	Person in charge of transporting rental clothes
4	Nhân viên kế toán	Account Staff	Person in charge of managing, controlling costs, controlling revenue, accounting for income and reporting
5	Nhân viên vệ sinh	Cleaning staff	Person in charge of laundry, repairing clothes.
6	Nhân viên kỹ thuật	Technical Staff	Person in charge of technical issues, machinery, and store systems
7	Nhân viên quản lý kho	Warehouse Manager	The person in charge of controlling the quantity and quality of clothes in the warehouse
8	Nhân viên quản lý cửa hàng	Store Manager	Manager of store operations, managing other warehouse managers
9	Nhân viên quảng bá	Marketing staff	The person in charge of managing the store's fanpage, updating content on digital media channels, supporting and attracting

	Vietnamese meaning	English meaning	Explain
			online customers.
1	Nhân viên bảo an	Security Staff	Person responsible for looking after and protecting store and customer property
1	Nhân viên quản lý chuỗi cửa hàng	Administrative Manager	Monitor and operate the operations of one or more stores in a chain of stores
B.Customer-related terminology			
1	Khách hàng	Customer	People come to rent costume
1	Trẻ em	Children	Customer under 12
1	Học sinh	Student	Customer between 12 and 18
1	Sinh viên	Undergraduate	College student
1	Người cao tuổi	Elder	Customer over 60

	Vietnamese meaning	English meaning	Explain
1	Người lớn	Adult	Customer between 20 and 40
1	Khách hàng U22	Under 22 customer	Customer under 22
1	Khách hàng thẻ thường	Regular customer	Customer who register normal membership card
2	Khách hàng thẻ VIP	VIP customer	Customer who register VIP membership card
Costume-related terminology			
2	Áo dài	Ao Dai	Traditional Vietnamese costume, designed with a flowing lap
2	Áo sơ mi	Shirt	Shirt has collar and buttons in front, can be long or short sleeved.
2	Áo vest	Vest	A formal overcoat, usually worn with trousers and a shirt on formal occasions.

	Vietnamese meaning	English meaning	Explain
2	Áo khoác	Jacket	A short jacket worn as an outer layer to keep warm or protect against the wind.
2	Áo len	Sweater	The shirt is made of wool, often used in cold weather.
2	Áo măng tô	Overcoat	Long coat, often with lining inside to keep warm.
2	Áo nỉ	Sweatshirt	Thick sweatshirts, often with a layer of fur inside to keep warm.
2	Áo ghi-lê	Waistcoat	Sleeveless shirt, usually worn under a vest.
2	Áo thun	T-shirt	Simple, collarless shirt made of cool cotton.
3	Áo tắm	Bikini	Women's swimwear.
3	Áo choàng có mũ trùm đầu	Cagoule	A hooded cloak, often used to protect against rain or keep warm.

	Vietnamese meaning	English meaning	Explain
3	Áo gió	Windbreaker	Thin, light, windproof and waterproof jacket.
3	Áo cổ lọ	Turtleneck	The shirt has a high collar, covering the neck to help keep warm.
3	Quần âu	European-style trousers	Formal pants, usually worn with a shirt or vest.
3	Quần dài	Trousers	Long pants that cover the legs.
3	Quần ống đứng	Straight-leg Pants	Pants have a straight leg design, not tight.
3	Quần yếm	Overalls	Pants with front bib and shoulder straps.
3	Quần giả váy	Pen Skirt	Pants that look like skirts but are actually wide-leg pants.
3	Váy dài	Dress	Long one-piece dress, available in many different styles.

	Vietnamese meaning	English meaning	Explain
4	Váy dạ hội	Gown	Formal long dress, often worn at major events.
4	Váy đuôi cá	Mermaid dress	The dress hugs the body and flares out at the legs.
4	Váy chữ A	Pencil dress	The dress is slightly fitted at the waist and flares slightly downwards.
4	Váy trễ vai	Off-the-shoulder dress	The dress reveals the shoulders, creating charm.
4	Váy xẻ tà	Slit dress	A skirt with a slit at the leg, creating a stylish accent
4	Váy suông	Shift dress	A loose-fitting skirt that does not hug the body.
4	Quần bơi	Swimming pants	A type of pants worn for swimming
4	Đồ bảo hộ	Protective clothing	Protective clothing for the body when working or participating in hazardous activities

	Vietnamese meaning	English meaning	Explain
4	Đồ lụa	Silk clothes	Clothing made from silk fabric, providing a soft and smooth feel.
4	Chân váy ngắn	Miniskirt	A skirt with a length above the knee, often featuring a youthful style.
5	Chân váy xếp ly	Pleated skirt	Pleated skirt with parallel folds.
Themed costume-related terminology			
5	Quần áo học sinh	Student costumes	Outfits worn by students for specific purposes, such as school events, performances, or themed activities
5	Quần áo cảnh sát	Police costumes	Outfits designed to resemble the uniforms worn by law enforcement officers
5	Quần áo bác sĩ, quần áo y tá	Doctor costumes, nurse costumes	Outfits designed to resemble the professional attire worn by doctors and nurses.

	Vietnamese meaning	English meaning	Explain
5	Quần áo công nhân	Worker costumes	Outfits designed to resemble the clothing worn by workers in various professions
5	Quần áo đầu bếp	Chef costumes	Outfits designed to resemble the professional attire worn by chefs in kitchens and restaurants
5	Quần áo phi hành gia	Astronaut costumes	Outfits designed for astronauts, intended to provide protection and support in the harsh conditions of space.
5	Quần áo quân đội	Military costumes	Outfits designed for military personnel, intended to provide functionality, protection, and identification in various military settings.
5	Quần áo cử nhân	Graduate costumes	Outfits designed for graduates, typically worn during commencement ceremonies to signify academic achievement and the completion of a degree program.

	Vietnamese meaning	English meaning	Explain
5	Quần áo quản gia, người hầu	Janitor costumes, maid costumes	Outfits designed for butlers and servants, intended to provide a professional appearance and functionality while performing household duties and maintaining a formal atmosphere.
6	Quần áo thể thao	Sports costumes	Outfits designed for athletes, intended to provide comfort, flexibility, and support during physical activities and sports.
6	Quần áo dân tộc	Ethnic costumes	Outfits designed to reflect the traditional clothing of a particular ethnic group, intended to showcase cultural heritage and identity.
Accessory-related terminology			
6	Nơ	Ribbon	A long, narrow strip of fabric, often used for decoration, tying, or symbolic purposes.
6	Cà vạt	Necktie	A long, narrow piece of fabric worn around the neck,

	Vietnamese meaning	English meaning	Explain
			usually tied in a knot, as part of formal or business attire.
6	Kẹp tóc	Hairpin	A small, often decorative accessory used to secure hair in place or style it in various ways.
6	Vương miện	Tiara	A decorative, often jeweled headpiece worn as a symbol of royalty, elegance, or special occasions.
6	Thắt lưng	Waist band	A strip of fabric, elastic, or other material that encircles the waist of a garment to provide structure, support, or adjustability.
6	Khăn lụa	Scarf	A piece of fabric worn around the neck, shoulders, head, or waist for warmth, fashion, or cultural significance.
6	Băng đô	Headband	A flexible band worn around the head, typically over the forehead or hair, to hold hair in place, absorb

	Vietnamese meaning	English meaning	Explain
			sweat, or serve as a fashion accessory.
6	Trâm cài	Brooch	A decorative jewelry piece, usually fastened to clothing with a pin or clasp. It is worn for both aesthetic and symbolic purposes.
7	Cài áo	Pin	A small, pointed object used for fastening, decoration, or identification. It can be functional or ornamental, depending on its purpose.
7	Kẹp cà vạt	Tie clip	A small accessory used to secure a necktie to a shirt, keeping it in place and preventing it from shifting.
7	Kính râm	Sunglasses	Eyewear designed to protect the eyes from bright sunlight and harmful UV rays while also serving as a fashion accessory.

	Vietnamese meaning	English meaning	Explain
7	Găng tay	Glove	A piece of clothing worn on the hands to provide protection, warmth, or enhance grip. It covers the fingers and palm, leaving the hands free for movement.
7	Quạt cầm tay	Hand fan	A portable, handheld device used to create airflow and provide cooling. It is typically made of a rigid or foldable structure and is moved manually to generate a breeze.
7	Vòng tay	Bracelet	A piece of jewelry worn around the wrist for decoration, cultural significance, or symbolic meaning.
7	Đồng hồ	Watch	A timepiece worn on the wrist or carried in a pocket to track time. It is both a functional device and a fashion accessory.
7	Túi xách	Handbag	A small to medium-sized bag carried by hand or over the shoulder, used for holding personal items such as money, keys, phone, and

	Vietnamese meaning	English meaning	Explain
			cosmetics.
7	Ô	Umbrella	A device used for protection against rain or sunlight. It consists of a fabric canopy stretched over a collapsible frame with a handle for easy carrying.
7	Mũ cao bồi	Cowboy hat	A wide-brimmed, high-crowned hat traditionally worn by cowboys and ranchers, especially in North America.
8	Mũ nồi	Beret	Soft, round, flat-crowned hat, typically made of wool, felt, or acrylic, and often associated with French, military, and artistic styles.
8	Mũ len	Beanie	A type of close-fitting, soft hat, usually made of knitted fabric, designed to cover the head and keep it warm.
8	Dây chuyền	Necklace	A piece of jewelry worn around the neck, often made from various materials such as metal,

	Vietnamese meaning	English meaning	Explain
			beads, gemstones, or fabric. It is commonly used for fashion, personal expression, or cultural significance.
8	Nhẫn	Ring	Refers to a circular band, usually made of metal, worn on a finger as jewelry or a symbol of status, commitment, or fashion.
8	Bông tai	Earring	A piece of jewelry worn on the ear, usually attached through a piercing in the earlobe or other parts of the ear.
8	Dép quai hậu	Sandals	A type of open-toe footwear that consists of a sole secured to the foot with straps, buckles, or other fastenings.
8	Giày da	Leather shoes	Footwear made primarily from leather, a durable and flexible material sourced from animal hides.
8	Giày cao gót	Heels	Shoes with a raised heel,elevating the back of

	Vietnamese meaning	English meaning	Explain
			the foot higher than the front.
8	Giày hở mũi	Peep toe	A style of footwear where there is a small opening at the front of the shoe, exposing part of the toes.
8	Giày thể thao	Sneakers	A type of comfortable, casual footwear designed for sports, exercise, and everyday wear.
9	Giày búp bê	Ballet flats	A type of lightweight, flat-soled shoes inspired by ballet slippers.
9	Bốt cao gót	Knee high boot	A type of footwear that extends up to or just below the knee.
Costume-related terminology by color			
9	Trắng	White	The lightest color.
9	Đen	Black	The color perceived when no visible light reaches the

	Vietnamese meaning	English meaning	Explain
			eye.
9	Xanh dương	Blue	The color whose hue is that of the clear sky.
9	Be	Beige	A very light brown, sometimes with a yellowish tinge, similar to the colour of undyed wool.
9	Cam	Orange	The colour between yellow and red on the spectrum of visible light.
9	Kem	Cream	Having a pale yellowish color that is close to white.
9	Đỏ	Red	Red is the color that is on the edge of the rainbow.
9	Hồng	Pink	Any of a group of colors bluish red to red in hue, of medium to high lightness, and of low to moderate saturation.
1	Xanh lá	Green	Green is the color between cyan and yellow on the

	Vietnamese meaning	English meaning	Explain
			visible spectrum.
1	Vàng	Yellow	Yellow is the color between green and orange on the spectrum of light.
1	Bạc	Silver	Is a color tone resembling gray that represents the color of polished silver.
1	Nâu	Brown	It is usually made by combining the colors orange and black.
1	Xám	Grey	The colour that is a mixture of black and white
1	Xanh rêu	Moss-green	A muted green hue with tranquil and soothing qualities.
1	Xanh lam	Turquoise	A light greenish blue
Style-related terminology			

	Vietnamese meaning	English meaning	Explain
1	Thanh lịch	Elegant	tastefully fine or luxurious in dress.
1	Sang trọng	Luxurious	very comfortable and expensive.
1	Quyến rũ	Glamorous	attractive in an exciting and special way
1	Cá tính	Unique	Being the only existing one of its type or, more generally, unusual, or special in some way.
1	Cổ điển	Classic	Having a high quality or standard against which other things are judged.
1	Nhẹ nhàng	Lightly	Gently.
1	Trẻ trung	Youthful	Characteristics of young people, or relating to the period of life when you are young.

