THE POSTS AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY 1

INTRODUCTION TO SOFTWARE ENGINEERING

Class	**:**	**E22CQCN03-B**
Course group	**:**	**Group 6**
Group topic	**:**	**Mini football field management**
Project group	**:**	**Group 12**
Group members	**:**	Trần Minh Hiếu – B22DCCN320 Nguyễn Việt Hoàng – B22DCVT214 Nguyễn Tuấn Minh – B22DCKH077 **Bùi Minh Tùng – B22DCDT293**

MODULE 4: GOODS IMPORTING

REPORT: REVISING +TESTING

Hà Nội - 2025

Concept Exploration

Concepts and Explanations			
#	Concept	Vietnamese	Explanation
	Human-related concept		
1	Field facility owner/proprietor (yard manager)	Chủ sở hữu cơ sở/Quản lý sân (Quản lý sân)	The primary owner or manager of the football field who oversees all business operations, ensures profitability, and manages key decisions.
2	Operations manager	Quản lý vận hành	Manages daily activities, schedules, and ensures smooth operations of the facility, including customer service and field management.
3	General manager	Giám đốc điều hành	Oversees all departments, ensuring they work together efficiently. Handles major business strategies and administrative policies.
4	Financial administrator	Nhà quản lý tài chính	Manages financial records, invoices, budgets, and ensures all transactions, salaries, and expenses are recorded properly.
5	Marketing manager	Nhà quản lý tiếp thị	Promotes the facility, organizes advertising campaigns, manages social media, and develops strategies to attract more customers.
6	HR manager	Nhà quản lý nhân sự	Responsible for hiring, training, and managing staff, along with handling employee relations and payroll.
7	IT administrator	Quản trị viên công nghệ thông tin	Manages technical systems, software, databases, and ensures smooth operation of digital tools and security.
8	Facilities maintenance manager	Quản lý bảo trì cơ sở vật chất	Ensures that the football field, equipment, and other facilities are maintained in good condition and meet safety standards.
9	Front desk receptionist	Lễ tân	Greets customers, handles inquiries, manages bookings, and assists in basic administrative work.
10	Booking coordinator	Điều phối viên đặt chỗ	Manages the scheduling and reservations of football fields, ensuring availability and conflict-free bookings.

11	Field attendant	Nhân viên sân	Assists players, ensures equipment is ready, and helps maintain the playing field.
12	Equipment manager	Quản lý thiết bị	Oversees all sports equipment, ensures they are in good condition, and manages inventory.
13	Maintenance staff	Nhân viên bảo trì	Handles repairs and upkeep of the facility, including field maintenance, lighting, and drainage.
14	Cleaning staff	Nhân viên vệ sinh	Ensures cleanliness of the facility, including changing rooms, shower room, and general areas.
15	Security personnel	Nhân viên an ninh	Maintains order, ensures safety, and enforces facility rules to prevent disturbances.
16	First aid/medical staff	Nhân viên y tế/cấp cứu	Provides medical assistance in case of injuries or emergencies on the field.
17	Cashier/payment processor	Thu ngân/nhân viên xử lý thanh toán	Handles transactions, receives payments, and provides receipts for bookings and other purchases.
18	Inventory clerk	Nhân viên kiểm kê	Manages stock levels of equipment and supplies, ensuring proper documentation of usage and purchases.
19	Canteen staff	Nhân viên căng tin	Prepares and serves food and drinks for players and visitors.
20	Guard	Bảo vệ	Protects the premises and ensures security of the facility and customers.
21	Customer	Khách hàng	Individuals or teams who book and use the football field for matches, training, or recreational play.
22	Equipment suppliers/vendors	Nhà cung cấp thiết bị/nhà thầu	Companies or individuals who supply football-related equipment such as balls, goals, nets, and training gear.
23	Food and beverage providers	Nhà cung cấp thực phẩm và đồ uống	Vendors who supply snacks, drinks, and refreshments for the facility’s canteen.

24	Maintenance service providers	Nhà cung cấp dịch vụ bảo trì	Companies or individuals responsible for repairs, field upkeep, and infrastructure maintenance.
25	Marketing partners	Đối tác tiếp thị	Businesses or agencies that assist in promoting the facility through sponsorships or advertisements.
26	Advertisements	Quảng cáo	Sponsored ads placed within the facility, website, or social media to generate extra revenue.
27	Insurance providers	Nhà cung cấp bảo hiểm	Companies that offer coverage for facility damages, injuries, and liabilities.
28	Utility service providers	Nhà cung cấp dịch vụ tiện ích	Providers of electricity, water, and other essential services to run the facility.
29	Payment gateway providers	Nhà cung cấp dịch vụ cổng thanh toán	Companies that facilitate online payments for bookings and services.
30	Technology vendors	Nhà cung cấp công nghệ	Suppliers of software, hardware, and digital systems used for managing bookings, payments, and security.
31	Transportation service providers	Nhà cung cấp dịch vụ vận tải	Companies or individuals that offer transport services for teams or customers visiting the facility.
	Object-related concept		
32	Footballs/soccer balls	Bóng	Essential sports equipment used in all matches and training sessions.
33	Training cones	Các cột chóp nhỏ	Used for agility training, drills, and marking specific areas on the field.
34	Agility ladders	Thang agility	Helps improve foot speed and coordination during training.
35	Goal nets	Lưới cầu môn	Attached to goalposts to catch the ball when a goal is scored.

36	Training bibs/pinnies	Áo bib/Pinnies tập luyện	Worn by players to differentiate teams during practice sessions.
37	Portable goals	Cầu môn di động	Smaller goals used for training, youth games, or temporary field setups.
38	Corner flags	Cờ góc	Placed at the four corners of the field to mark boundaries.
39	Field markers	Dấu vạch sân	Used to outline playing areas and specific zones on the field.
40	Referee equipment (whistles, cards, etc.)	Thiết bị trọng tài (Còi, thẻ, v.v.)	Tools used by referees to enforce rules and maintain fair play.
41	First aid kits	Bộ sơ cứu	Contains medical supplies to treat minor injuries on-site.
42	Cooler box	Thùng lạnh	Used to store and provide cold beverages during matches and training sessions.
43	Ball pumps	Bơm bóng	Used to inflate soccer balls to the required pressure.
44	Equipment storage containers	Container lưu trữ thiết bị	Helps organize and store sports equipment properly.
45	Maintenance tools	Công cụ bảo trì	Tools required for field upkeep and general repairs.
46	Field cleaning equipment	Thiết bị làm sạch sân	Items used to maintain cleanliness of the playing surface and facility.
47	Artificial turf maintenance supplies	Dụng cụ bảo trì thảm nhân tạo	Specialized tools and materials to care for artificial playing surfaces.
48	Lighting equipment/bulbs	Thiết bị chiếu sáng/bóng đèn	Used to ensure proper illumination of the field for night games and training.

49	Scoreboards/timing devices	Bảng điện tử/Thiết bị thời gian	Displays game scores and match time for players and spectators.
50	Benches/seating	Ghế/Chỗ ngồi	Provides seating for players, staff, and spectators.
51	Lockers/storage units	Tủ/lưu trữ	Used for storing players' belongings securely.
52	Shower	Phòng tắm	Facility for players to clean up after a match or training session.
53	Field surface materials	Vật liệu bề mặt sân	Materials used to construct and maintain the playing surface.
54	Boundary markings	Dấu vạch giới hạn	Lines that define the playing area and different field zones.
55	Goal posts	Cột gôn	A board where goals are scored in a match.
56	Fencing materials	Vật liệu hàng rào	Used for enclosing the field to maintain security and prevent unauthorized access.
57	Lighting fixtures	Thiết bị chiếu sáng	Installed to provide adequate visibility during nighttime events.
58	Drainage systems	Hệ thống thoát nước	Ensures proper water flow and prevents field flooding.
59	Changing room fixtures	Thiết bị phòng thay đồ	Includes lockers, benches, and other furnishings in player dressing rooms.
60	Shower facilities	Tiện nghi phòng tắm	Equipped with running water for players to use post-match.
61	HVAC equipment	Thiết bị HVAC	Heating, ventilation, and air conditioning systems for indoor areas.

62	Security systems	Hệ thống an ninh	Cameras, alarms, and access control measures for facility protection.
63	Booking ledgers/scheduling tools	Sổ đăng ký đặt chỗ/công cụ lịch trình	Records and manages reservations for field usage.
64	Receipt books/financial records	Sổ biên nhận/hồ sơ tài chính	Documents all transactions and payments.
65	Invoices	Hóa đơn	Bills issued for services rendered or products sold.
66	Inventory tracking forms	Biểu mẫu theo dõi tồn kho	Used to monitor stock levels of equipment and supplies.
67	Equipment inspection checklists	Danh sách kiểm tra kiểm tra thiết bị	Ensures regular checking of equipment for safety and functionality.
68	Maintenance logs	Nhật ký bảo trì	Records past repairs, maintenance work, and future servicing schedules.
69	Purchase orders	Đơn đặt hàng mua	Documents requests for purchasing supplies and equipment.
70	Delivery notes	Ghi chú giao hàng	Confirms receipt of items ordered from suppliers.
71	Warranty documents	Tài liệu bảo hành	Details manufacturer guarantees and terms of replacement or repair.
72	Certificates of quality	Chứng nhận chất lượng	Certifies that equipment meets required standards.
73	Inventory database records	Hồ sơ cơ sở dữ liệu tồn kho	Digital records of stock levels and inventory usage.
74	Supplier catalogs	Danh mục nhà cung cấp	Lists available products from suppliers.

75	Price lists	Danh sách giá	Contains updated pricing information for goods and services.
76	Product specifications	Thông số kỹ thuật sản phẩm	Detailed descriptions and features of equipment and supplies.
77	Import/export documentation	Tài liệu nhập khẩu/xuất khẩu	Required paperwork for importing or exporting goods.
78	Equipment lifecycle records	Hồ sơ vòng đời thiết bị	Tracks the lifespan of various equipment items.
79	Depreciation schedules	Lịch trình khấu hao	Calculates the reduction in value of assets over time.
80	Equipment usage statistics	Thống kê sử dụng thiết bị	Data on how frequently equipment is used and its effectiveness.
81	Maintenance schedules	Lịch trình bảo trì	Planned routine maintenance to keep equipment functional.
82	Replacement forecasts (optional)	Dự báo thay thế (nếu có)	Predicts when equipment will need replacing based on usage trends.
83	Rental Schedules	Lịch trình cho thuê	Documents the reservation of equipment for temporary use.
	Action-related concept		
84	Ordering supplies	Đặt hàng vật tư	Process of purchasing necessary equipment and materials.
85	Receiving shipments	Nhận hàng	Accepting deliveries and checking for accuracy.
86	Inspecting delivered goods	Kiểm tra hàng hóa giao	Ensuring received items meet quality standards and order specifications.

87	Logging new inventory	Ghi nhận tồn kho mới	Adding newly received stock into the system.
88	Updating stock quantities	Cập nhật số lượng hàng tồn kho	Adjusting inventory records based on new stock levels.
89	Categorizing equipment	Phân loại thiết bị	Sorting equipment into appropriate groups for easier management.
90	Tagging/labeling items	Dán nhãn/đánh dấu các mặt hàng	Marking inventory for tracking and identification.
91	Storing equipment properly	Lưu trữ thiết bị đúng cách	Ensuring equipment is kept in a safe and organized manner.
92	Conducting inventory audits	Tiến hành kiểm kê tồn kho	Reviewing stock levels to verify accuracy and prevent discrepancies.
93	Identifying low-stock items	Xác định các mặt hàng tồn kho thấp	Flagging items that need restocking before running out.
94	Searching for suppliers	Tìm kiếm nhà cung cấp	Finding new vendors to provide equipment, materials, or services.
95	Adding new suppliers to the database	Thêm nhà cung cấp mới vào cơ sở dữ liệu	Entering details of newly onboarded suppliers for future reference.
96	Updating supplier information	Cập nhật thông tin nhà cung cấp	Keeping supplier contact details and product lists current.
97	Requesting price quotes	Yêu cầu báo giá	Asking vendors for pricing on required goods and services.
98	Approving purchase orders	Phê duyệt đơn đặt hàng mua	Authorizing the procurement of goods and services from suppliers.
99	Scheduling deliveries	Lên lịch giao hàng	Coordinating the arrival of supplies to avoid disruptions.

100	Tracking order status	Theo dõi tình trạng đơn hàng	Monitoring shipment progress to ensure timely delivery.
101	Communicating with vendors	Giao tiếp với nhà cung cấp	Maintaining relationships with suppliers and addressing concerns.
102	Resolving delivery issues	Giải quyết vấn đề giao hàng	Handling problems such as late shipments or incorrect orders.
103	Processing invoices	Xử lý hóa đơn	Reviewing and approving payment documents from suppliers.
104	Verifying pricing	Xác minh giá cả	Ensuring billed amounts match agreed-upon prices.
105	Authorizing payments	Phê duyệt thanh toán	Granting approval for financial transactions related to purchases.
106	Recording expenses	Ghi nhận chi phí	Logging all costs associated with operations and purchases.
107	Reconciling delivery notes with orders	Đối chiếu ghi chú giao hàng với đơn hàng	Checking received goods against order details to ensure accuracy.
108	Calculating inventory value	Tính giá trị tồn kho	Determining the total worth of stock items.
109	Processing returns/refunds	Xử lý trả lại/hoàn tiền	Managing defective or incorrect items that need to be sent back.
110	Managing warranties	Quản lý bảo hành	Keeping track of product guarantees and service agreements.
111	Forecasting future expenses	Dự báo chi phí trong tương lai	Predicting upcoming costs for better financial planning.
112	Logging into app	Đăng nhập vào ứng dụng	Accessing the system for booking, inventory, and financial tasks.