	Vietnamese meaning	English meaning	Explain
1	Thời thượng	Trendy	Modern and influenced by the most recent fashions or ideas.
1	Tư nhiên	Natural	As found in nature and not involving anything made or done by people.
1	Tối giản	Minimalist	Belonging or relating to a style in art, design, and theatre that uses the smallest range of materials and colours possible, and only very simple shapes or forms.
1	Trưởng thành	Mature	An adjective that describes something or someone that has reached full development, growth, or advanced understanding.
Material-related terminology			
1	Bông	Cotton	Fabric made from the fibers of cotton plants, intended to provide softness, breathability, and comfort, commonly used for casual clothing, bed linens, and

	Vietnamese meaning	English meaning	Explain
			towels.
1	Da	Suede	Material made from the hide of animals, intended to provide durability, flexibility, and a smooth texture, commonly used for clothing, shoes, bags, and upholstery.
1	Lông thú	Leather	A durable and flexible material made from the processed skin of animals, such as cows, sheep, or goats.
1	Lụa	Silk	A natural fiber known for its softness, shine, and luxurious feel. It is produced by silkworms and is widely used in textiles and fashion.
1	Len	Wool	Fabric made from the hair of sheep, intended to provide warmth, softness, and insulation, commonly used for sweaters, coats, and other cold-weather clothing.

	Vietnamese meaning	English meaning	Explain
1	Vải nhung	Velvet	Fabric designed with a soft, velvety texture, intended to provide a luxurious feel and rich appearance, commonly used for elegant clothing, upholstery, and accessories.
1	Sợi tổng hợp	Polyester	Fibers designed from synthetic materials, intended to provide durability, flexibility, and resistance to wear, commonly used in the costumeion of various textiles and clothing.
1	Vải lanh	Linen	Fabric designed for breathable and lightweight clothing, intended to provide comfort in warm weather, commonly used for shirts, dresses, and summer apparel.
1	Vải bò	Denim	Fabric designed for denim clothing, intended to provide durability and comfort, commonly used for jeans and casual wear.

	Vietnamese meaning	English meaning	Explain
1	Vải voan	Chiffon	Fabric designed for lightweight, sheer clothing, intended to provide a delicate and airy feel, commonly used for dresses, blouses, and scarves.
Activity-related terminology			
1	Thuê quần áo	Rent costumes	Customers come to rent the costume.
1	Kiểm tra tình trạng quần áo	Costume-state check	Customers can contact to check the clothes in advance, to avoid running out or not having the clothes they need to rent.
1	Trả quần áo	Return costumes	Customers come to do the return procedure.
1	Thanh toán	Pay	Customers do the payment procedure.
1	Vận chuyển quần áo	Costume transfer	Transfer clothes from the store

	Vietnamese meaning	English meaning	Explain
			to the customer.
1	Hỗ trợ khách hàng	Customer support	Services to answer questions and solve customer problems.
1	Sửa chữa, lắp đặt	Repair, installation	Maintain and improve the condition of the costume.
1	Hủy thuê	Cancel renting	Customers cancel rental orders.
1	Đặt cọc	Deposit	Renters pay a small amount of money to show their commitment to the transaction.
1	Vệ sinh quần áo	Costumes sanitation	Clean costume before and after rental.
1	Kiểm tra quần áo	Costumes check	Check costumes before and after rental.
1	Đăng ký thẻ thành	Membership card	Customers register to become

	Vietnamese meaning	English meaning	Explain
	viên	registration	members.
1	Tích điểm thẻ thành viên	Membership point tick	Points are accumulated, customers can upgrade to VIP members or enjoy incentives.
1	Ưu đãi	Discount	Promotions and incentives when customers meet the requirements.
1	Chạy quảng cáo	Advertisement	Staff use the media to introduce costumes and attract customers.
1	Đánh giá cửa hàng	Shop review	Customers rate the store, share their experiences when using the store's services.
Policy-related terminology			
1	Sau 20h không nhận thuê đồ	No rental service after 8 PM	The store or rental service does not allow customers to rent items after 8 PM.
1	Sau 22h không nhận trả đồ	No returns accepted after 10 PM	The store or rental service does not accept item returns after 10 PM.

	Vietnamese meaning	English meaning	Explain
1	Cọc 50% giá tiền sản phẩm, cọc CCCD khi thuê đồ	Deposit 50% of costume price, deposit personal IID when renting items	A deposit of 50% of the costume price and an ID card deposit are required for rental.
1	Nếu trả đồ trễ, phụ thu thêm phí	If the item is returned late, an additional fee will be charged.	A late return fee will be charged if the item is returned late.
1	Khi trả đồ được hoàn lại cọc đã khấu trừ tiền thuê và phí (nếu có)	When returning the item, the deposit will be refunded minus the rental fee and fees (if any).	Upon returning the item, the deposit will be refunded after deducting the rental fee and any applicable charges.
1	Làm bẩn, làm rách, làm hỏng đồ phụ thu thêm phí	Dirty, torn, damaged items will incur additional charges	An additional fee will be charged for staining, tearing, or damaging the costumes.
1	Tuân thủ quy định thuê đồ của cửa hàng	Comply with the shop's rental regulations	Following the rules and guidelines set by the store when renting items.

II. Business model by natural language

1. Objective & Scope:

- Objective:
  + The purpose of the project is to develop an application for managing costume rentals on desktop computers.
  + Optimize workflow: Helps staff efficiently manage rental orders, costumes, and customer records.
  + Reduce management errors: Ensures accurate tracking of costume status, rental periods, and returns to prevent loss or misplacement.
  + Improve work efficiency: Streamlines processes compared to manual record-keeping.
- Scope:
  + Type of application: desktop-based application (uniform).
  + Serves a chain of costume rental shops.
  + For internal staff use only: warehouse managers can access the system to input data, manage rental orders, and track inventory.
  + No customer-facing interface: Customers place rental requests through warehouse managers directly or other channels.

2. Who uses it? What could it do?

- Administrative manager:
  + Manage information about costume
  + Manage information about customer
  + User statistic: - Costume statistic

    - Customer statistic
- Sales Agent:
  + Help customer to search information about costume (name, theme, style, price, description, discount)
  + Rent/ cancel costume (directly or indirectly via phone) (with the requirements from the user)

- Account staff:
+ Recheck and checkout (check costume's status, collect additional fee if there are any damage)
+ Manage promotional information (% discount)
- Warehouse manager:
+ Import costume
+ Manage costume information(add, search for clothing information)
+ Manage supplier .(add, search by supplier name)
- Store Manager:
+ Statistical report (Statistics by date, sorted in descending order of the total borrowed column, followed by the descending of the total proceeds column).
- Suppliers:
+ Import costume
+ Manage supplier (add)
- Customer:
+ Search for information about costume

3. How the function work?

- Module "Import costume":
- A warehouse manager selects the function to import costumes from a supplier

  → The interface to search supplier by name appears

- The warehouse manager enters the name of the supplier and searches it

  → Shows a list of suppliers containing the new name

- If the supplier is in the system -> The system displays the supplier name, the item the supplier has entered

  → Clicks the correct supplier (If the supplier is not in the system -> The system displays a message: "The supplier does not exist"

  + Add new supplier ->Enter the supplier name and click on the add new supplier button ->The system notifies that the supplier has been added and updates the supplier name in the list and continue)

- Repeat until all costumes need to be imported from that supplier: select search costumes by name and add costume (The system will check if the outfit already exists in the database or not
  + If the costume does not exist -> The system will display a completely new costume interface -> warehouse manager must enter costume information including: name , color, style, material, price, button of search, add -> After entering the necessary information, warehouse manager clicks on create new costume -> The system adds this new costume.
  + If the costume already exists -> The system will display an interface to add existing costumes -> warehouse manager must enter information including: quantity, unit price and warehouse date -> After entering the necessary information, warehouse manager clicks on add costume -> The system will update the costume quantity and warehouse date.)

  → confirm the invoice entered with the supplier and pay to the supplier, receive costumes.

- The system saves and prints the invoice to ask the supplier to sign and save.

● Module "Costume renting":

- Customer calls to the shop to rent a/ many costume(s)
- A sales agent receives the call

+ Ask which one and how long the customer want to rent for:

- Customer answers the style of costume and the rent time
- Sales agent login into the system
- Sales agent search in the system for available costume in the shop
- If no result found:

+ The sales agent tells the customer no more available costumes

→ The renting process ends

- If there are still some available costumes:

+ The sales agent tells the customer all kind of available style of costumes and tell the customer to choose one/ many of them
- If the customer does not choose any:

  → The process ends

- If the customer chooses one/ many costume(s):

  → The sales agent selects the related costume(s) in the interface and clicks rents

  → The system → customer information

  → The sales agent asks customer for personal information, name, tel

  → Customer give the information to the sales agent

- If there is no information

  → Add new customer info

- If already exists

  → Select the related info

- Sales agent repeats all the related information to the customer:
+ Info about costume(s): name, theme, style, price, discount
+ Info about period: rent/ return time
+ Info about customer: name, tel number, address
- Sales agent asks the customer to confirm the reservation:
+ If the customer confirms:

  → Sales agent confirm

+ If the customer cancels:

  → Return to the main interface of the sales agent

- If the customer can take the costume(s) directly:

→ The renting process ends successfully

- If the customer can not take the costume(s) directly:

  → The delivery staff ship the rented costume(s) to the customer’s address

  → The renting process ends successfully

- Module "Customer returns and pays"

- Chooses the menu to search the list of borrowed costumes by the

  customer's name

  → The system displays a list of customers whose names contains the entered keyword and a list of the costumes that the customer is borrowing, each costume on a line with full information about the costume, loan date, loan price per day, number of days

  borrowed, and rental amount up to the date of payment, the last column is the check box to select pay

- Clicks on the pay button for the costumes that the customer returns (with may not pay in full), enter the status of the costume and the fine if any, finally clicks the payment button

  → The system displays the invoice with full customer information (id card number, name, address, tel, email, note), a list of the costumes to be returned, is the total amount paid, the amount deposited, the amount the customer has to pay or return to the customer

- Clicks confirm

  → System updates to the database

- Module "Statistics of costumes":

- A Store Manager selects the menu of statistics of costumes

+ Enter the time period (start - end date) statistics

→ The system displays the list of costumes borrowed in the form of a table costume with complete information:

+ Code

+ Name

+ Model

+ Genre

+ Column total number of rentals

+ Column total amount collected

(Sorted in descending order of the total borrowed column, followed by the descending of the total proceeds column).

- The Store manager clicks on a line of a costume

→ The system displays the details of the invoice with the borrowed costume:

+ Id

+ Name of the borrower

+ Borrowed date and time

+ Payment date and time

+ Total amount.

4. Objects related to?

- Shop: name, address, star rating (from 0 to 5 stars), description (including image/video/text)

- Customer: id card number, name, address, tel, email, note
- Supplier: name, address, tel, email, note
- Employee account: full name, username, password, role, note
- Costume: name, color, style, material, price, discount.
- Bill: paymentDate, paymentMethod
+ This contains the information about the creator (sale agent): name, role.
+ The information about the customer: id card number, address, tel.
+ The information about rented costume(s): name, theme, style, price, rent/return time, total amount.
+ The last row of the bill represents the total amount of the bill, the paid amount and the remaining amount to pay.
- Additional fee: costume name, costume theme, costume style, total rent fee, additional fee follows by:
+ Tattered costume: 50% of the original costume price.
+ Damaged costume: 100% of the original costume price.
+ Stained costume: 20% of the costume rental fees.
- Statistic on costume(s) by revenue: costume name, costume theme, costume style, total rented days, total income.
- Statistic on customer by revenue: customer id card, customer name, customer address, total rent days, total payment.
- Statistic on revenue by time: rented time (day, month, year), total number, total income.

5. Relationship among objects

- A customer can rent one or more costumes (One to Many).
- A costume can be rented by many customers (at different times) (One to Many).
- An order can consist of many costumes (One to Many)
- An order can be paid for in multiple installments (corresponding to many invoices). (One to Many)
- A shop has many costumes. A costume belongs to a shop. (One to Many)
- A supplier can provide many costumes, a costume can be provided by many suppliers. (Many to many)
- A supplier can have many bills, a bill contains a supplier. (One to Many)
- A warehouse manager may create many bills. (One to Many)

- A costume could be rented by many customers in different periods.
- A customer could rent many costumes in different periods. A customer could also rent many costumes at the same time (representative for a group of customers). (One to many)
- A bill can contain many costumes, a costume can be contained in many bills (in different periods). (Many to many)
- A store manager can view many bills. A bill can be viewed by many store manager (many to many)
- The customer in a rented receipt could use shipping service.
- A service could be used by many customers in many rented receipts.
- A costume can be rented if and only if the costume is still available during the desired rent period.
- For each rental, the customer could pay several times until the return time.
- The customer could cancel their rented costume (free).
- A sales agent could rent many times for customers.
- A sales agent could process payment for many customers.

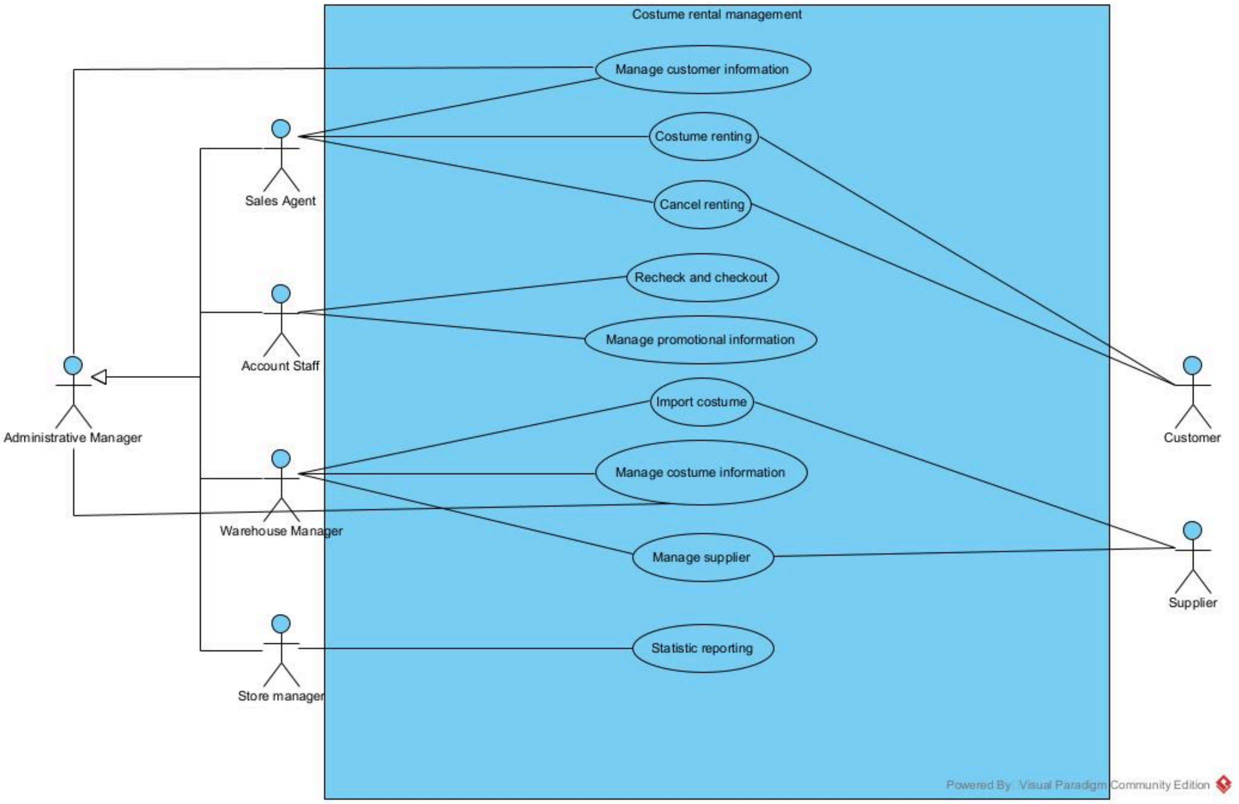
III. By UML

1. Create actors:

- Direct actors: Administrative manager, Sales agent, Account staff, Warehouse manager, Delivery staff. They are the same type as warehouse manager of the shop (employee - abstract actor)
- Indirect actor: customer, supplier
- Actor related functions:
    + Administrative Manager: Manage customer information, Manage costume information.
    + Sales Agent: Manage customer information, Costume renting, Cancel renting.
    + Account Staff: Recheck and Checkout, Manage promotional information.
    + Warehouse Manager: Manage costume information, Import costume, Manage supplier.
    + Store Manager: Statistical report.

- Customer: Costume renting, Cancel renting.
- Suppliers: Import costume, Manage supplier.

2. General use case diagram



3. Describe UC

- Manage customer information: this use case enables the sales agent, administrative manager to manage customer information.
- Costume renting: this use case enables the sales agent to rent a new costume.
- Cancel renting: this use case enables the sales agent to cancel a renting for a customer.
- Recheck and Checkout: this use case enables the account staff to process return and payment costumes.
- Manage promotional information: this use case enables the Account staff to design promotional programs.
- Manage costume information: this use case enables the sales agent, administrative manager to manage information costume.

- Statistical report: this use case enables the store manager to allow for statistics on costume.
- Manage supplier: this use case enables the warehouse manager to manage suppliers.
- Import costume: this use case enables the warehouse manager to import costumes to the warehouse.

4. Detail UC for module “Returns and Pays”

- Use Case Diagram

B. Analysis:

I. Scenario (Version 1)

Scenario	Returns and Pays
Actor	Seller, Customer
Pre-condition	The seller is logged into the system. The customer has rented costumes and their information is available in the system.
Post-condition	The costume return details are updated in the system. A payment Bill is generated.

Main event	1. The Staff enters the Login interface, inputs Username and Password, and presses the Login button. 2. The system displays an interface including the following options: Manage Costumes, Customer Information, Rental Process, Returns, Payments, and Reports. 3. The staff selects the "Returns" option. 4. The system displays an interface including the following options: Search Rental Form, Inspect Costume, Apply Fees, Process Return, and Payment. 5. The staff selects the "Search Rental Form" function in the system. 6. The system displays a search bar. 7. The staff enters the customer’s fullName or idCard in the search box and clicks the "Search" button. 8. The system displays a list of rental forms matching the search criteria, including customer details (fullName, tel, email). 9. The staff selects the correct rental form based on the displayed information. The system retrieves and displays the details of the rental form, including: ● Rental date ● Costume details (name, category, size, color, type, style, material) ● Condition at rental time The staff inspects the returned costume and selects the "Inspect Costume" option. The system prompts the staff to input the condition of the returned costume via the Rental Inspection Form (inspect date, condition). The staff enters the costume condition: ● If damaged, the staff fills out the Damage Fee Form (damaged costume, level of damage, damage fee).

	● If not damaged, the process continues to the next step. The system continues to the next step. The staff checks for late returns and selects the "Apply Fees" option. The system displays the Late Return Penalty Form (return due date, real return date, late fee rate, late fee amount) and calculates any applicable late fees. The staff confirms the fees and proceeds to the "Process Return" option. The system updates the Return Form (date, costume) and marks the costume as returned. The staff clicks the "Payment" button. The system generates a Bill containing: ● Borrowed date and time (from Rental Form) ● Return date and time (from Return Form) ● Total amount (including rental fee, late fee from Late Return Penalty Form, and damage fee from Damage Fee Form) ● Customer name (fullName) ● Costume details (name, category) The staff confirms and clicks the "Confirm Payment" button. The system updates the costume status in the inventory (via Warehouse Entry Form) and records the transaction in the Rental Payment Form (level of damage, rental fee). The transaction is completed, and the staff can print the Bill for the customer if needed.

II. Entity Classes Extraction

1. Describe the module in one paragraph:

+ When a customer returns rented costumes, the staff selects the "Return" function from the menu, enters the customer's name into the search box, and clicks the search button. The system displays a list of customers whose names contain the entered keyword, and the staff selects the correct customer based on personal information such as full name, address, phone number, email, and ID card number. After selecting the customer, the system retrieves the list of costumes currently rented by the customer, displaying detailed information including the rental date, daily rental price, number of rental days, and the total rental amount calculated up to the current date, with each costume accompanied by a checkbox for the staff to select the items being returned. After making the selections, the staff clicks the "Return" button and enters the actual condition of each costume, recording any additional charges if necessary, such as compensation fees for damaged costumes or late return penalties. The system automatically generates related forms such as the return form recording the costumes being returned and their conditions, the warehouse entry form confirming the returned costumes' conditions in storage, the rental inspection form documenting the inspection results, the deposit form recording the initial deposit amount, the late return penalty form for overdue returns, and the damage fee form for damaged costumes. Simultaneously, the system consolidates all charges and generates a final bill, which includes detailed customer information, a list of returned costumes, total rental cost, deposit amount, any penalty or damage fees, and the final amount the customer must pay or be refunded. When the staff clicks the "Confirm" button, the system saves all information into the database, linking the forms and the bill to the customer and the staff who processed the transaction, thereby completing the costume return process.