113	Searching product database	Tìm kiếm cơ sở dữ liệu sản phẩm	Finding specific items within the system.
114	Filtering search results	Lọc kết quả tìm kiếm	Sorting product lists based on different criteria.
115	Adding items to the import list	Thêm mặt hàng vào danh sách nhập khẩu	Including products for procurement or stocking.
116	Removing items from the import list	Xóa mặt hàng khỏi danh sách nhập khẩu	Taking off unnecessary or incorrect items from purchase lists.
117	Adjusting quantities	Điều chỉnh số lượng	Modifying stock levels before finalizing orders.
118	Calculating totals	Tính tổng số	Summing up costs for purchases, sales, or invoices.
119	Generating reports	Tạo báo cáo	Creating summaries of financials, inventory, or usage data.
120	Printing documents	In tài liệu	Producing hard copies of invoices, purchase orders, and records.
121	Exporting data	Xuất dữ liệu	Saving system data for external use or backups.
122	Adding invoices	Thêm hóa đơn	Entering new billing information into the financial system.
123	Removing invoices	Xóa hóa đơn	Deleting incorrect or outdated billing records.
124	Scheduling equipment inspections	Lên lịch kiểm tra thiết bị	Planning routine checks for equipment functionality.
125	Documenting equipment condition	Ghi nhận tình trạng thiết bị	Recording the status of items to track wear and tear.

126	Marking damaged items	Đánh dấu các thiết bị hư hỏng	Labeling broken or malfunctioning equipment for repair or replacement.
127	Initiating repair procedures	Khởi động quy trình sửa chữa	Starting the process of fixing damaged equipment.
128	Tracking repair status	Theo dõi tình trạng sửa chữa	Monitoring ongoing maintenance and repair work.
129	Recording maintenance history	Ghi nhận lịch sử bảo trì	Keeping logs of past repairs and servicing.
130	Planning equipment replacement	Lập kế hoạch thay thế thiết bị	Scheduling when worn-out items should be replaced.
131	Archiving obsolete items	Lưu trữ các thiết bị đã hết hạn sử dụng	Removing outdated equipment from active inventory.
132	Disposing of unusable equipment	Vứt bỏ các thiết bị không thể sử dụng	Properly discarding equipment that is beyond repair.
133	Recycling materials when possible	Tái chế vật liệu khi có thể	Reusing or disposing of materials in an eco-friendly way.
		Booking	
134	Deposit	Đặt cọc	A pre-payment made to secure a booking or service.
135	Book 1 court	Sổ đăng ký đặt chỗ sân 1	Reserving a single football field for a session.
136	Book 2 adjacent small courts into 1 large court	Đặt sân 2 các sân nhỏ liền kề thành 1 sân lớn	Combining two smaller fields into one larger playing area.
137	Book 4 adjacent small courts into 1 large court	Đặt sân 4 các sân nhỏ liền kề thành 1 sân lớn	Merging four small fields into a bigger playing space.

138	Booking slip	Phiếu đặt sân	A receipt or document confirming a reservation.
139	Clicks on the correct customer name with the current customer	Nhấp vào tên khách hàng đúng với khách hàng hiện tại	Selecting the right user in the system for booking or payment purposes.
140	If the customer first comes to book a court, must add a new one	Nếu khách hàng lần đầu tiên đến đặt sân, phải thêm mới khách hàng	Creating a new customer profile if they are booking for the first time.
141	Multi-session booking	Đặt sân nhiều phiên	Reserving multiple game slots in one transaction.
		Update used items of the rental session	
142	Session check-in/checkout	Kiểm tra/kiểm tra phiên	Tracking customer arrival and departure for booked sessions.
143	Search for goods by name	Tìm kiếm hàng hóa theo tên	Finding specific inventory items using their names.
144	Enter unit and quantities	Nhập đơn vị và số lượng	Inputting the amount of goods being processed.
145	Total amount of customer	Tổng số tiền khách hàng	Calculating the final bill for a customer.
		Payment	
146	Session payment	Thanh toán phiên	Processing fees for booked sessions.
147	Late payment fee	Phí thanh toán muộn	Additional charges for overdue payments.
148	Changing session detail	Chi tiết thay đổi phiên	Modifying the time, date, or duration of a booked session.

149	Confirming payment	Xác nhận thanh toán	Finalizing and approving a customer's transaction.
		Goods importing	
150	Adding new good (if the good does not appear in the system)	Thêm hàng hóa mới (nếu hàng hóa không có trong hệ thống)	Registering a new item in the inventory database.
151	Update new good (if the good appears in the system -> increase the quantity)	Cập nhật hàng hóa mới (nếu hàng hóa có trong hệ thống -> tăng số lượng)	Adjusting stock levels when additional units arrive.
152	Return details of the imported invoice	Chi tiết trả lại hóa đơn đã nhập khẩu	Recording and reviewing the specifics of received shipments.
153	Import success confirmation	Xác nhận thành công nhập khẩu	Verifying that a new stock entry has been added successfully.
	Policy		
154	Loss of properties	Mất mát tài sản	A policy stating the facility is not responsible for lost items.
155	No use of alcoholic drinks	Cấm sử dụng đồ uống có cồn	Prohibition of alcohol on the premises.
156	Damage properties	Hư hỏng tài sản	Rules against vandalizing or misusing facility equipment and spaces.
157	No weapons and fireworks	Cấm vũ khí và pháo	Banning dangerous items to ensure safety.
158	No drugs	Cấm ma túy	Strict prohibition of illegal substances.
159	Not hold responsible for any injuries/accident	Không chịu trách nhiệm về bất kỳ chấn thương/ tai nạn nào	A liability waiver stating the facility is not accountable for injuries.

Business model

A. Business Model by Natural Language

1. Object & Scope

a. Object

- This is a desktop-based application for managing mini football field. It will be internally used inside a mini football field and support only one mini football field.

b. Scope

- Application type: Desktop Based (Business Management Software) which could be installed on many computers of the mini football field employees. However, the database is stored in the mini football field server.
  - User: Only staff could use:
    - Receptionist
    - Staff
    - Yard manager
  - Function:
    - Booking
    - Update used item of the rental session
    - Customer paying
    - Goods importing

2. User & function (Who, what to do)

- Receptionist:
  - Book field(s) on the site customers (with the requirement of the customer)
  - Cancel booking (with the requirement of the customer)
  - Check in (with the requirement of the customer)
  - Check out (with the requirement of the customer)
  - Process the payment (with the requirement of the customer)
- Canteen Staff:
  - Update used items (with the requirement of the customer)
- Yard manager:
  - Import goods (with the requirement of the suppliers)

3. How function work

- Booking:

  → A customer comes to book a mini football field
  →The staff asks the customer what period of time that the customer wants to book.
  → The staff logs in into the system
  → The staff's UI is appeared, it has the following options:
  - Booking
  - Cancel Booking
  - Update used items
  - Process the payment

  → The staff selects the Booking function
  → The system displays the interface to find an empty court according to the time slot
  → The staff enters the time slot + select the type of court as requested by the customer + click search
  → The system displays a list of available courts according to the selected time slot
  → Clicks on a court
  → The system displays an interface to fill in customer information

→ The staff enters customer’s name and search
→ The system displays a list of customers whose names contains the entered keyword. Clicks on the correct customer name with the current customer. If the customer first comes to book a court, must add a new one
→ The system displays the interface to enter the time period of the start date, end date of the booking (preferred to book by quarter)
→ Clicks confirm
→ The system displays a booking slip with full customer information, booking information, booking price, booking time slot, total number sessions according to the selected time, the estimated total amount and the deposit amount
→ Clicks confirm
→ The system prints the booking slip and updates it to the database.

- *Update used items of the rental session*:
  → A customer arrives to receive the court and return the court for that session
  → The staff logs in into the system
  → The staff’s UI is appeared, it has the following options:
    - Booking
    - Cancel Booking
    - Update used items
    - Process the payment

  → The staff selects the Update used items function
  → The menu UI appeared with an input text to enter
  → The staff enters the customer's name + click search
  → The system displays a list of customers with the name entered
  → The staff selects the correct customer name with the current customer information
  → The order’s UI displays a list of orders that the customer is booking
  → Clicks on the checkout button rental session 1 booking ticket
  → The system displays an interface to enter the court reception time, return time, and rent (early payment will not be reduced, but late payment will be charged more) with More items used button + Repeat the following steps until complete the list of food products that customers have used during the rental sessions: Click more items used
  → The interface to search for goods by name appears
  → Enters the name of the goods and search
  → The interface for the list of goods with the name entered appears
  → Clicks on 1 item
  → The interface to enter the quantity appears with an input text
  → Enters the quantity and confirms
  → The used item information is added to the list of used items of the session → The last line is the total amount of customers
  → Clicks to confirm
  → The system updates to the database (no payment required).

- *Customer paying*:
  → A customer comes to pay for a session
  → The staff logs in into the system
  → The staff’s UI is appeared, it has the following options:
    - Booking
    - Cancel Booking
    - Update used items
    - Process the payment

  → The staff selects the Process the payment function
  → The menu UI appeared with an input text to enter

→ The staff enters the customer's name + click search
→ The system displays a list of customers with the name entered
→ The staff selects the correct customer name with the current customer information
→ The order’s UI displays a list of orders that the customer is booking
→ Clicks on the payment button for 1 booking ticket
→ The system displays the invoice with full customer information + 1 list of food and beverage products that the customer has used during the rental sessions as described above + the last line is the total amount paid
→ If the customer complains about a change in the quantity or information information about used items, the staff must change, update the detailed list in the corresponding invoice
→ Clicks confirm
→ The system updates to the database.

- *Goods importing*:
  → The yard manager selects the Menu to find Importing
  → The yard Importing UI appears, it has the following 2 options: goods importing and suppliers importing
  → The yard manager selects to suppliers importing
  → The supplier importing UI appears with 3 options: entering, adding and searching button
  → The yard manager enter the name of the supplier need to be searched and then click search button
  → The system displays a list of the providers whose name contains the entered keyword, each row corresponds with the information of the provider: name, address, phone, email, deposit. The yard manager clicks on the row which has the same information to the current supplier(if no rows are satisfying, it has to click on the add new supplier button to add a new supplier → The system displays the confirmation UI with: supplier's information
  → The yard manager clicks on the confirm button
  → The system announce a success alert and then, return to the importing UI
  → The yard manager selects to goods importing
  → The goods importing UI appears with 3 options: entering, adding and searching button
  → The yard manager enter the name of the goods need to be searched and then click search button
  → The system displays a list of the goods whose name contains the name just entered, each row corresponds with the information of the goods: id, name, quantity, price, provider’s name. The yard manager clicks on the row which has the same information to the current good(if no rows are satisfying, it has to click to the add new goods button to add a new goods
  → The system displays the confirmation UI with prices, quantity and confirm button
  → The yard manager have to enter the prices, the quantity (Repeat goods searching and adding process until all imported goods are selected)
  → After that yard manager needs to click to confirm button
  → The system announces a success alert and then returns to the importing UI.

4. Object related in application

- Football field complex: name, address, description.
- Court: name, description, price.
- Customer: id, name, address, phone, email, note.
- User: name, username, password, role, note.

- Voucher: id, date, owner’s information, customer’s information, court’s information, rental price per session, rental time slot of the week, start date, end date, total expected rent
- Invoice/ bill for customer:
    - + Receptionist information: name, role
    - + Customer information: name, address, phone, email
    - + Court information: id, number of courts, price
    - + List of used items: item’s name, quantity, price.
- Invoice for provider:
    - + Provider information: name, address, phone, email, deposit
    - + List of used items: item’s name, quantity, price.
- Item: id, name, quantity, price, provider.