2. Nouns extraction and classify:

+ Customer: fullName, address, tel, email, idCard, note
+ Costume: name, style, color, material, price
+ Rental Costume: borrow date, daily price, rental days, total, note
+ Return Costume: return date, condition, damage fee, late fee, note, total
+ Final Bill: payment type, payment date, total amount, deposit, late fee, damage fee, amount due/refund, note
+ User (staff): userName, fullName, role
+ SearchButton, PayButton, ConfirmButton, Checkbox
+ System: interface, database, search & confirmation features
+ Database: stores entities like Customer, Costume, Final Bill, Return Info, User, etc.

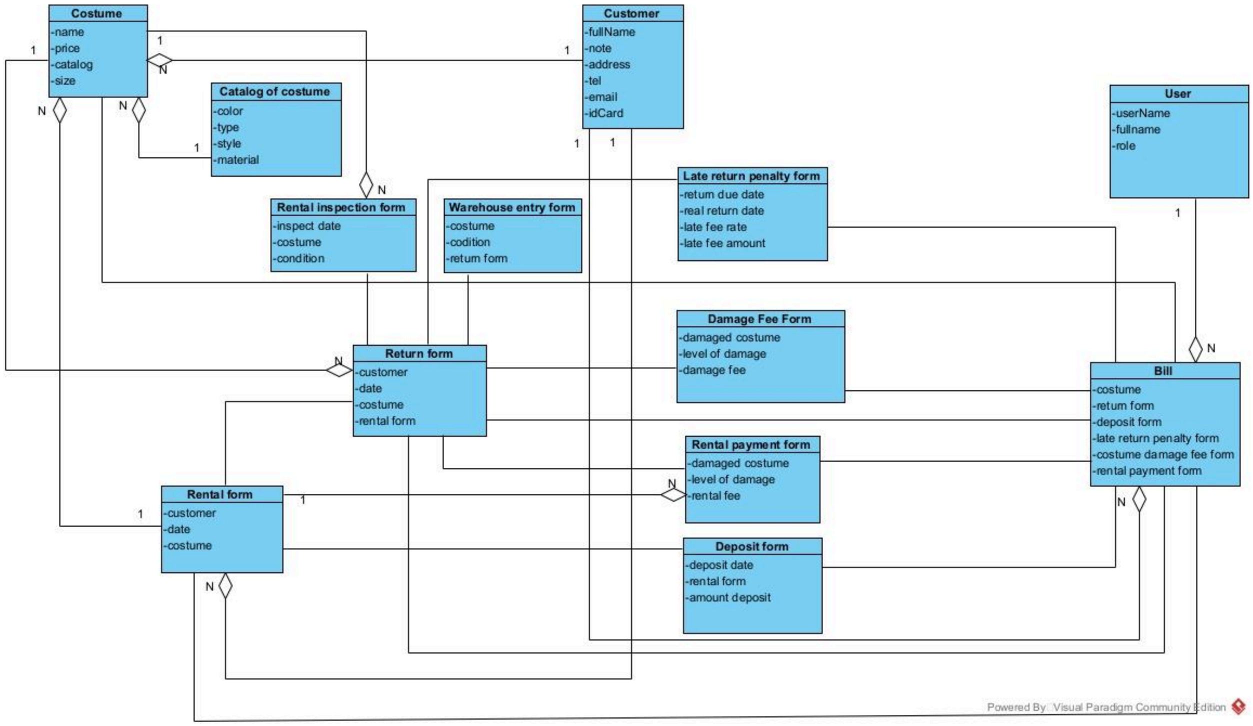
- Forms (not entities, but generated during process): Return Form, Warehouse Entry Form, Rental Inspection Form, Deposit Form, Late Return Penalty Form, Damage Fee Form

3. Quantitative relationship among classes

- Customer – Bill
- A customer can have many bills, but each bill belongs to only one customer
  → Customer (1) — (N) Bill
- Customer – Rental form / Return form
- A customer can have multiple rental and return forms
  → Customer (1) — (N) Rental form
  → Customer (1) — (N) Return form
- Rental form – Costume
- One rental form can include multiple costumes, and a costume can appear in many rental forms
  → Rental form (1) — (N) Costume
  → Costume (1) — (N) Rental form
- Return form – Costume
- One return form can include multiple costumes, and a costume can appear in many return forms
  → Return form (1) — (N) Costume
  → Costume (1) — (N) Return form
- Costume – Catalog of costume
- A costume belongs to one catalog, and a catalog can include many costumes
  → Catalog of costume (1) — (N) Costume
- Costume – Rental inspection form / Warehouse entry form
- A costume can have multiple inspection and warehouse entry records
  → Costume (1) — (N) Rental inspection form
  → Costume (1) — (N) Warehouse entry form
- Return form – Warehouse entry form
- A return form can be associated with multiple warehouse entries
  → Return form (1) — (N) Warehouse entry form
- Return form – Rental form

- Each return form is linked to one rental form
  → Rental form (1) — (1) Return form
- Rental form – Deposit form
- A rental form may have one deposit form
  → Rental form (1) — (0..1) Deposit form
- Costume – Rental payment form / Damage Fee Form / Late return penalty form
- A costume can be related to multiple fee forms
  → Costume (1) — (N) Rental payment form
  → Costume (1) — (N) Damage Fee Form
  → Costume (1) — (N) Late return penalty form
- Bill – Return form / Deposit form / Late return penalty form / Damage Fee Form / Rental payment form
- A bill can include one or more of the related forms
  → Bill (1) — (1) Return form
  → Bill (1) — (1) Deposit form
  → Bill (1) — (1) Late return penalty form
  → Bill (1) — (1) Damage Fee Form
  → Bill (1) — (N) Rental payment form

4. Class Diagram



III. View Class

Enter the system → The login interface appears → need a class: LoginView

Input for username → inUsername

Input for password → inPassword

A submit button to login → subLogin

Enter the username/password → The system must check if the login is correct → need a method:

Name: checkLogin()

Input: username, password (of the class User)

Output: boolean

Assign to the entity class: User

Once login is successful → The main interface of the seller appears → need a class: HomeView, which has at least:

An option to search for rental forms → subSearchRentalForms
An option to view return forms → subViewReturnForms
An option to view deposit forms → subViewDepositForms
An option to view late return penalty forms → subViewLatePenaltyForms
An option to view damage fee forms → subViewDamageFeeForms
An option to view rental payment forms → subViewRentalPayments
An option to view warehouse entries → subViewWarehouseEntries
An option to view bills → subViewBills

When the seller chooses the search option for rental forms → The interface to search for a customer appears → need a class: SearchCustomerView

Input field to enter the customer’s name → inCustomerName
Search button → subSearchCustomer
A list of results → outCustomerList

Enter the customer's name to search → The system must search for all customers whose name contains the entered keyword → need a method:

Name: searchCustomer()
Input: keyword
Output: List<Customer>
Assign this method to the entity class: Customer

The results are returned to (and displayed on) SearchCustomerView.

The seller chooses the correct customer → The interface displaying the list of rental forms appears → need a class: RentalFormView

Input field for customer ID → inCustomerID
Search button → subSearchRentalForms
A table displaying the list of rental forms, including:
Rental Form ID
Rental Date
List of borrowed costumes
Rental Price per Costume
Number of Costumes
Total Rental Amount

The last column is a checkbox to select costumes to return → outRentalCostumeList

Enter the customer ID → The system must search for all rental forms that the customer has → need a method:
Name: searchRentalForms()
Input: customerID
Output: List<RentalForm>
Assign this method to the entity class: RentalForm

Before confirming the rental → The deposit form interface appears → need a class: DepositFormView
Input deposit date → inDepositDate
Input deposit amount → inDepositAmount
Submit button → subConfirmDeposit

The system must save the deposit → need a method:
Name: createDeposit()
Input: rentalForm, amount
Output: boolean
Assign to: DepositForm

The seller selects the costumes to return → The interface for entering return form appears → need a class: ReturnFormView
Input field for condition of costumes upon return → inCostumeCondition
A save button to submit return information → subSaveReturnForm

Enter the return information → The system must save the return form → need a method:
Name: saveReturnForm()
Input: ReturnForm
Output: boolean
Assign this method to the entity class: ReturnForm

Then the system checks for late return → The late return penalty form interface appears → need a class: LatePenaltyFormView
Input expected return date → inReturnDueDate

Input actual return date → inRealReturnDate
System calculates late fee rate and amount → outLateFeeRate, outLateFeeAmount
Confirm button → subConfirmLatePenalty

Need a method to calculate the penalty →
Name: calculateLatePenalty()
Input: dueDate, realDate
Output: double
Assign to: LateReturnPenaltyForm

Then the system checks for damaged costumes → The costume damage fee form interface appears → need a class: DamageFeeFormView
Input level of damage → inDamageLevel
Output damage fee → outDamageFeeAmount
Confirm button → subConfirmDamageFee

Need a method to calculate damage fee →
Name: calculateDamageFee()
Input: costume, damageLevel
Output: double
Assign to: DamageFeeForm

Then the system proceeds to calculate the rental payment → The rental payment form interface appears → need a class: RentalPaymentFormView
Display damaged costumes and corresponding rental fee → outRentalFees
Confirm button → subConfirmRentalPayment

Need a method to calculate rental fee:
Name: processRentalPayment()
Input: rentalForm, List<Costume>
Output: boolean
Assign to: RentalPaymentForm

Then the system summarizes all amounts → The payment receipt interface appears → need a class: PaymentReceiptView
Display customer information → outCustomerInfo