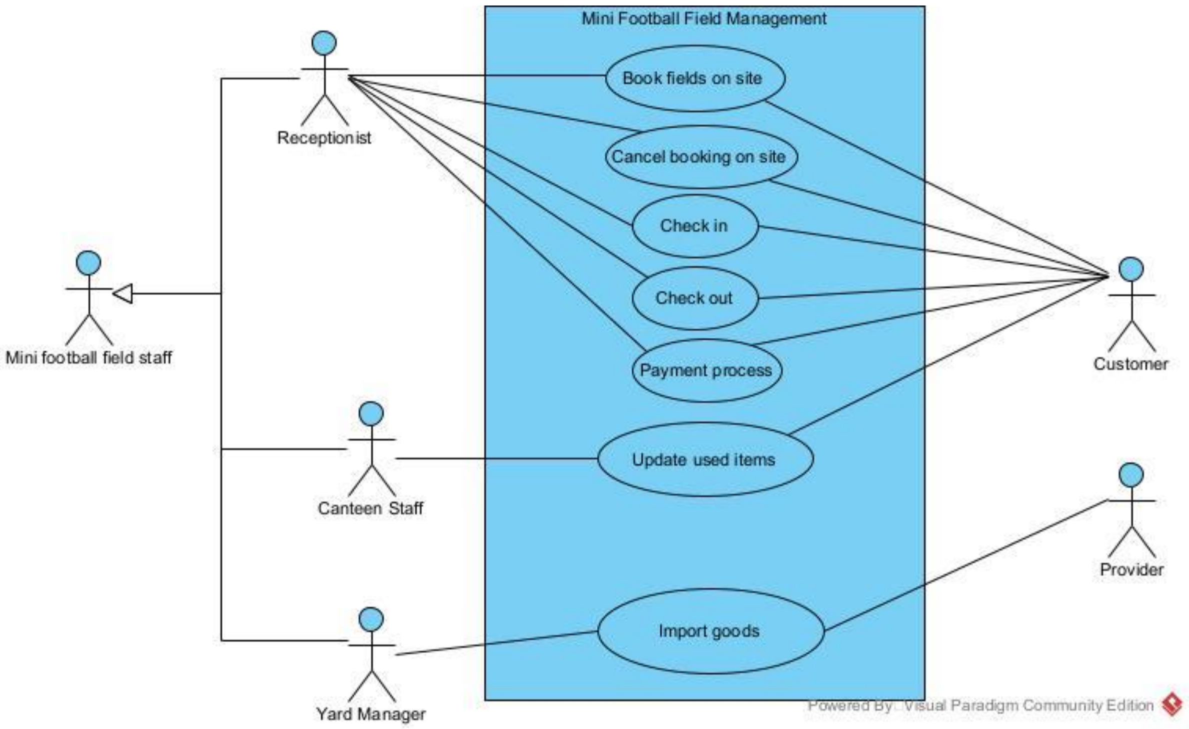
5. Relationship

- A complex has many courts. A court belongs to a complex.
- A court could be booked by many customers in different periods.
- A customer could book many courts in different periods and could also book many courts at the same time.
- 2 or 4 courts could be combined into a bigger court if those courts are adjacent.
- When making a contract to rent a court, the customer receives a voucher. A voucher is received by a customer.
- An invoice for the customer is generated when booking. An invoice for the customer may contain many items. An item may be contained in many invoices.
- An invoice for the provider contains many items. An item may be contained by many invoices for the provider.

B. Business Model by UML

1. Description in UML

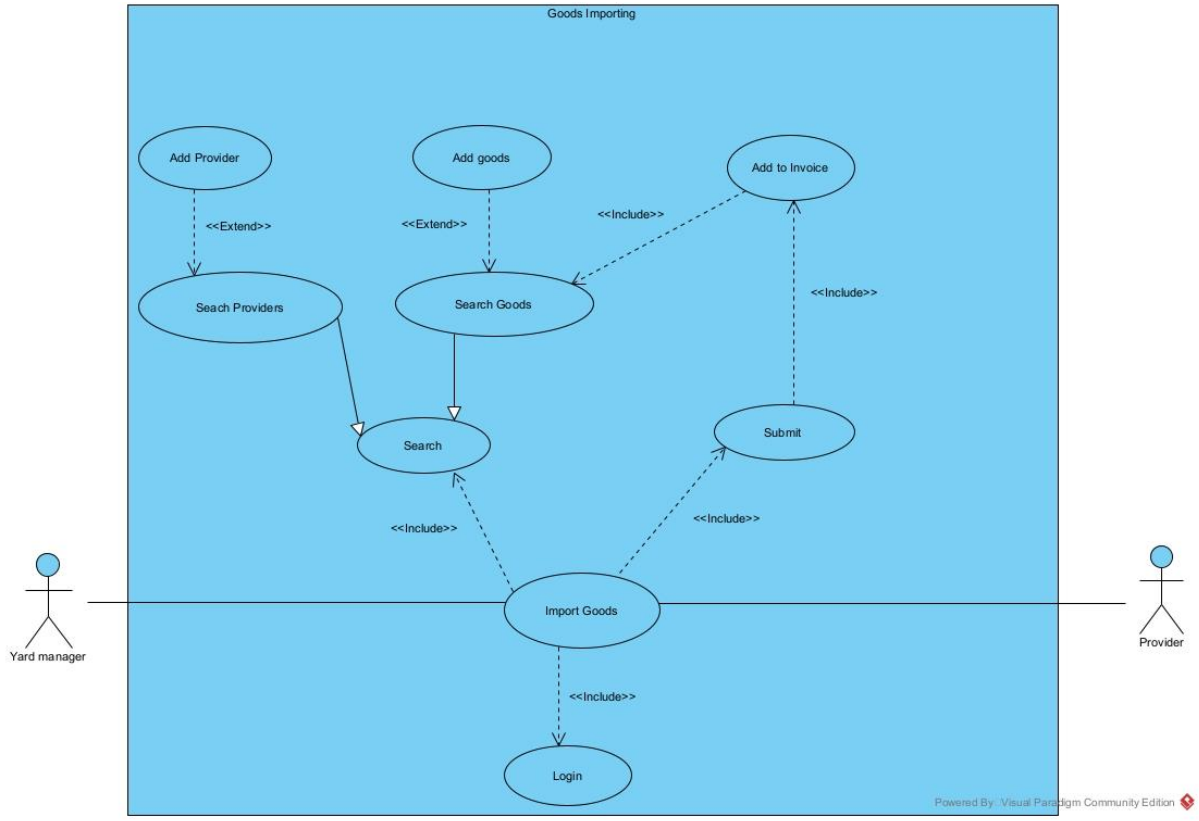
- **General Use Case of the input Q2 (Who? What to do?):**



2. Describe Use Case

- Book fields on site: This Use Case enables the Receptionist to book fields on site with the requirement of the Customer.
- Cancel booking on site: This Use Case enables the Receptionist to cancel a booking on site with the requirement of the Customer.
- Check in: This Use Case enables the Receptionist to check in with the requirement of the Customer.
- Check out: This Use Case enables the Receptionist to check out with the requirement of the Customer.
- Process Payment: This Use Case enables the Receptionist to process the payment with the requirement of the Customer.
- Update used items: This Use Case enables the Canteen Staff to update the list of food product that the Customer has use with the requirement of the Customer.
- Import goods: This Use Case enables the Yard Manager to import goods with the requirement of the Provider.

3. Detail Use Case diagram for module 4: Goods Importing



Use case description:

Login: This use case let the yard manager log in to the system

Submit: This use case enables the yard to make sure that the invoices after have all of the imported goods must be submitted to the system.

Add goods: this use case enables the yard manager to add a new good to the system (if it does not exist on the system)

Search goods: this use case enables the yard manager to search a good (by name) in the system

Add providers: this use case enables the yard manager to add a new supplier to the system(if it does not exist on the system)

Search providers: this use case enables the yard manager to search for a supplier(by name) in the system

Add to invoices: this use case enables the yard manager to add the information about goods to an invoice.

C. Analyses

1. Scenario

Scenario	Goods importing
Actor(s)	Yard manager, provider
precondition	The yard manager successfully logged in to their account.
postcondition	The yard manager successfully imported the goods provided by the provider
Main events	1.The Yard Manager open the application for goods importing 2. The login interface appears with: an input text for username, an input text for password, a button to login. 3. The Yard Manager enters username as “YM01”, password as “YM123” and then click on the login button. 4. The main UI appears with: Search and Submit 5. The Yard Manager clicks on Search.

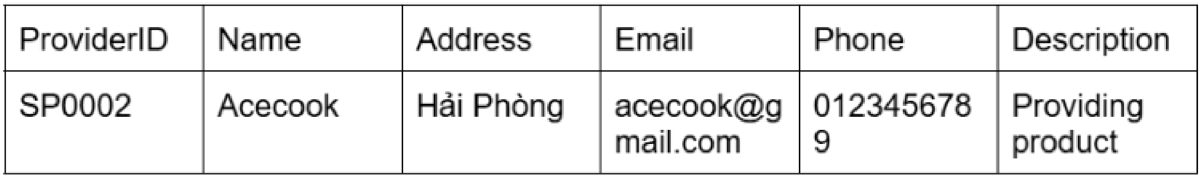
6. The Search’s UI appears with: Search Providers and Search Goods

7. The Yard Manager clicks on Search Providers

8. The Search Provider UI appears with: an input field to search providers by name, a search button, a an add provider button.

9. The Yard Manager enters a provider name as “Acecook” and then clicks on the search button.

10. The system displays the provider information with a table and a confirm button(A confirm provider button)



ProviderID	Name	Address	Email	Phone	Description
SP0002	Acecook	Hải Phòng	acecook@gmail.com	0123456789	Providing product

11. The yard manager clicks on the first line, corresponding to name: “Acecook”

12. The UI for goods importing appears with the attribute and the value of the selected goods: provider SP0002, name: Acecook, address: Hải Phòng, email: acecook@gmail.com, phone: 0123456789, description: providing products for sport.

13. After seeing the provider's information, the yard manager needs to click the confirm button to finish the import provider process.

14. The system displays a success alert, confirming that the goods have been successfully imported a then returns to the Search UI.

15. The Yard Manager clicks on Search Goods.

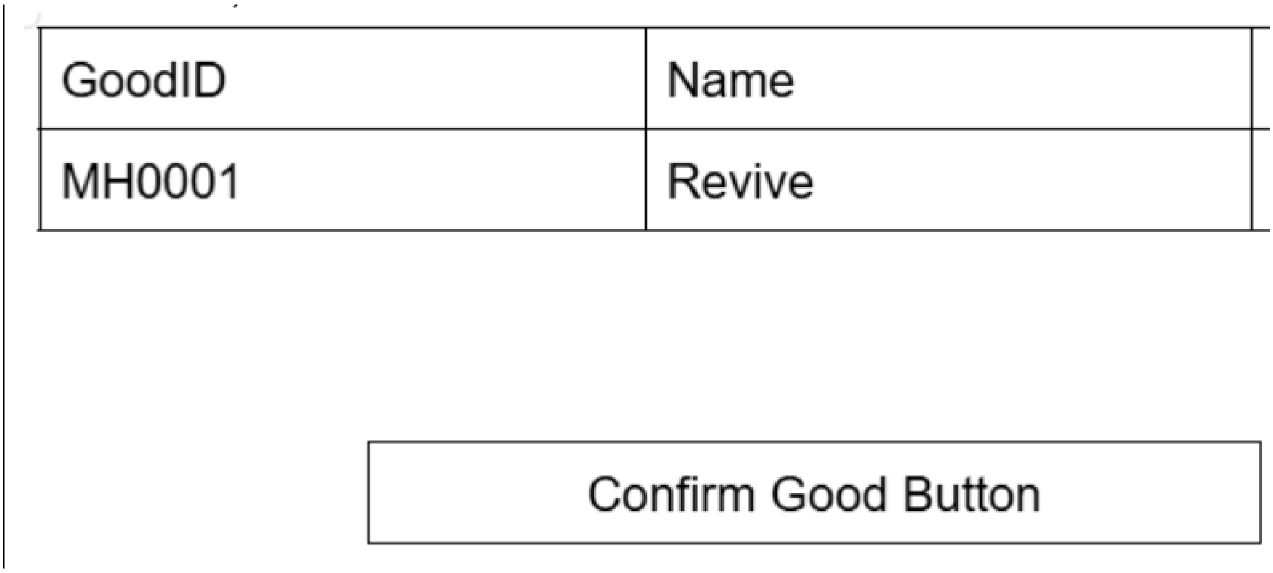
16. The Search Goods UI appears with: an input field to search providers by name, a search button, a
an add provider button.

17. The yard manager enters the keyword “Revive” in the search bar and clicks on the Search Button.

18. The system displays the Good Information with a table and a confirm button(A Confirm Goods button).

19. The yard manager enters the keyword “Revive” in the search bar and clicks Search.

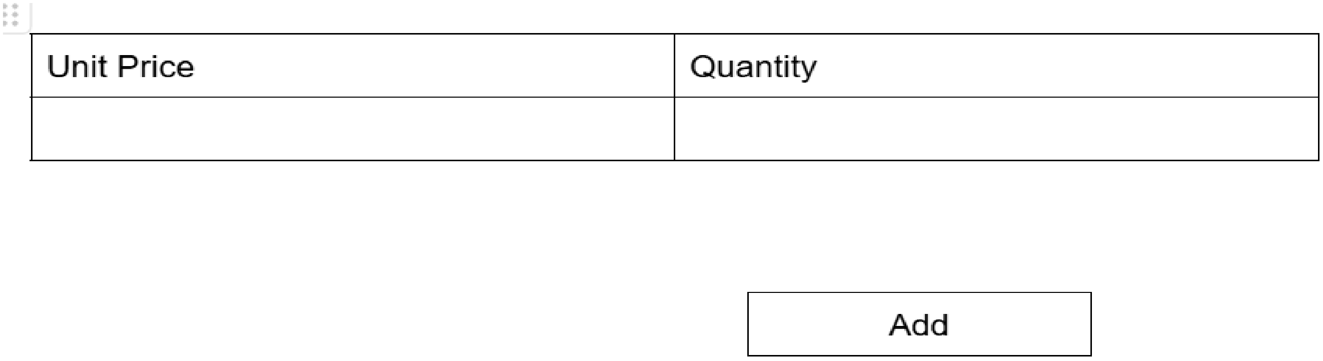
20. The system displays the good information with a table and a confirm button(A Confirm Goods butto



GoodID	Name
MH0001	Revive

21. The yard manager clicks on the first line, which corresponds to the keyword “Revive” to see the attribute and the value of the selected goods: GoodId: MH0001, name: Revive.

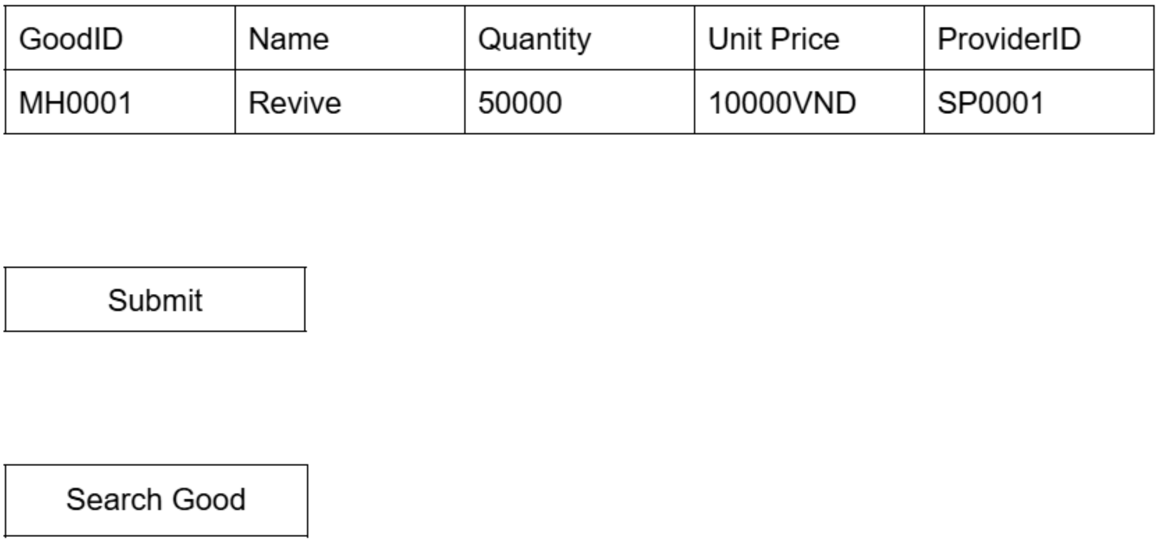
22. After that, the Add to Invoice UI appears, including a table and an add button:



Unit Price	Quantity

23. The Yard Manager enters the unit price = 10000VND, quantity = 50000 and clicks the add Button.

24. The Add to Invoice UI after confirmed the information about unit price and quantity will be change to Invoice UI with a table,search good button and a submit button.



GoodID	Name	Quantity	Unit Price	ProviderID
MH0001	Revive	50000	10000VND	SP0001

25. The Yard Manager will check the information in this table, and if all the imported goods are added t the invoice, they will click the submit button. Else, they have to click to Search Good to return to the Search Good UI to do every above again.

26. After clicked on submit button, the system will print the imported invoice to the provider and return the yard manager UI.

Exception	27. If the system does not find the goods, it displays an alert: "**The good is not found in the system**!" 27.1. The yard manager clicks on the “Add Goods” 27.2. The system displays the "Add Good UI". 27.3. The yard manager has to enter GoodID = MH0003, name = Coca. 27.4 After that, the yard manager clicks on the confirm supplier button to ensure that the information about the goods is added to the system. 27.5. The system announces a success alert and then returns to the search good UI. 28. If the system does not find the supplier, it displays an alert: "**The provider is not found in the system!"** 28.1.The yard manager clicks on the "Add Provider". 28.2.The system displays the "Add Provider UI". 28.3.The yard manager has to enter providerID = MH0003, name = CocaCola, address = “Đồng Nai”, email = cocacola@gmail.com, phone = 0988223122, description = providing product for everyone 28.4. After that the yard manager clicks on the confirm provider button to ensure the information about providers is added to the system. 28.5. The system displays a success alert, confirming that the providers have been successfully impor and then returns to the search provider UI.

2. Entity class extraction

- Step 1: Describe the system in a paragraph or a scenario + exceptions

- **Step 2+3**: Extract and evaluate nouns in Step 1

The yard manager: this is a kind of member in the system(user) -> a class: user

Sale: out of the scope of the system -> **rejected**

Goods: need to be managed -> a class: goods.

Providers: need to be managed -> a class: providers

Invoices: need to be managed -> a class: invoices

So we obtain the initial classes: User, goods, providers, invoices.

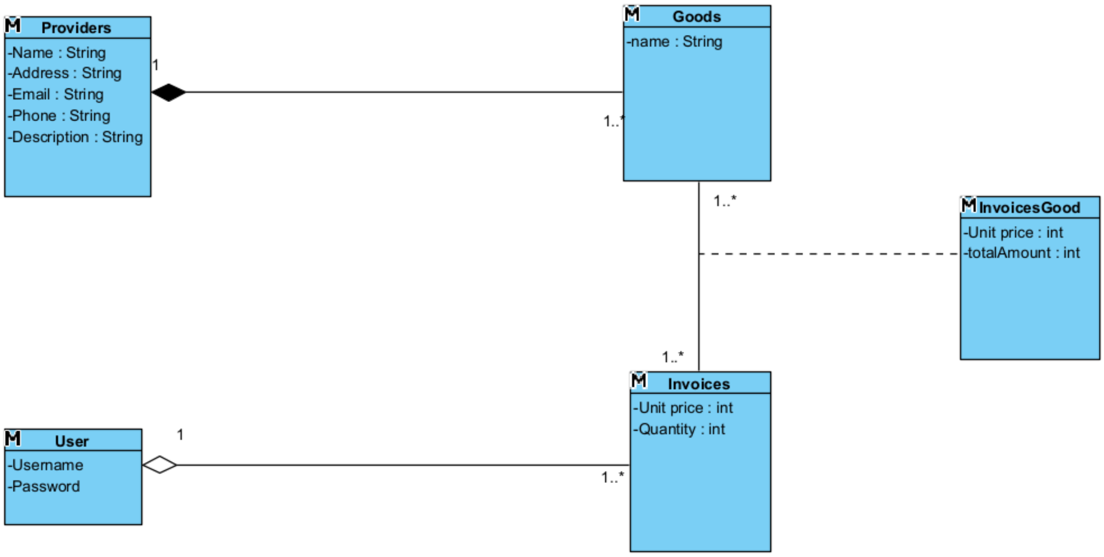
- Quantity and object relationships among classes.

A good can be provided by many providers, a provider can provide many goods. So good - provider: n-n.

A user can manage many provider invoices, a provider invoice can be managed by a user. So, user-invoices: 1-n

A good can be contained by many invoices. An invoice can contain many goods. So: good - invoices: n-n

A good can belong to many invoices, an invoice can contain many rooms -> We could purpose a class between them: Invoices Good.



3. Full Class Diagram.

Enter the system, the login interface appears -> need a class -> LoginView

- Input: Username -> inUserName
- Input: Password -> inPassword
- A submit login button -> 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
- owner class: User

Once the login is successful, the main interface of the yard manager appears -> need a class: YardManagerHomeView which has at least.

An option to manage importing goods: subImportManage()

Select the importing option, the importing interface appears -> need a class: YardImportingView

- Options
    - Search: subSearch
    - Submit: subSubmitInvoice

Click on search, the search interface appears -> need a class: searchView

- Options to search:
    - Search Provider: subSearchProviders
    - Search Good: subSearchGoods.

Click on search provider, the search provider interface appears -> need a class: SearchProviderView

- Input field to search providers by name -> inProviderName
- Search Button: subSearchProvider
- Add Provider Button: subAddProvider

Enter provider's name and click search -> The system searches for providers by name -> need a method:

- Name: searchProvider()
- Input: provider name
- Output: list of providers
- Owner class: provider

The system displays a provider list with a confirm button -> need an attribute: outProviderList: List<Provider>

Click on a provider name -> The UI for goods import appears -> need a class: ImportGoodView

- Provider detail: providerID, name, address, email, phone, description.
- Confirm Button: subConfirmProvider

Click the confirm button -> The system completes and return to searchView

Clicks on Search Goods, the search good interface appears, need a class: searchGoodView

- Input field to search good by name: inGoodName
- Search Button: subSearchGoods
- Add Goods Button: subAddGoods

Enter a good name and click search -> The system searches for goods by name- need a method:

- Name: searchGoods()
- Input: goods name
- Output: list of goods
- Owner class: Good.

The system displays a list of goods with a confirm button -> need a class attribute: outGoodList: List<Goods>

Click on a goods item, the add to invoice interface appears -> Need a class: InvoiceView:

- Input of unit price: inUnitPrice
- input for quantity: inQuantity
- Confirm button: subConfirmInvoice

Enter unit price and quantity, then click confirm, the system update the invoices and displays a table with a submit button, need a method:

- Name: addToInvoice()
- Input: goodID, unit price, quantity
- Add Button: subAdd
- Output: boolean
- Owner class: Invoice

Click the add button, the system display the Invoice UI for yard manager to see finally before print to provider -> need a class: InvoiceView:

- Output: outInvoice
- A button to submit: subSubmit

The yard manager click to submit button to print the invoice to provider and returns to YardManagerHomeView

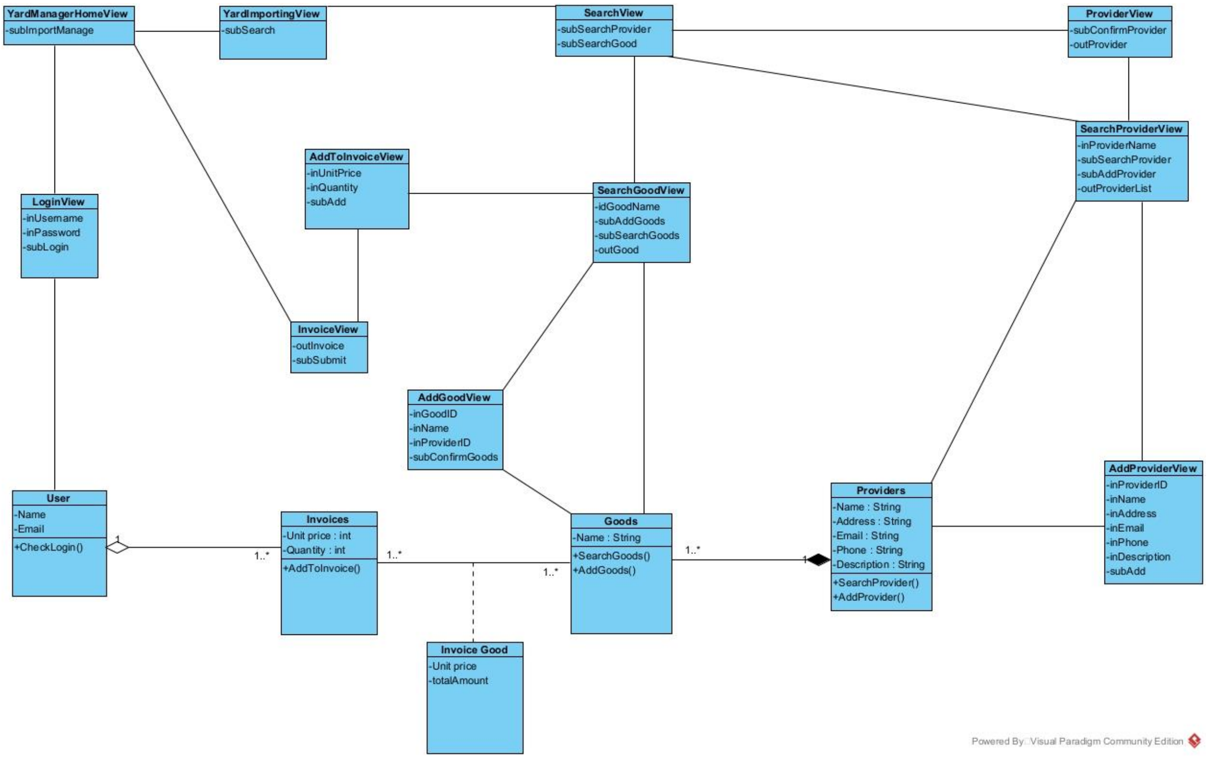
If goods are not found, the system shows an alert and provides an option to add new goods -> need a class: AddGoodsView

- Input: GoodID, name, ProviderID

- Confirm Button: subConfirmGoods
- Output: boolean
- Owner class: Good

if providers are not found, the system shows an alert and provides an option to add new provider -> need a class: AddProviderView:

- Name: AddProvider()
- Input: providerID, name, address, email, phone, description.
- Output: boolean
- Owner class: Provider



4. Scenario version 2

1. The yard manager enters the username/password and clicks the Login Button.
2. The LoginView calls the class User to process.
3. The class Users calls the method CheckLogin. The login is successful.
4. The class Users returns the results to the LoginView
5. The class LoginView calls the class YardManagerHomeView.
6. The class YardManagerHomeView displays itself to the yard manager.
7. The yard manager chose to manage imports.