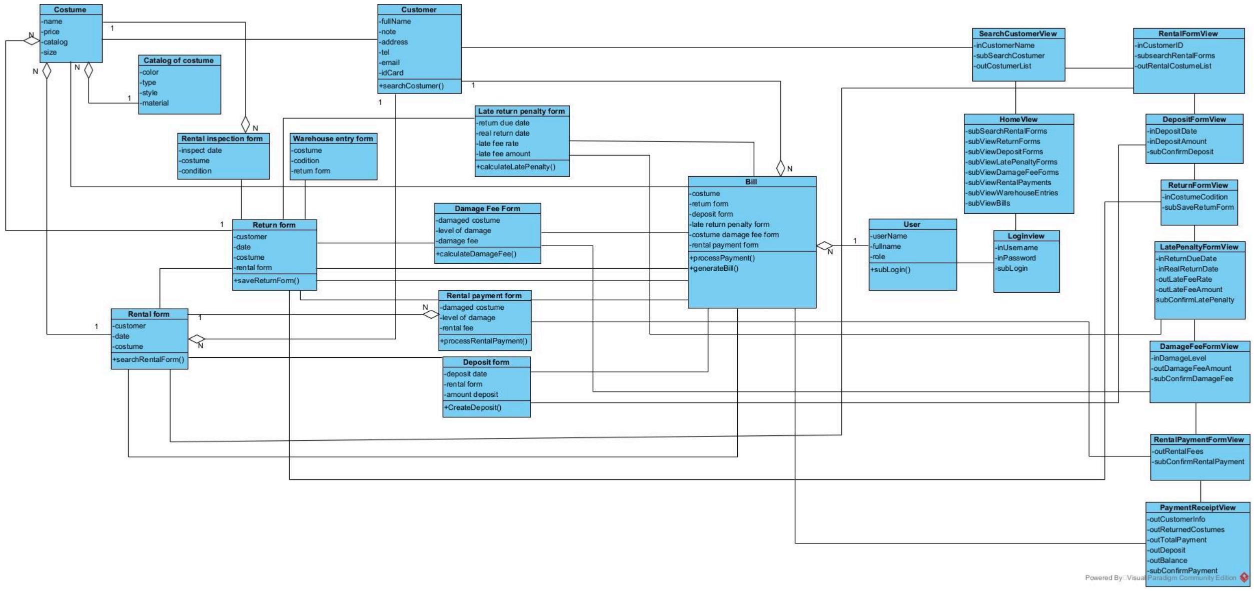
Display list of returned costumes → outReturnedCostumes
Display total payment → outTotalPayment
Display deposit → outDeposit
Display balance (pay/refund) → outBalance
A confirm payment button → subConfirmPayment

System must process the payment →
Name: processPayment()
Input: List<ReturnedCostume>, condition, fine
Output: boolean or void
Assign to the entity class: Bill

Then system must generate bill →
Name: generateBill()
Input: List<ReturnedCostume>, amountPaid
Output: Bill
Assign to the entity class: Bill

After bill is generated → The warehouse entry form appears → need a class: WarehouseEntryFormView
Input warehouse condition → inWarehouseCondition
A save button to save warehouse entry → subSaveWarehouseEntry

Save the warehouse entry → The system returns to HomeView.



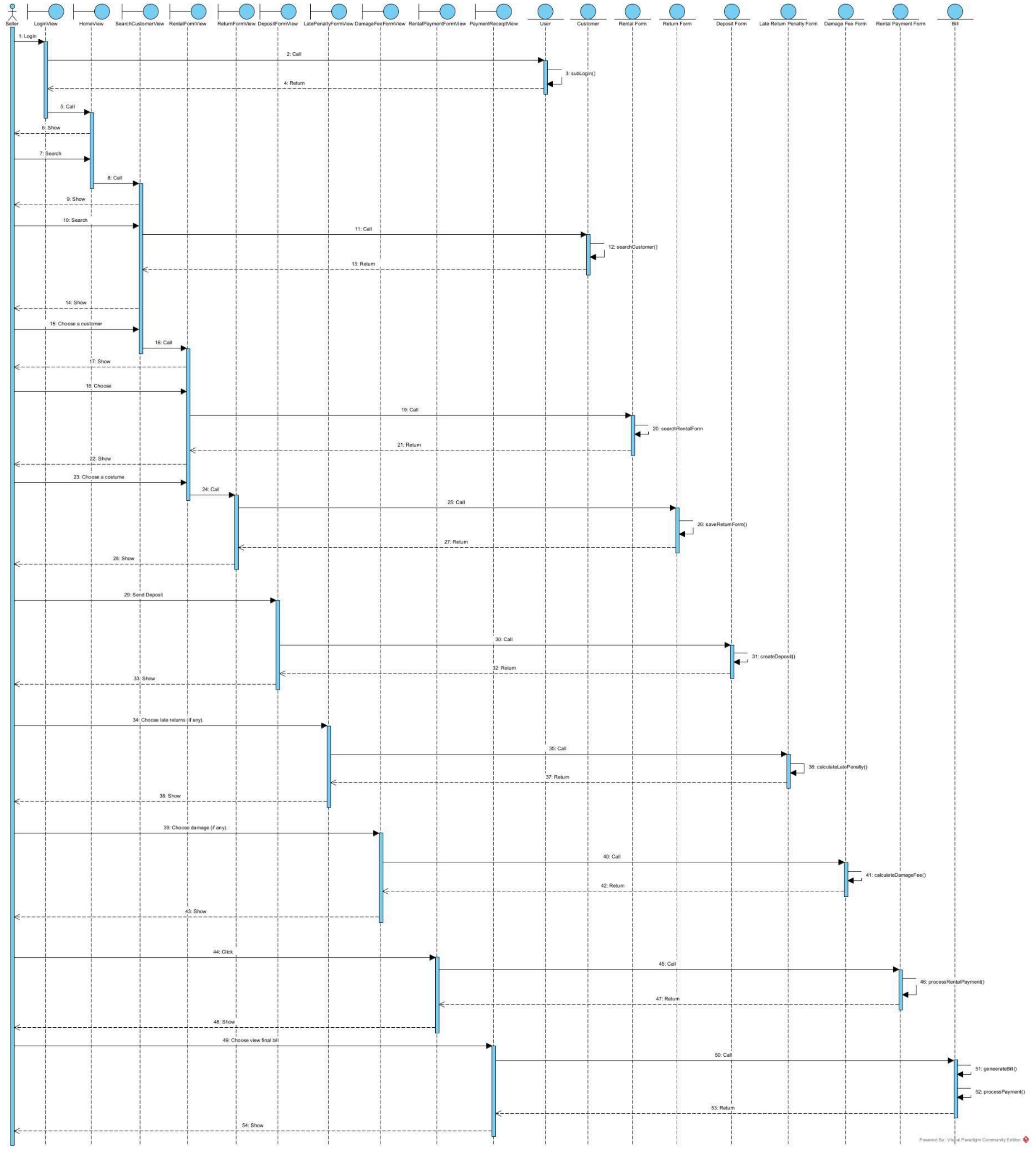
IV. Sequence Diagram

1. Scenario Ver 2

1. The seller enters username and password, then clicks the Login button.
2. The class LoginView calls the class User to process.
3. The class User calls the method checkLogin(). The login is successful.
4. The class User returns the result to the class LoginView.
5. The class LoginView calls the class HomeView.
6. The class HomeView displays itself to the seller.
7. The seller searches for a customer.
8. The class SellerHomeView calls the class SearchCustomerView.
9. The class SearchCustomerView displays itself to the seller.
10. The seller enters a keyword and clicks the search button.
11. The class SearchCustomerView calls the class Customer to process.
12. The class Customer calls the method searchCustomer().
13. The class Customer returns the result to the class SearchCustomerView.
14. The class HomeView displays the list to the seller.
15. The seller chooses a customer.
16. The class SearchCustomerView calls the class RentalFormView.
17. The class RentalFormView calls the class RentalForm to process.
18. The class RentalForm calls the method searchRentalForm().

19. The class RentalForm returns the rental forms to the class RentalFormView.
20. The class RentalFormView displays the rental forms to the seller.
21. The seller selects a costume to return.
22. The class RentalFormView calls the class ReturnFormView.
23. The class ReturnFormView calls the class ReturnForm to process.
24. The class ReturnForm calls the method saveReturnForm().
25. The class ReturnForm returns the confirmation to the class ReturnFormView.
26. The class ReturnFormView displays the confirmation to the seller.
27. The seller chooses to send a deposit.
28. The class DepositFormView calls the class DepositForm to process.
29. The class DepositForm calls the method createDeposit().
30. The class DepositForm returns the result to the class DepositFormView.
31. The class DepositFormView displays itself to the seller.
32. The seller chooses the option to process a late return penalty.
33. The class LatePenaltyFormView calls the class LateReturnPenaltyForm to process.
34. The class LateReturnPenaltyForm calls the method calculateLatePenalty().
35. The class LateReturnPenaltyForm returns the result to the class LatePenaltyFormView.
36. The class LatePenaltyFormView returns the result to the seller.
37. The seller chooses the option to process a damage fee.
38. The class DamageFeeFormView calls the class DamageFeeForm to process.
39. The class DamageFeeForm calls the method calculateDamageFee().
40. The class DamageFeeForm returns the result to the class DamageFeeFormView.
41. The class DamageFeeFormView returns the result to the seller.
42. The seller chooses the option to process rental payment.
43. The class RentalPaymentFormView displays itself to the seller.
44. The class RentalPaymentFormView calls the class RentalPaymentForm to process.
45. The class RentalPaymentForm calls the method processRentalPayment().
46. The class RentalPaymentForm returns the result to the class RentalPaymentFormView.
47. The class RentalPaymentFormView returns the result to the seller
48. The seller chooses to view the final bill.

49. The class PaymentReceiptView calls the class Bill to process.
50. The class Bill calls the methods generateBill() and processPayment().
51. The class Bill returns the final bill to the class PaymentReceiptView.
52. The class PaymentReceiptView displays the final confirmation to the seller.



C. Design

I. Entity classes design

1. Add the id attribute for the classes which DO NOT inherit from other classes

- Costume, Customer, User, CatalogOfCostume, RentalForm, ReturnForm, RentalInspectionForm, WarehouseEntryForm, LateReturnPenaltyForm, DamageFeeForm, RentalPaymentForm, DepositForm, Bill.

2. Convert all association relationships to correspond aggregation/composition relationships

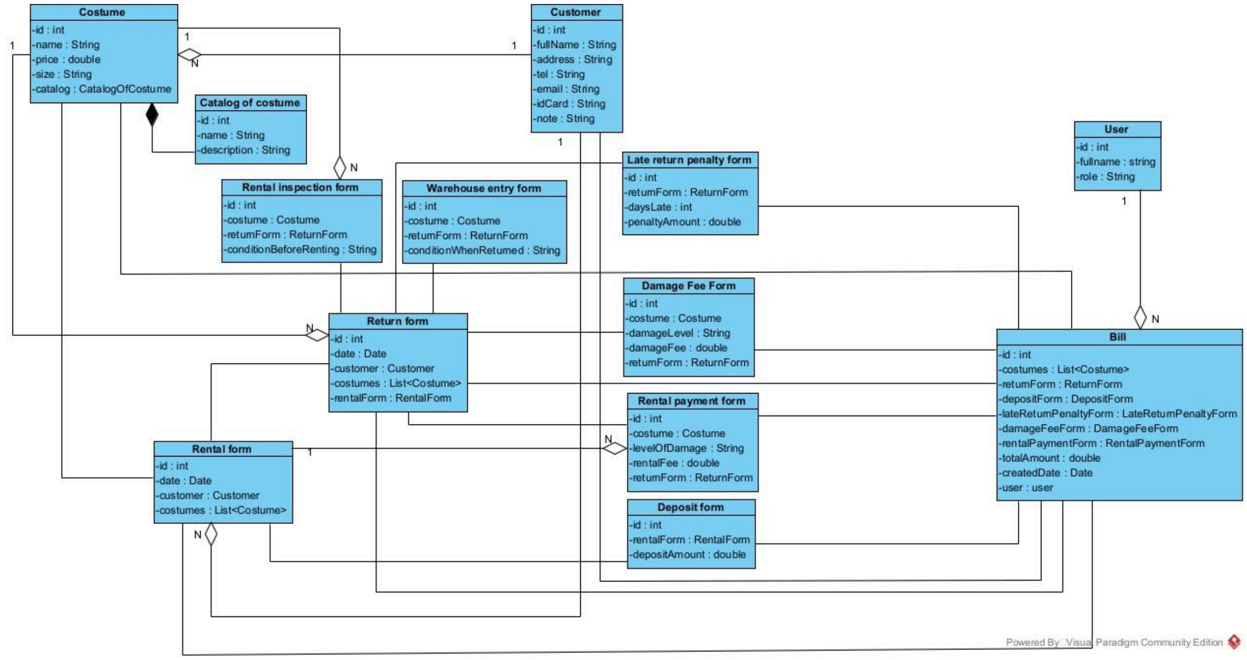
- Costume + CatalogOfCostume -> Costume is a component of CatalogOfCostume.
- Costume + RentalForm -> Costume is a component of RentalForm.
- Costume + ReturnForm -> Costume is a component of ReturnForm.
- Costume + Bill -> Costume is a component of Bill.