8. The YardManagerHomeView calls the class YardImportingView.

9. The YardImportingView displays itself to the yard manager.

10. The yard manager clicks on the search button.

11. The YardImportingView calls the class SearchView.

12. The SearchView displays itself to the yard manager.

13. The yard manager clicks on the search provider button.

14. The SearchView calls the class SearchProviderView.

15. The class SearchProviderView displays itself to the yard manager.

16. The yard manager enters a keyword and clicks on the search button.

17. The SearchProviderView calls the class Provider to process.

18. The class Provider calls the method SearchProvider.

19. The class Provider returns the result to the SearchProviderView.

20. The SearchProviderView returns the result to the ProviderView.

21. The ProviderView displays itself to the yard manager.

22. The yard manager reviews the information and clicks the confirm provider button.

23. The ProviderView returns to the class SearchView.

24. The SearchView displays itself to the yard manager.

25. The Yard Manager clicks on the search good button.

26. The SearchView calls the class SearchGoodView.

27. The SearchGoodView displays itself to the yard manager.

28. The yard manager enters a keyword and clicks on the search button.

29. The SearchGoodView calls the class Good to process.

30. The class Good calls the method SearchGood.

31. The class Good returns the result to the SearchGoodView.

32. The SearchGoodView displays itself to the yard manager.

33. The yard manager views the information and clicks on the row of the information.

34. The SearchGoodView calls the class AddToInvoiceView.

35. The AddToInvoiceView displays itself to the yard manager.

36. The yard manager enters the unit price and quantity, and then confirms the add button.

37. The AddToInvoiceView calls to the class Invoice.

38. The Invoice calls to the method add to invoice.

39. The Invoice returns the result to the InvoiceView

40. The class InvoiceView displays itself to the yard manager.

41. The yard manager views the information, and they can click the submit button if they have imported all the goods.

42. The invoices after submitted are print to provider.

43. The Invoices display a success status to the InvoiceView.

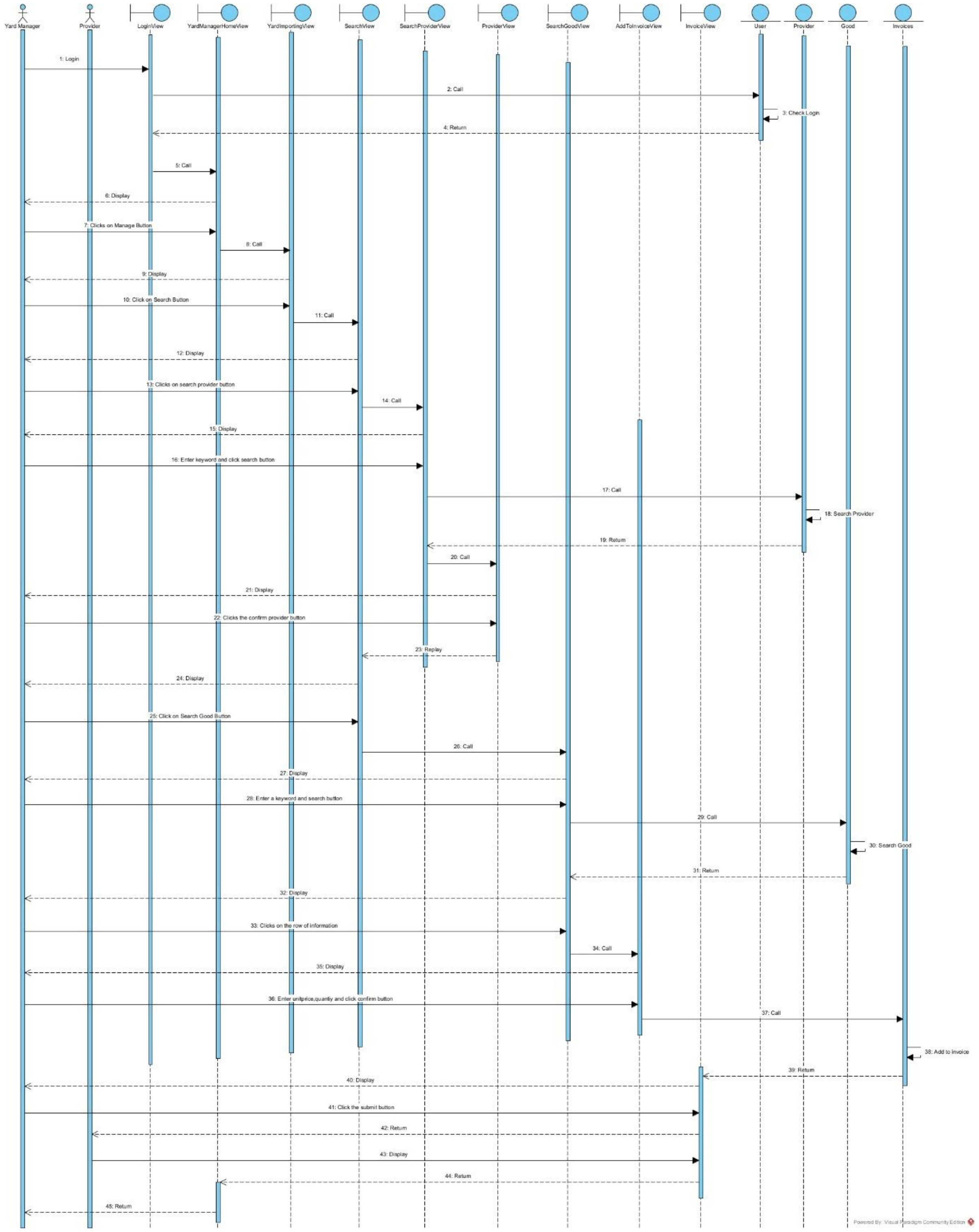
44. The InvoiceView returns to the YardManagerHomeView.

45. The YardManagerHomeView displays itself to the yard manager.

Exception:

3. The User use checkLogin() to check the Login. If the Login false:

- 3.1: User returns result to the LoginView.
- 3.2: LoginView shows the alert “Wrong Login, please Login again”.
- 3.3: Yard Manager enters username and password again and clicks on the Login Button.
- 3.4: User use checkLogin() to check the Login again. The Login successful in step 4.



4. Entity class Design

Input: The analysis entity class diagram. We need to process in the following steps:

Step 1: Add the id attribute for the classes that do not inherit from other classes: Providers, Users, Invoices, Goods.

Step 2: Add the type of each attribute in all classes.

User class:

id: String

Name: String

Email: String

Provider class:

id: String

Name: String

Address: String

Phone: String

Description: String

Goods class:

id: String

Name: String

Invoices class:

id: String

Unitprice: int

Quantity: int

Invoices Good class:

id: String

Unit Price: int

totalAmount: int

Step 3: Convert all association relationships to correspond to aggregation/composition relationships

Goods + Invoices -> Invoices Goods is converted to: Good is the component of Invoices Goods, Invoices Goods is the component of Invoices.

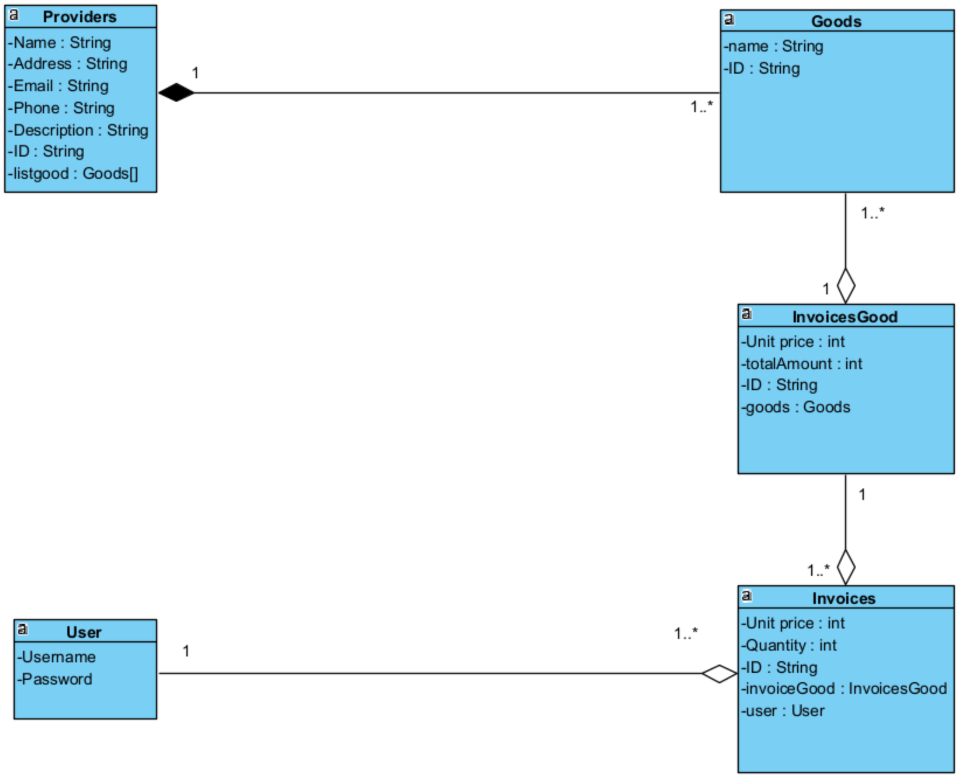
Step 4: Add the object attributes that correspond to the aggregation/composition relationships

Goods is the component of the provider, of type n-1 -> provider has a list of goods

Goods is the component of the Invoice Goods, of type: n-1 -> Invoice Goods has a list of goods.

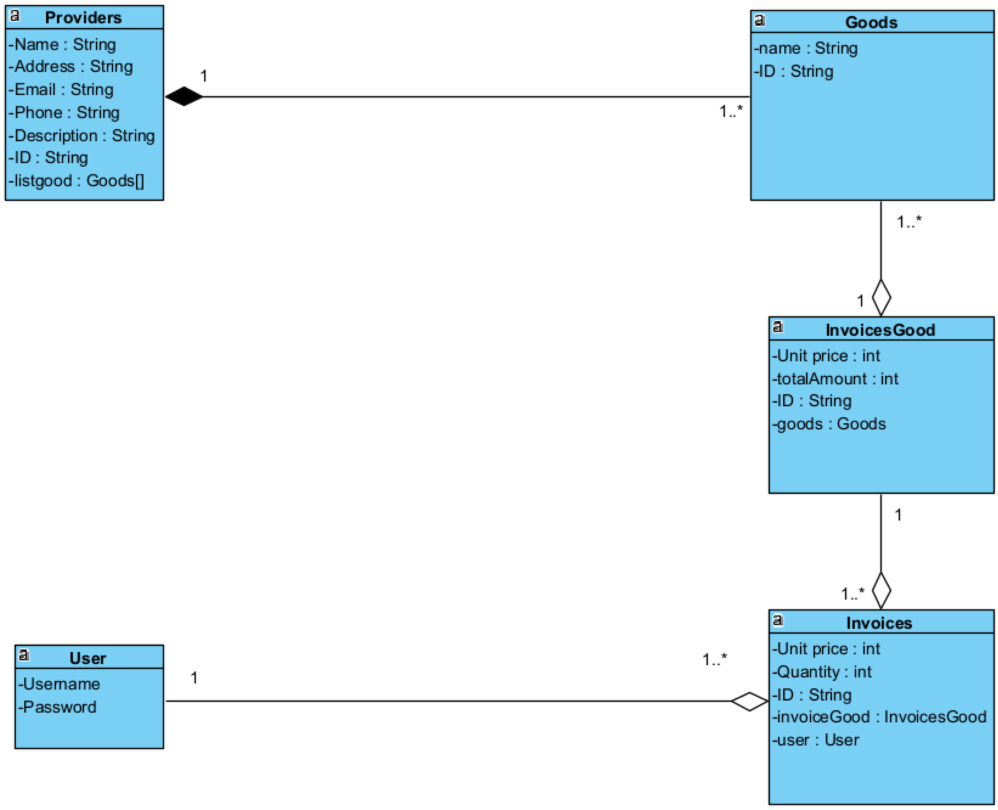
Invoices Good is the component of the Invoices, of type: 1-n -> Invoice has a Invoices Good.

Invoices is the User component, of type n -1: User has a list of Invoices.



5. Database design

Input is the entity class diagram from the design phase



Step 1: For each entity class, create a corresponding table.

Providers -> tblProviders

Users -> tblUser

Invoices -> tblInvoices

Invoices Good -> tblInvoicesGood

Goods -> tblGoods

Step 2: For each entity class, transfer all non-object attributes to contribute as the columns of the corresponding table.

tblProviders:

id: varchar(50)

name: varchar(50)

address: varchar(50)

email: varchar(50)

phone: varchar(50)

Description: varchar(50)

tblGoods:

id: varchar(50)

name: varchar(50)

tblUser:

id: varchar(50)

username: varchar(50)

password: varchar(50)

tblInvoices:

id: varchar(50)

unit price: integer(10)

quantity: integer(10)

tblInvoiceGood:

id: varchar(50)

unit price: integer(10)

totalAmount: integer(10)

Step 3: Consider the quantity relationships among entity classes. These relationships will be those among corresponding tables

1 tbProviders - n tbGoods

n tbInvoices - n tbGoods

1 tbUser - n tbInvoices

Step 4: Configure the key column for tables.

tblUser:

- id: primary key

- The relationship between tbUser and tbInvoices is 1-n -> tbInvoices will save the foreign key of UserID references to tbUser.

tblGoods:

- id: primary key
- The relationship between tbProviders and tbGoods is: 1-n -> tbGoods will save the foreign key providerID reference to tbProviders
- The relationship between tbGoods and tbInvoicesGoods is 1-n -> tbInvoicesGoods will save the foreign key GoodsID reference to tbGoods.
-

tblInvoices:

- id: primary key
- The relationship between tbInvoices and tbInvoicesGoods is 1-n -> tbInvoiceGoods will save the foreign key of InvoicesGoods reference to tbInvoicesGoods.

tblInvoicesGoods:

- Invoicesid: primary key.

tblProviders:

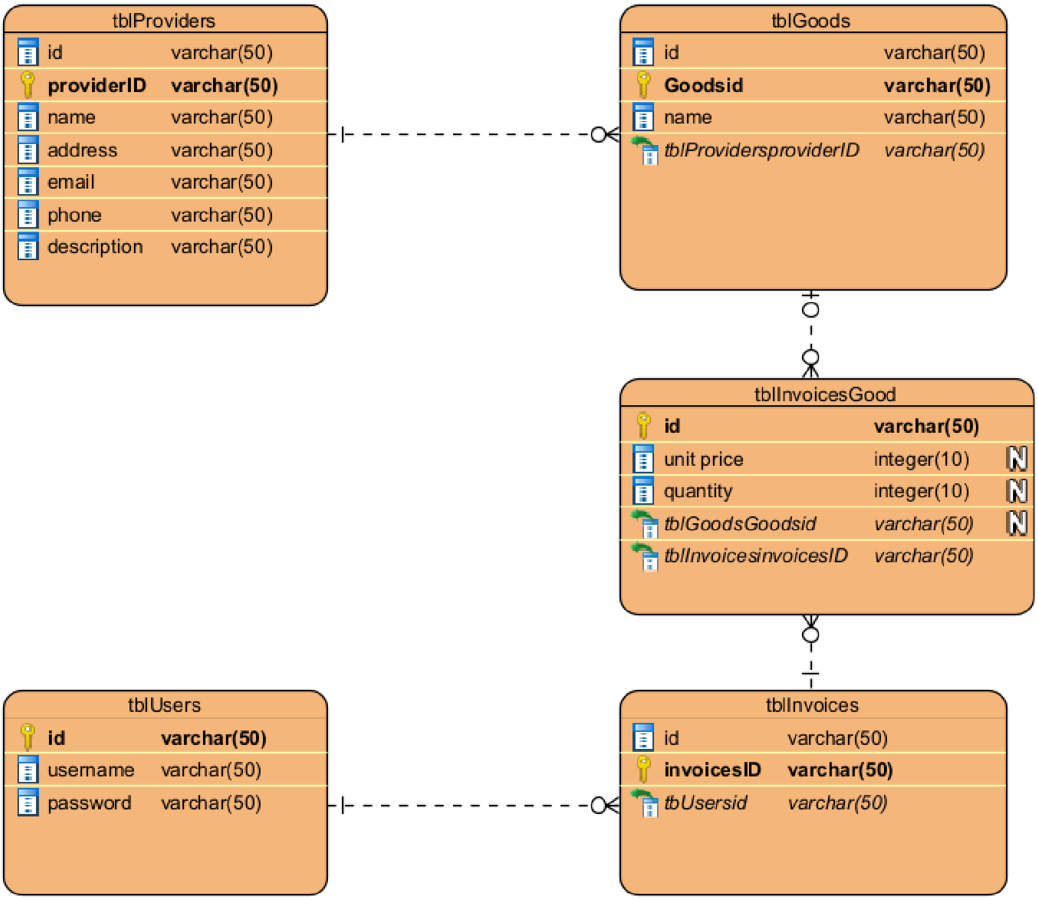
- providerID: primary key.
- The relationship between tbProviders and tbGoods í 1-n -> tbGoods will save the foreign key of ProviderID reference to tbProviders

Step 5: Delete the derived attribute and abundant attribute

Abundant attribute: Price in tbInvoices.

Derive Attribute: totalAmount in tbInvoicesGood.

The database for the system:



6. Classes diagram(design)

Interface Design

In this module, the interface includes 7 main interface:

+Login Page.

+ YardManagerHome Page

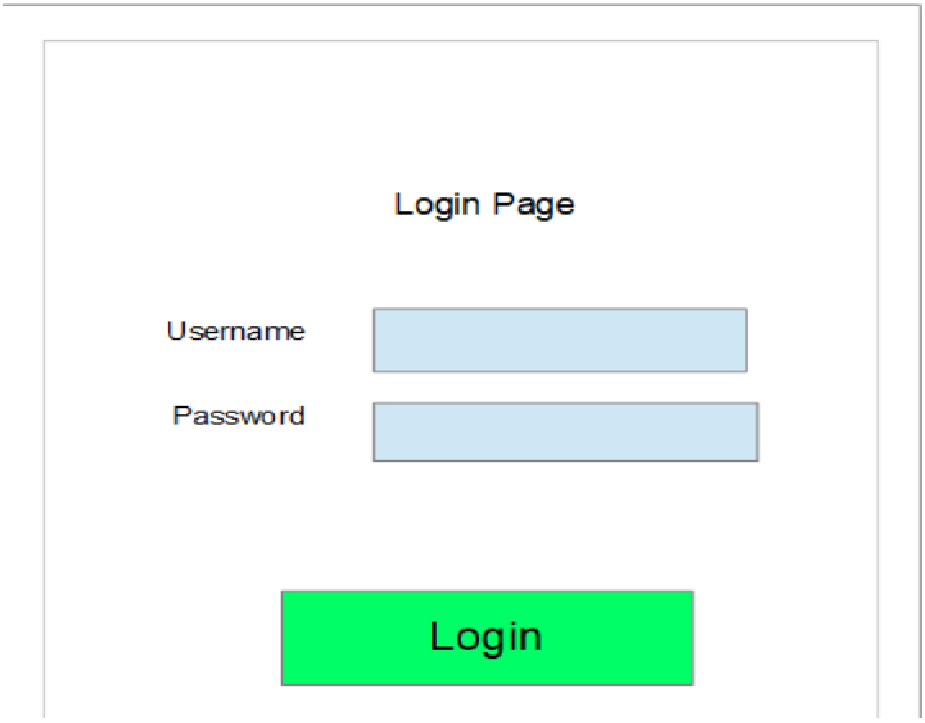
+ YardManagerImporting Page

+ Search Page

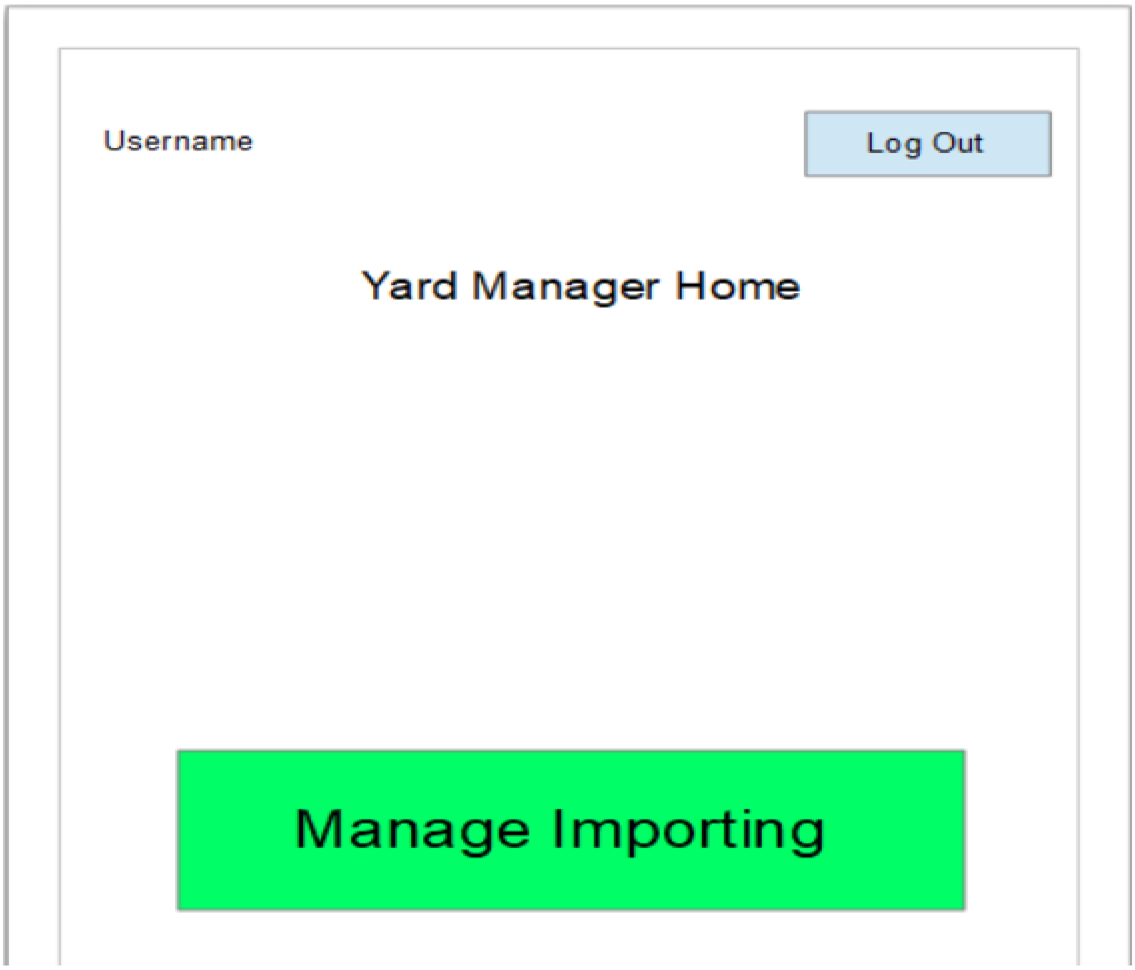
+ SearchProvider Page

+ SearchGood Page

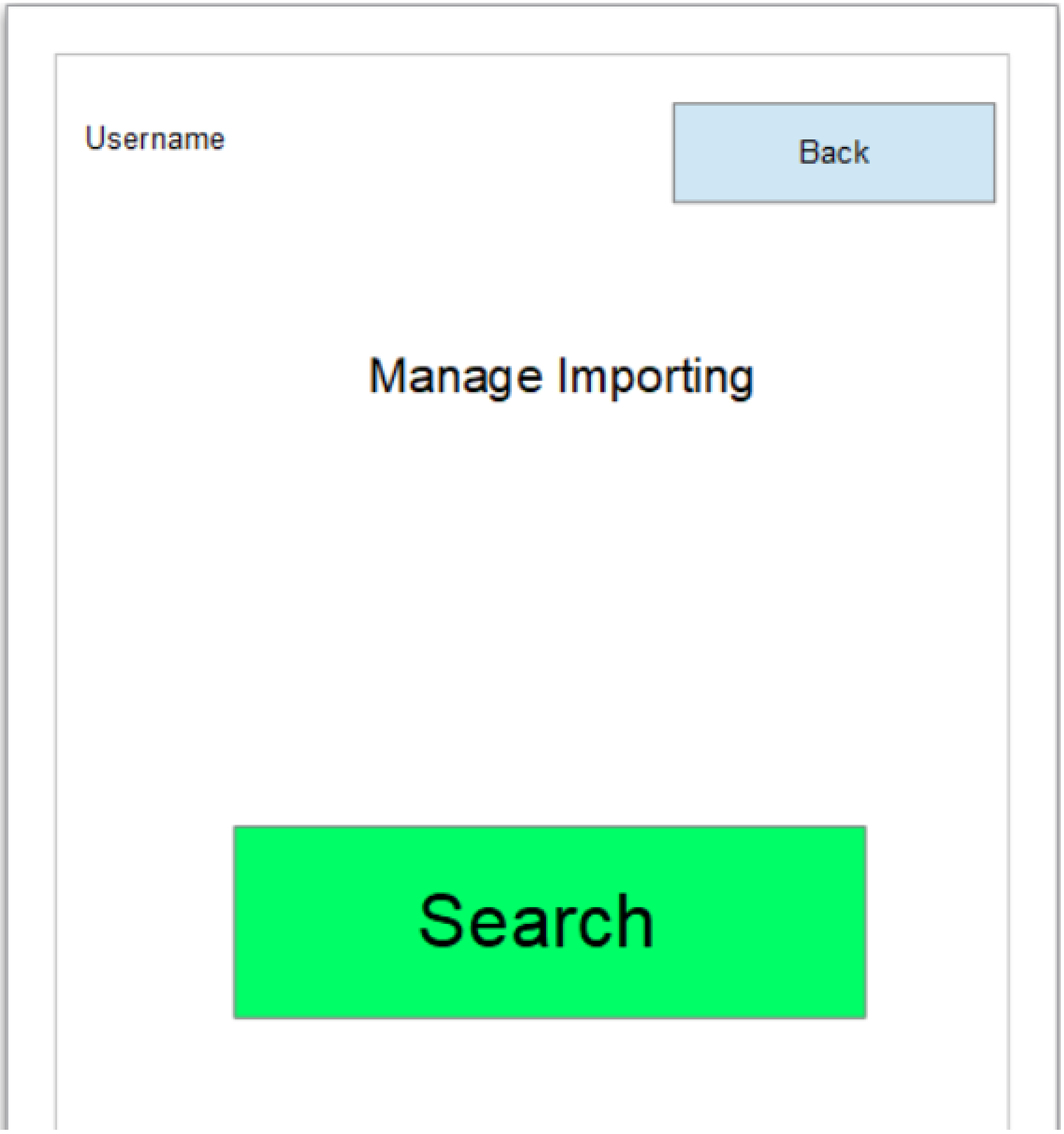
+ AddToInvoice Page.