3. Add the object attributes which correspond to the aggregation/composition relationships

- Costume is a component of CatalogOfCostume, of type n-1 -> CatalogOfCostume has a list of Costumes.
- Costume is a component of RentalForm, of type n-1 -> RentalForm has a list of Costumes.
- Costume is a component of ReturnForm, of type n-1 -> ReturnForm has a list of Costumes.
- Costume is a component of RentalInspectionForm and WarehouseEntryForm, of type 1-1 -> RentalInspectionForm and WarehouseEntryForm have a Costume.
- Customer is a component of RentalForm, of type 1-1 -> RentalForm has a Customer.

- Customer is a component of ReturnForm, of type 1-1 -> ReturnForm has a Customer.
- ReturnForm is a component of Bill, of type 1-1 -> Bill has a ReturnForm.
- User is a component of Bill, of type 1-1 -> Bill has a User. DepositForm, LateReturnPenaltyForm, DamageFeeForm, RentalPaymentForm are components of Bill, of type 1-1 -> Bill has a DepositForm, a LateReturnPenaltyForm, a DamageFeeForm, and a RentalPaymentForm.

4. Entity Class Diagram



II. Database design

1. Create a corresponding table for each entity class

- Entity class **Customer** → table **tblCustomer**
- Entity class **Costume** → table **tblCostume**
- Entity class **Catalog** → table **tblCatalog**
- Entity class **RentalForm** → table **tblRentalForm**
- Entity class **ReturnForm** → table **tblReturnForm**

- Entity class **DepositForm** → table **tblDepositForm**
- Entity class **RentalPaymentForm** → table **tblRentalPaymentForm**
- Entity class **LateReturnPenalty** → table **tblLateReturnPenalty**
- Entity class **DamageFeeForm** → table **tblDamageFeeForm**
- Entity class **WarehouseEntry** → table **tblWarehouseEntry**
- Entity class **RentalInspection** → table **tblRentalInspection**
- Entity class **Bill** → table **tblBill**
- Entity class **User** → table **tblUser**

2. Transfer all NON-OBJECT attributes to contribute as the columns of the corresponding table

- **tblCustomer**: id, fullName, address, tel, email, idCard, note
- **tblCostume**: id, name, price, size, tblCatalogID
- **tblCatalog**: id, name, description
- **tblRentalForm**: id, date, tblCustomerID
- **tblReturnForm**: id, date, tblCustomerID, tblRentalFormID
- **tblDepositForm**: id, levelOfDamage, damageFee, tblCostumeID
- **tblRentalPaymentForm**: id, rentalFee, tblCostumeID, tblReturnFormID
- **tblLateReturnPenalty**: id, daysLate, penaltyAmount, tblReturnFormID
- **tblDamageFeeForm**: id, levelOfDamage, damageFee, tblCostumeID
- **tblWarehouseEntry**: id, conditionWhenReturned, tblCostumeColumn, tblReturnFormID, tblCostumeID
- **tblRentalInspection**: id, conditionBeforeRenting, tblCostumeColumn, tblRentalFormID, tblCostumeID
- **tblBill**: id, totalAmount, createDate, tblReturnFormID, tblDepositFormColumn, tblLateReturnPenaltyID, tblDamageFeeFormID, tblDamageFeeFormID2, tblRentalPaymentFormID, tblUserID
- **tblUser**: id, fullName, role

3. Consider the quantity relationships among entity classes

- 1 **Customer** - N **RentalForm**
- 1 **RentalForm** - N **RentalInspection**
- 1 **RentalForm** - 1 **ReturnForm**
- 1 **ReturnForm** - N **RentalPaymentForm**

- 1 **ReturnForm** - N **LateReturnPenalty**
- 1 **ReturnForm** - N **WarehouseEntry**
- 1 **ReturnForm** - N **Bill**
- 1 **Costume -** N **RentalInspection**, N **WarehouseEntry**, **RentalPaymentForm**, N **DepositForm**, N **DamageFeeForm**
- 1 **Bill -** 1 **User**

4. Config the key columns for table

Primary key: Primary key is set with id attribute of corresponding tables
Foreign key:

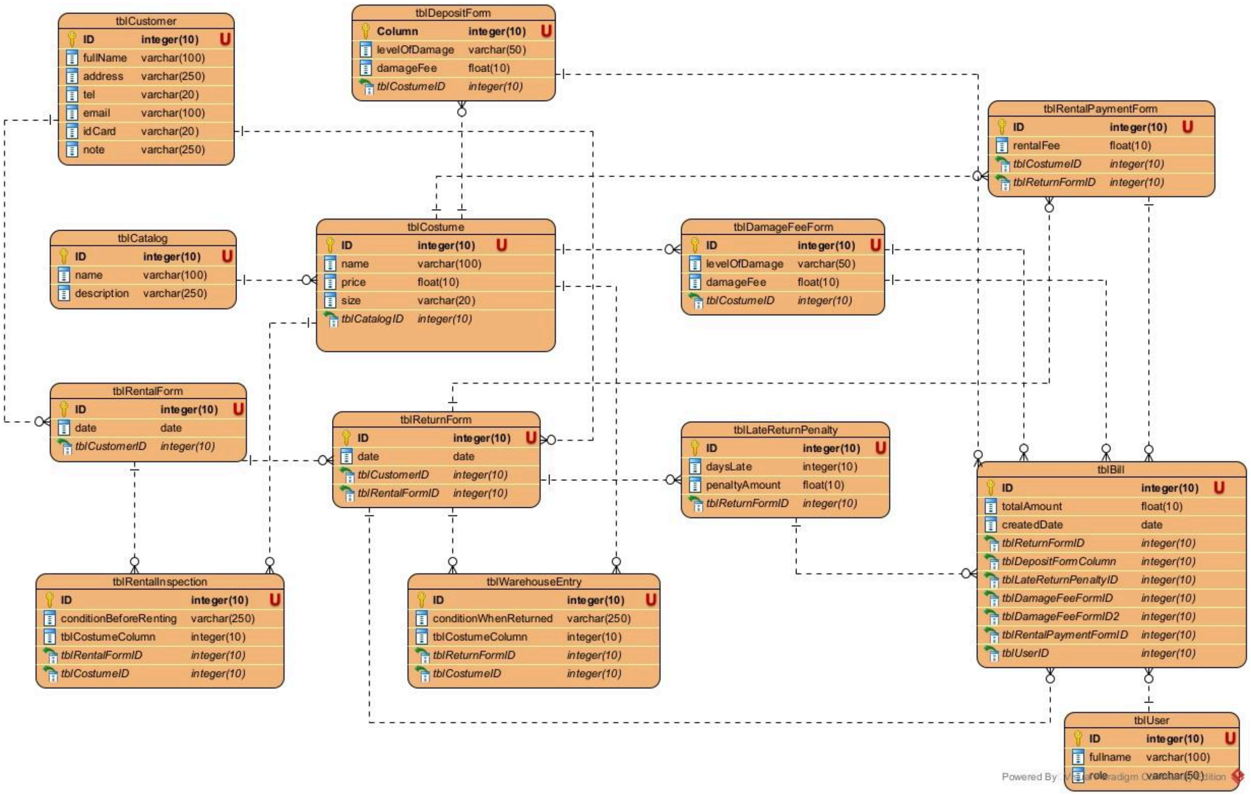
- 1 tblCustomer - N tblRentalForm → tblRentalForm has a foreign key tblCustomerID
- 1 tblRentalForm - 1 tblReturnForm → tblReturnForm has a foreign key tblRentalFormID
- 1 tblReturnForm - N tblRentalPaymentForm → tblRentalPaymentForm has a foreign key tblReturnFormID
- 1 tblReturnForm - N tblLateReturnPenalty → tblLateReturnPenalty has a foreign key tblReturnFormID
- 1 tblReturnForm - N tblWarehouseEntry → tblWarehouseEntry has a foreign key tblReturnFormID
- 1 tblCostume - N tblWarehouseEntry → tblWarehouseEntry has a foreign key tblCostumeID
- 1 tblCostume - N tblRentalInspection → tblRentalInspection has a foreign key tblCostumeID
- 1 tblCostume - N tblRentalPaymentForm → tblRentalPaymentForm has a foreign key tblCostumeID
- 1 tblCostume - N tblDepositForm → tblDepositForm has a foreign key tblCostumeID
- 1 tblCostume - N tblDamageFeeForm → tblDamageFeeForm has a foreign key tblCostumeID
  1 tblRentalForm - N tblRentalInspection → tblRentalInspection has a foreign key tblRentalFormID
- 1 tblCatalog - N tblCostume → tblCostume has a foreign key tblCatalogID
- 1 tblReturnForm - N tblBill → tblBill has foreign key tblReturnFormID

- 1 tblUser - N tblBill → tblBill has foreign key tblUserID

5. Eliminate redundancy attributes

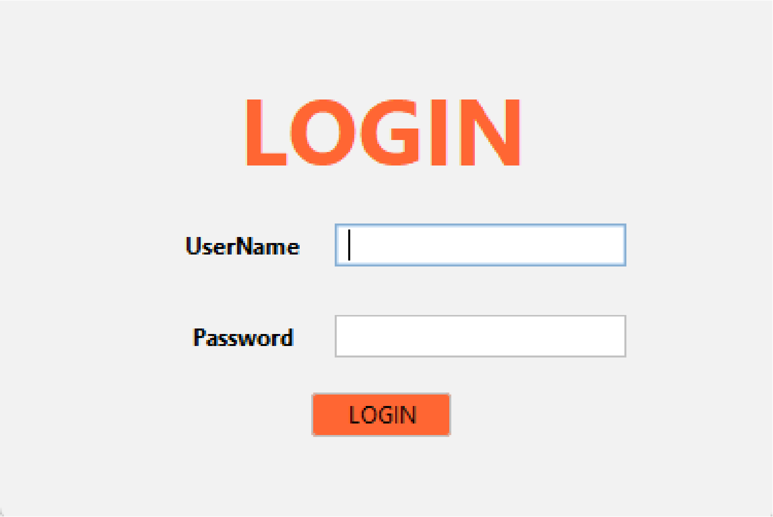
- Remove repeated damageFee or condition fields if they appear in multiple tables unnecessarily
- Consider combining similar fields where appropriate (e.g. conditionBeforeRenting and conditionWhenReturned could be normalized)

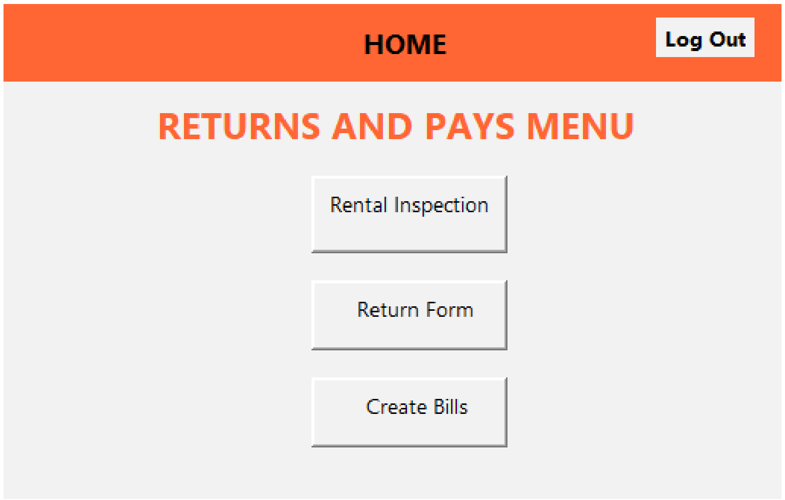
6. Entity relationship diagram

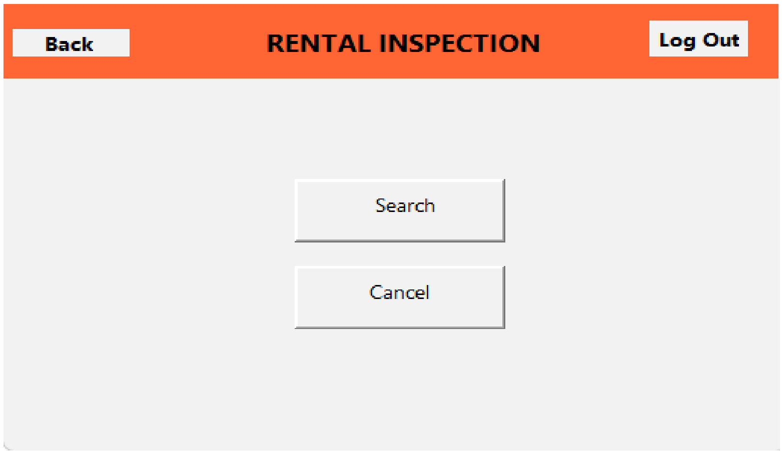


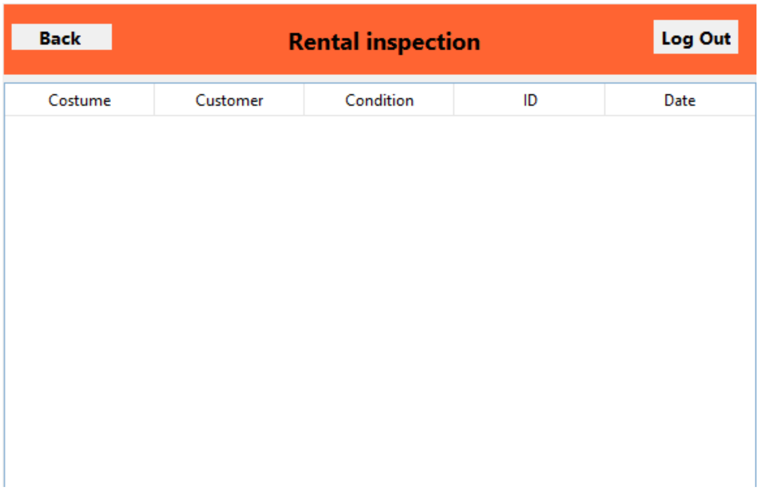
III. Static design

1. Interface Design


LOGIN
UserName
Password
LOGIN

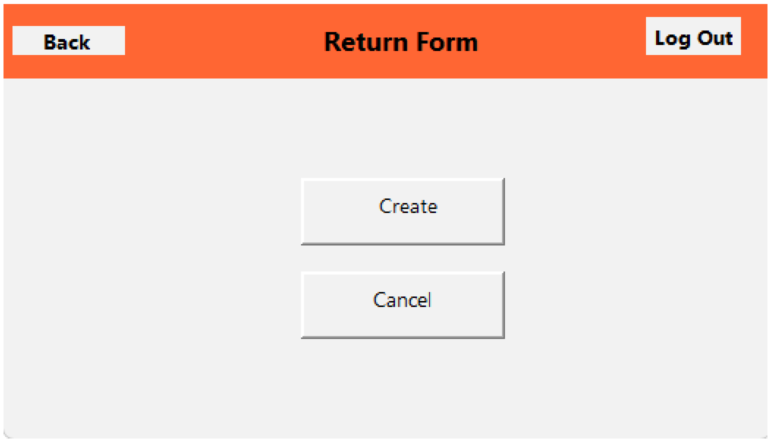

HOME
Log Out
RETURNS AND PAYS MENU
Rental Inspection
Return Form
Create Bills


Back
RENTAL INSPECTION
Log Out
Search
Cancel

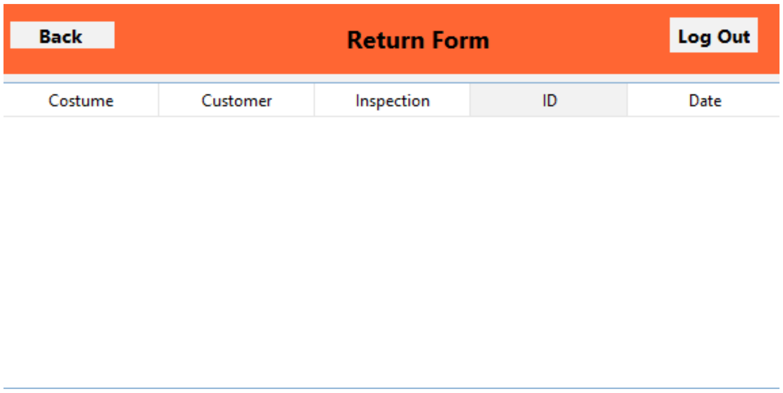


Rental inspection

Costume	Customer	Condition	ID	Date

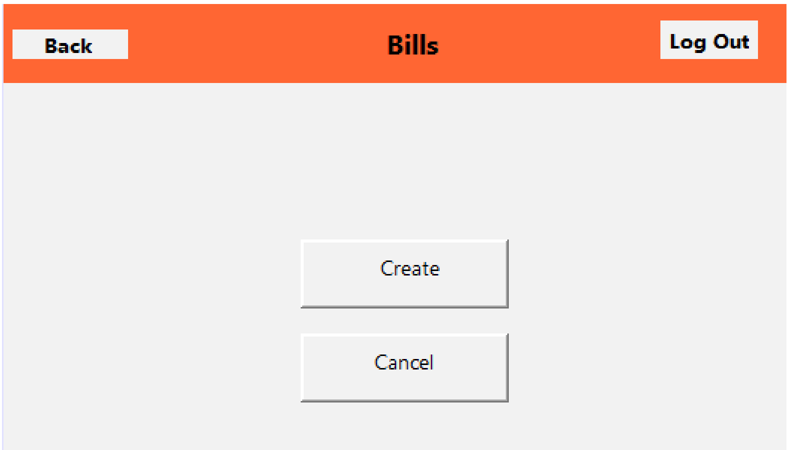


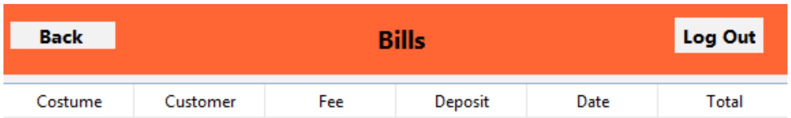
Return Form



Return Form

Costume	Customer	Inspection	ID	Date





Costume	Customer	Fee	Deposit	Date	Total

2. Design class diagram

Class: UserDAO

- Action: checkLogin()
- Input: username, password (User)
    - ○ checkLogin(username: String, password: String) : usable
    - ○ checkLogin(u: User) : usable
    - ○ checkLogin(us: User[]) : usable but complex → Chọn: checkLogin(u: User)

- Output: boolean, username/password, role
  - boolean : usable
  - User : usable but input already User → Kết luận: checkLogin(u: User): boolean

Class: RentalDAO

- Action: getRentalInspection()
- Input: No Input → Không cần xét input.
- Output: List of Rental
  - Rental[] : usable
  - List<Rental> : usable
  - ArrayList<Rental> : usable → Kết luận: getRentalInspection(): Rental[]

Class: ReturnDAO

- Action: createReturn()
- Input: Return
  - createReturn(r: Return) : usable
  - createReturn(rs: Return[]) : complex, hạn chế → Chọn: createReturn(r: Return)
- Output: boolean