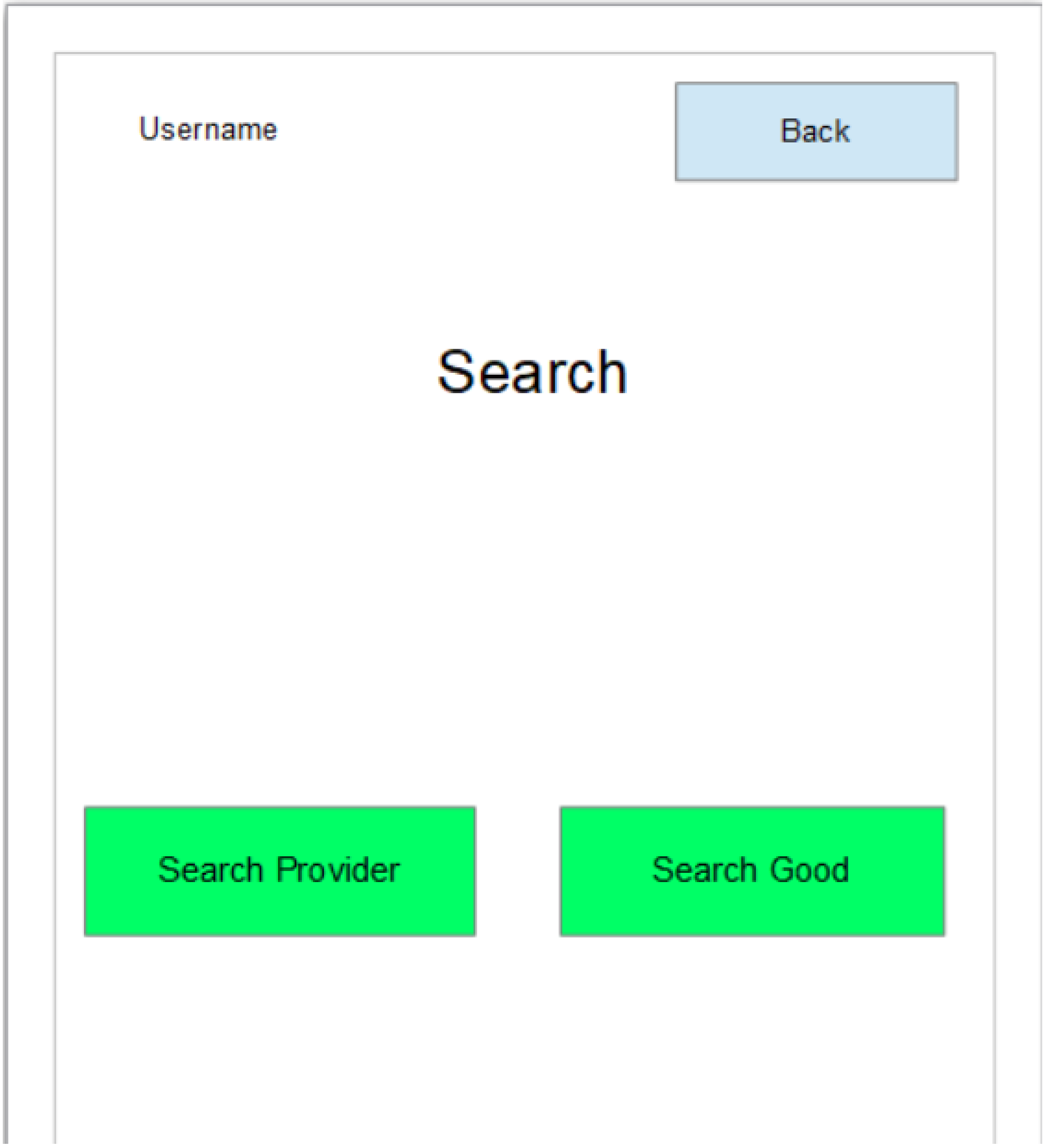
Login Page



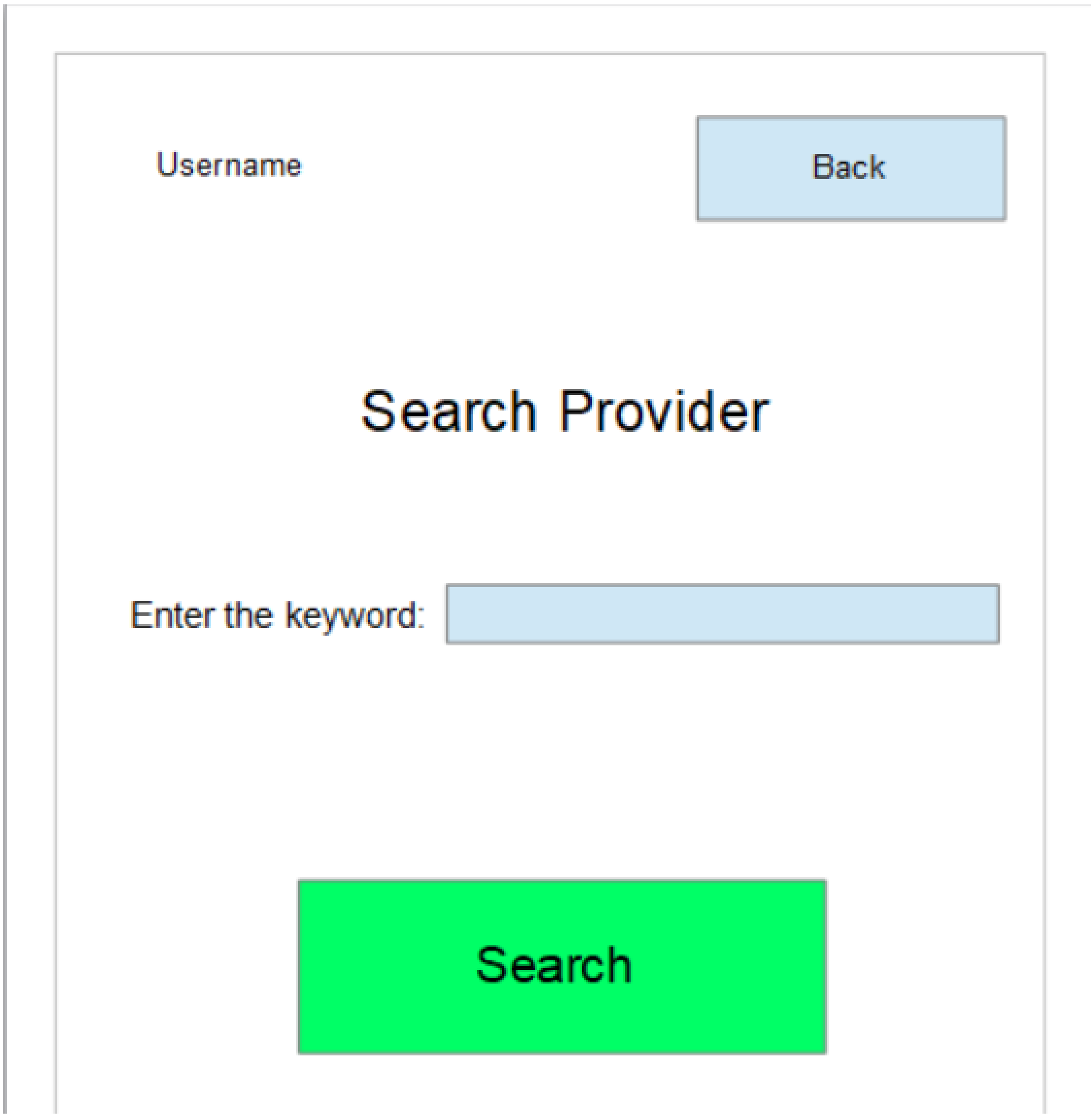
YardManagerHome Page



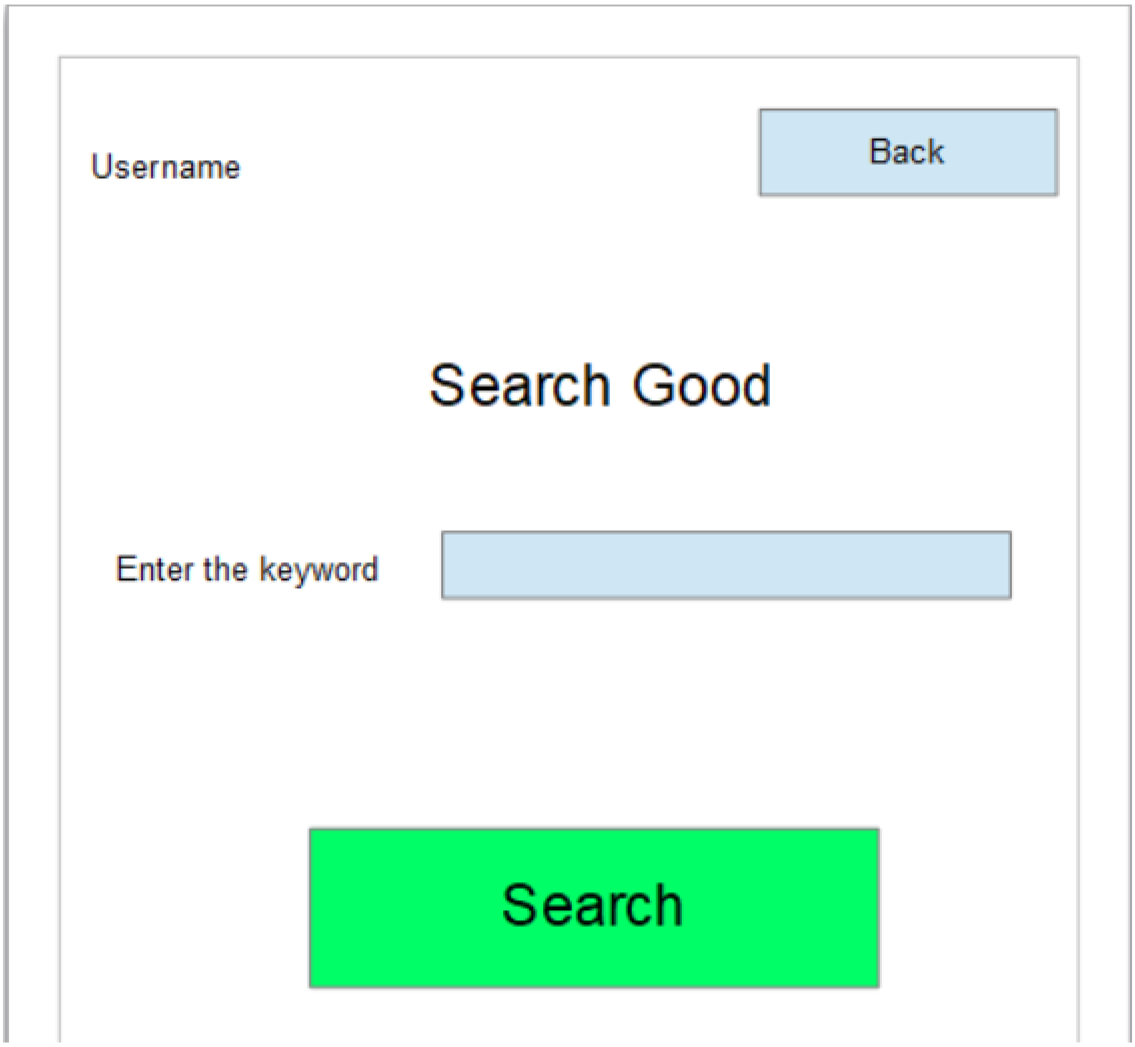
YardManagerImporting Page



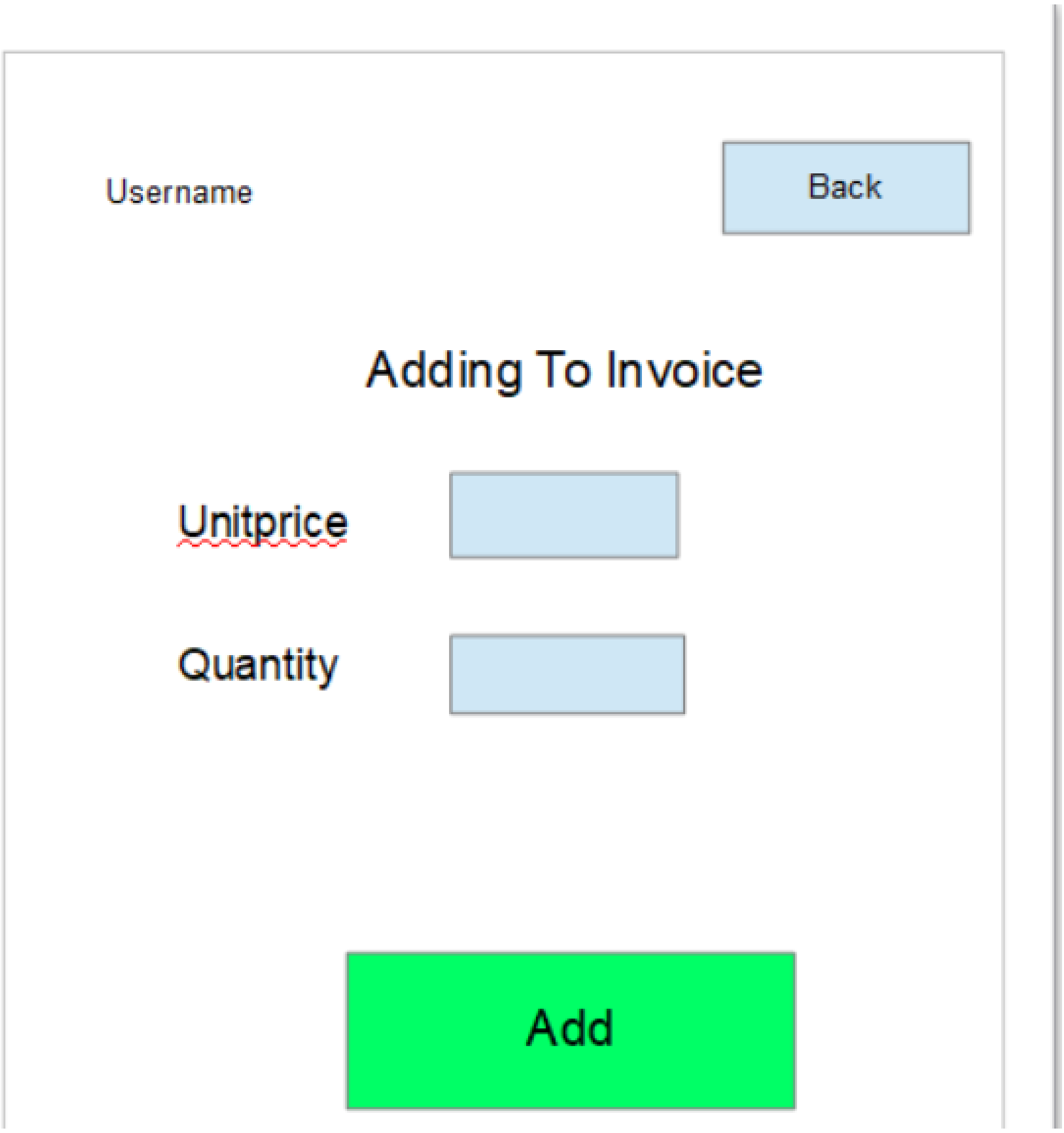
Search Page



SearchProvider Page



SearchGood Page



AddToInvoice Page

In this module, the login processing is omitted.