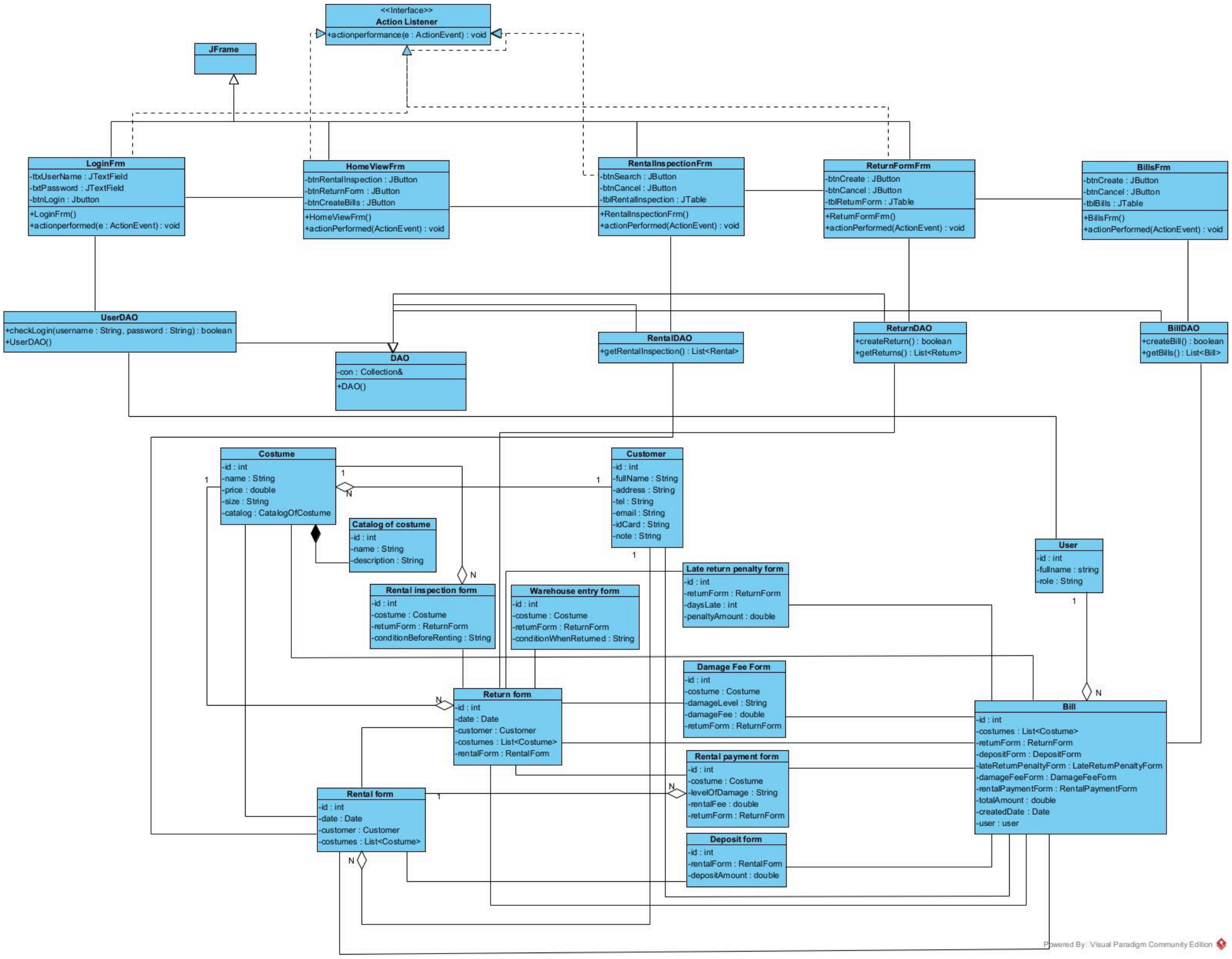
    - boolean : usable → Kết luận: createReturn(r: Return): boolean

- Action: getReturns()
- Input: No Input
- Output: List of Return
    - Return[] : usable
    - List<Return> : usable
    - ArrayList<Return> : usable → Kết luận: getReturns(): Return[]

Class: BillDAO

- Action: createBill()
- Input: Bill
    - createBill(b: Bill) : usable
    - createBill(bs: Bill[]) : complex → Chọn: createBill(b: Bill)
- Output: boolean
    - boolean : usable → Kết luận: createBill(b: Bill): boolean
- Action: getBills()
- Input: date (Bill)
    - getBills(date: Date) : usable

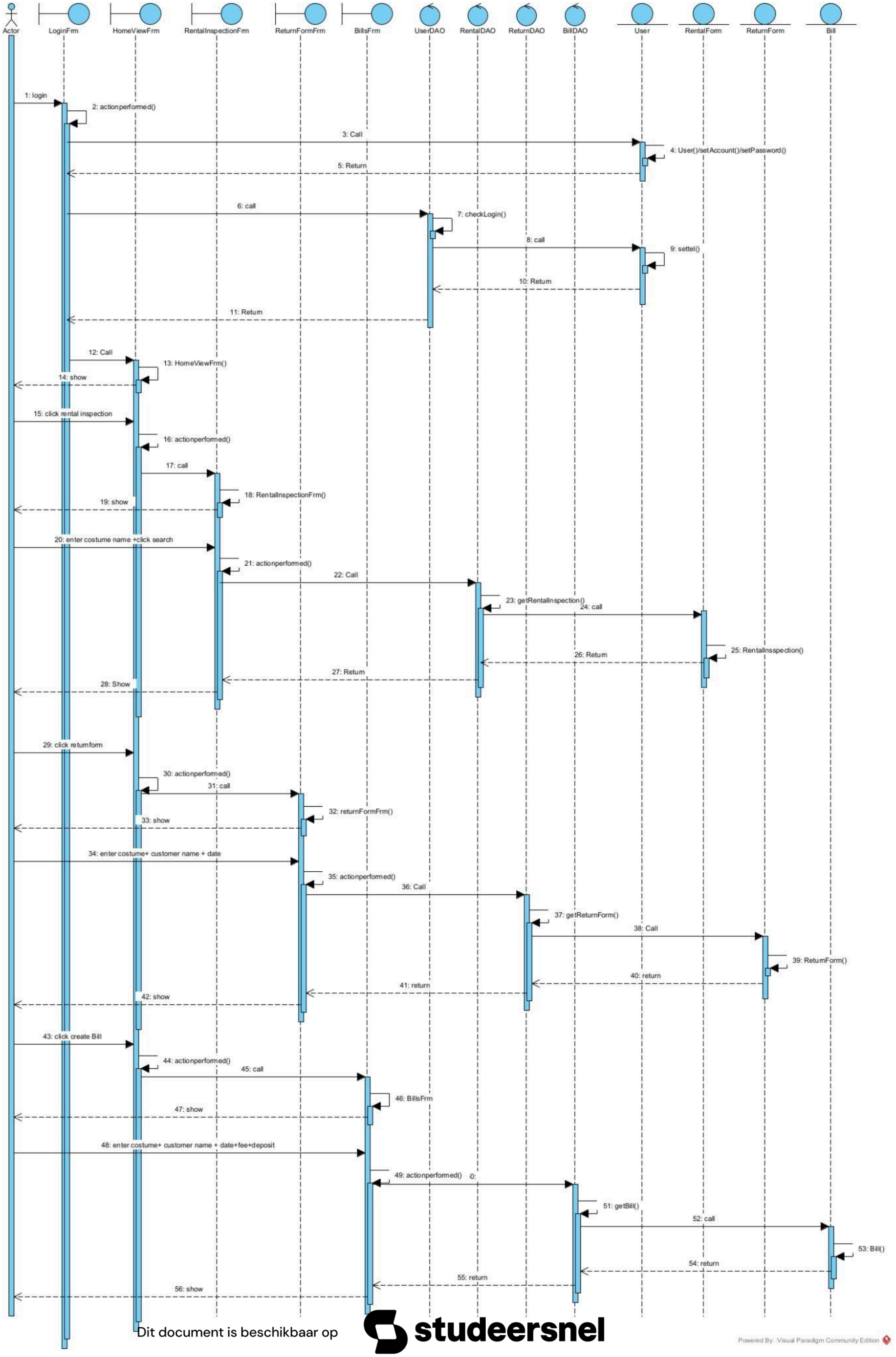
    - getBills(b: Bill) : usable
    - getBills(bs: Bill[]) : complex → Chọn: getBills(date: Date)
- Output: List of Bill
    - Bill[] : usable
    - List<Bill> : usable
    - ArrayList<Bill> : usable → Kết luận: getBills(date: Date): Bill[]



IV. Sequence Diagram:

1. Supplier calls the Warehouse Manager to provide costume.
2. Warehouse Manager enters username and password and clicks the login button in LoginFrm.
3. Method actionPerformed() of LoginFrm is triggered.
4. Method actionPerformed() creates a new User object.
5. User object packs username and password into itself.
6. User object returns to actionPerformed().
7. Method actionPerformed() calls checkLogin() method of UserDAO.
8. Method checkLogin() checks login credentials.
9. checkLogin() calls User.setId(), User.setFullname(), and User.setRole() to set more information.
10. User object updates attributes id, fullname, and role.
11. User object returns to checkLogin().
12. checkLogin() method returns result to actionPerformed().
13. actionPerformed() calls constructor of HomeViewFrm.
14. HomeViewFrm constructor is triggered.
15. HomeViewFrm screen is displayed to Warehouse Manager.
16. Warehouse Manager clicks on the rental inspection button.
17. Method actionPerformed() of HomeViewFrm is triggered.
18. Method actionPerformed() calls constructor of RentalInspectionFrm.
19. RentalInspectionFrm constructor is triggered.
20. RentalInspectionFrm screen is displayed.
21. Warehouse Manager enters costume name and clicks search button.
22. Method actionPerformed() of RentalInspectionFrm is triggered.
23. Method actionPerformed() calls getRentalInspection() method of RentalDAO.
24. Method getRentalInspection() is triggered.
25. Method getRentalInspection() calls Rental to pack the result.
26. Rental object packs rental information.
27. Rental object returns to getRentalInspection().
28. getRentalInspection() returns rental list to actionPerformed().
29. actionPerformed() displays results in RentalInspectionFrm.
30. Warehouse Manager clicks return form button.

31. Method actionPerformed() of RentalInspectionFrm is triggered.
32. actionPerformed() calls constructor of ReturnFormFrm.
33. ReturnFormFrm constructor is triggered.
34. ReturnFormFrm screen is displayed.
35. Warehouse Manager enters customer name and date, clicks search.
36. Method actionPerformed() of ReturnFormFrm is triggered.
37. actionPerformed() calls getReturnForm() method of ReturnDAO.
38. getReturnForm() method is triggered.
39. getReturnForm() calls ReturnForm to pack the return form info.
40. ReturnForm object packs the information.
41. ReturnForm object returns to getReturnForm().
42. getReturnForm() returns result to actionPerformed().
43. actionPerformed() displays return information on screen.
44. Warehouse Manager clicks on Create Bill button.
45. Method actionPerformed() of ReturnFormFrm is triggered.
46. actionPerformed() calls constructor of BillsFrm.
47. BillsFrm constructor is triggered.
48. BillsFrm screen is displayed.
49. Warehouse Manager enters customer name, date, fee, deposit.
50. Method actionPerformed() of BillsFrm is triggered.
51. actionPerformed() calls getBill() method of BillDAO.
52. getBill() method is triggered.
53. getBill() calls Bill to pack bill data.
54. Bill object returns to getBill(), and the bill is successfully created.


Actor
LoginFrm
HomeViewFrm
RentalInspectionFrm
ReturnFormFrm
BillsFrm
UserDAO
RentalDAO
ReturnDAO
BillDAO
User
RentalForm
ReturnForm
Bill
1: login
2: actionperformed()
3: Call
4: User()/setAccount()/setPassword()
5: Return
6: call
7: checkLogin()
8: call
9: settel()
10: Return
11: Return
12: Call
13: HomeViewFrm()
14: show
15: click rental inspection
16: actionperformed()
17: call
18: RentalInspectionFrm()
19: show
20: enter costume name +click search
21: actionperformed()
22: Call
23: getRentalInspection()
24: call
25: RentalInsspection()
26: Return
27: Return
28: Show
29: click returnform
30: actionperformed()
31: call
32: returnFormFrm()
33: show
34: enter costume+ customer name + date
35: actionperformed()
36: Call
37: getReturnForm()
38: Call
39: ReturnForm()
40: return
41: return
42: show
43: click create Bill
44: actionperformed()
45: call
46: BillsFrm
47: show
48: enter costume+ customer name + date+fee+deposit
49: actionperfomed()
0:
51: getBill()
52: call
53: Bill()
54: return
55: return
56: show
Powered By: Visual Paradigm Community Edition