View class

- LoginFrm is the interface to log in. It needs a text field to enter the username, a text field to enter a password, and a button to log in.
- YardManagerHomeFrm is the home interface of the yard manager. It needs at least a button to go to the room manager importing function and a button to logout.
- YardImportingFrm is the interface of importing. It includes a button: search to go to the search function.
- SearchFrm is the interface of searching. It includes 2 button: search provider to go to the search provider function and search good to go to the search good.

- SearchProviderFrm is the interface of provider searching. It includes a text field to enter the keyword to search provider by name, a button to search, a button to add and a table to show the list of found provider.

- SearchGoodFrm is the interface of good searching. It includes a text field to enter the keyword to search good by name, a button to search, a button to add and a table to show the list of found good.

- AddProviderFrm is the interface of provider adding. It needs the input text field to enter the provider information: providerID, name, address, email, phone, description and a button to confirm the provider's information.

- AddGoodFrm is the interface of good adding. It needs the input text field to enter the good information: GoodID, name and a button to confirm the information of good.

- AddToInvoiceFrm is the interface of invoice adding. It needs the input text field to enter the invoice information: unit price and quantity a button to add and the table to display all the information of the invoices.

Control(DAO) class:

- DAO is a general class of DAO. It has only the construction to connect to the DB and provides the common connection for all inherited DAO classes in the system.

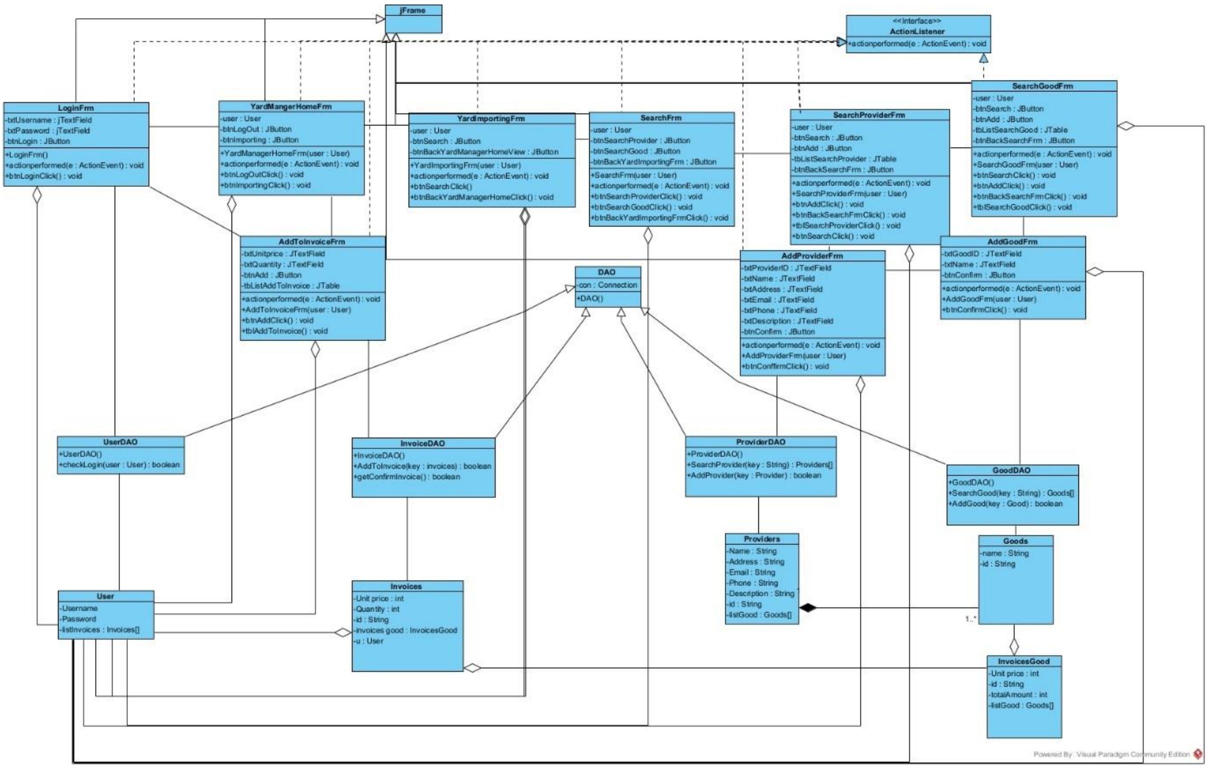
- UserDAO is the class for manipulating with DB related to the User object. In this module, it needs a method to verify whether the login information is correct or not, it is checkLogin() method.

- ProviderDAO is the class for manipulating with DB related to the Provider object. In this module, it’s need 2 method: searchProvider(): to search all providers whose name contains the keyword and AddProviders(): to add the provider which does not exist in the system.

- GoodDAO: is the class for mantipulating with DB related to the Good object. In this module, it’s need 2 method: searchGood(): to search all good which name contains the keyword and AddGoods(): to add all the good which does not exist in the system.

- InvoicesDAO: is the class for manipulating with DB related to the Invoices Object. In this module, it’s need a method: addToInvoice: to add all information of the good to the invoice, confirmInvoice: to confirm the invoice for giving Provider.

Entity classes: User, provider, good, invoices, invoice good.


jFrame
<<Interface>>
ActionListener
+actionperformed(e : ActionEvent) : void
LoginFrm
-txtUsername : jTextField
-txtPassword : jTextField
-btnLogin : JButton
+LoginFrm()
+actionperformed(e : ActionEvent) : void
+btnLoginClick() : void
YardMangerHomeFrm
-user : User
-btnLogOut : JButton
-btnImporting : JButton
+YardManagerHomeFrm(user : User)
+actionperformed(e : ActionEvent) : void
+btnLogOutClick() : void
+btnImportingClick() : void
YardImportingFrm
-user : User
-btnSearch : JButton
-btnBackYardManagerHomeView : JButton
+YardImportingFrm(user : User)
+actionperformed(e : ActionEvent) : void
+btnSearchClick()
+btnBackYardManagerHomeClick() : void
SearchFrm
-user : User
-btnSearchProvider : JButton
-btnSearchGood : JButton
-btnBackYardImportingFrm : JButton
+SearchFrm(user : User)
+actionperformed(e : ActionEvent) : void
+btnSearchProviderClick() : void
+btnSearchGoodClick() : void
+btnBackYardImportingFrmClick() : void
SearchProviderFrm
-user : User
-btnSearch : JButton
-btnAdd : JButton
-tbListSearchProvider : JTable
-btnBackSearchFrm : JButton
+actionperformed(e : ActionEvent) : void
+SearchProviderFrm(user : User)
+btnAddClick() : void
+btnBackSearchFrmClick() : void
+tblSearchProviderClick() : void
+btnSearchClick() : void
SearchGoodFrm
-user : User
-btnSearch : JButton
-btnAdd : JButton
-tbListSearchGood : JTable
-btnBackSearchFrm : JButton
+actionperformed(e : ActionEvent) : void
+SearchGoodFrm(user : User)
+btnSearchClick() : void
+btnAddClick() : void
+btnBackSearchFrmClick() : void
+tblSearchGoodClick() : void
AddToInvoiceFrm
-txtUnitprice : JTextField
-txtQuantity : JTextField
-btnAdd : JButton
-tbListAddToInvoice : JTable
+actionperformed(e : ActionEvent) : void
+AddToInvoiceFrm(user : User)
+btnAddClick() : void
+tblAddToInvoice() : void
DAO
-con : Connection
+DAO()
AddProviderFrm
-txtProviderID : JTextField
-txtName : JTextField
-txtAddress : JTextField
-txtEmail : JTextField
-txtPhone : JTextField
-txtDescription : JTextField
-btnConfirm : JButton
+actionperformed(e : ActionEvent) : void
+AddProviderFrm(user : User)
+btnConffirmClick() : void
AddGoodFrm
-txtGoodID : JTextField
-txtName : JTextField
-btnConfirm : JButton
+actionperformed(e : ActionEvent) : void
+AddGoodFrm(user : User)
+btnConfirmClick() : void
UserDAO
+UserDAO()
+checkLogin(user : User) : boolean
InvoiceDAO
+InvoiceDAO()
+AddToInvoice(key : invoices) : boolean
+getConfirmInvoice() : boolean
ProviderDAO
+ProviderDAO()
+SearchProvider(key : String) : Providers[]
+AddProvider(key : Provider) : boolean
GoodDAO
+GoodDAO()
+SearchGood(key : String) : Goods[]
+AddGood(key : Good) : boolean
Providers
-Name : String
-Address : String
-Email : String
-Phone : String
-Description : String
-id : String
-listGood : Goods[]
Goods
-name : String
-id : String
1..*
User
-Username
-Password
-listInvoices : Invoices[]
Invoices
-Unit price : int
-Quantity : int
-id : String
-invoices good : InvoicesGood
-u : User
InvoicesGood
-Unit price : int
-id : String
-totalAmount : int
-listGood : Goods[]
Powered By : Visual Paradigm Community Edition

7. Sequence Diagram(Dynamic Design)

Scenario version 3:

1. A yard manager enters his username, password, and clicks on the login button on LoginFrm.

2. The method actionPerformed() of LoginFrm is called.

3. The method actionPerformed() calls User to create a User object.

4. The class User packs the information into a User object.

5. The class User returns a User object to the method actionPerformed().

6. The method actionPerformed() calls the method checkLogin() of the class UserDAO.

7. The method checkLogin() checks login information.

8. The method checkLogin() calls the class User to set 3 attributes: id, name, and role.

9. The class User calls its method setID(), setName(), setRole().

10. The class User returns the User object to the method checkLogin().

11. The method checkLogin() returns the results to the actionPerformed().

12. The method actionPerformed() calls the class YardManagerHomeFrm.

13. The constructor YardManagerHomeFrm() is called.

14. The YardManagerHomeFrm interface is displayed to the yard manager.

15. The yard staff clicks on the "Manage Importing" button

16. The method actionPerformed() is called.

17. The method actionPerformed() calls the class YardImportingFrm.

18. The constructor YardImportingFrm() is called.

19. The interface YardImportingForm is shown to the yard manager.

20. The user clicks on the "Search" button.

21. The method actionPerformed() is called.

22. The method actionPerformed() calls SearchFrm().

23. The constructor SearchFrm() is called.

24. The interface SearchFrm is shown to the yard manager.

25. The yard manager clicks the “Search Provider” button.

26. The method actionPerformed() is called.

27. The method actionPerformed() calls SearchProviderFrm()

28. The constructor SearchProviderFrm() is called.

29. The interface SearchProviderFrm is shown to the yard manager.

30. The yard manager enters a keyword and clicks the “Search” button.

31. The method actionPerformed() is called.

32. The method actionPerformed() calls the method searchProvider() of the class ProviderDAO.

33. The method searchProvider() searches provider by name.

34. The method searchProvider() calls the class Provider to pack the results.

35. The class Provider packs each Provider object.

36. The class Provider returns the Provider object to the method searchProvider().

37. The method searchProvider() returns the results to the method actionPerformed() of the class SearchProviderFrm.

38. The method actionPerformed() displays the results to the yard manager.

39. The yard manager clicks on “Search Good” button to find good.

40. The method actionPerformed() is called.

41. The method actionPerformed() calls SearchGoodFrm().

42. The constructor SearchGoodFrm() is called.

43. The interface SearchGoodFrm is shown to the yard manager.

44. The yard manager enters the keyword and clicks search button.

45. The actionPerformed() is called.

46. The method actionPerformed() calls the method searchGood() of the class GoodDAO.

47. The method searchGood searches good by name.

48. The method searchGood() calls the class Good to pack the results.

49. The class Good packs each Good object.

50. The class Good calls the class Invoice to pack the results

51. The class Invoices pack each Invoice object.

52. The class Invoices return the result to the Good Class.

53. The Good class return the results to the AddToInvoiceFrm()

54. The class AddToInvoiceFrm displays itself to the yard manager.

55. The yard manager checks again if all the imported good are belongs to a provider, they click on confirm button.

56. The method actionPerformed() is called.

57. The class AddToInvoiceFrm calls the method ConfirmInvoices() of the class InvoiceDAO

58. The method ConfirmInvoices() are active with confirm button.

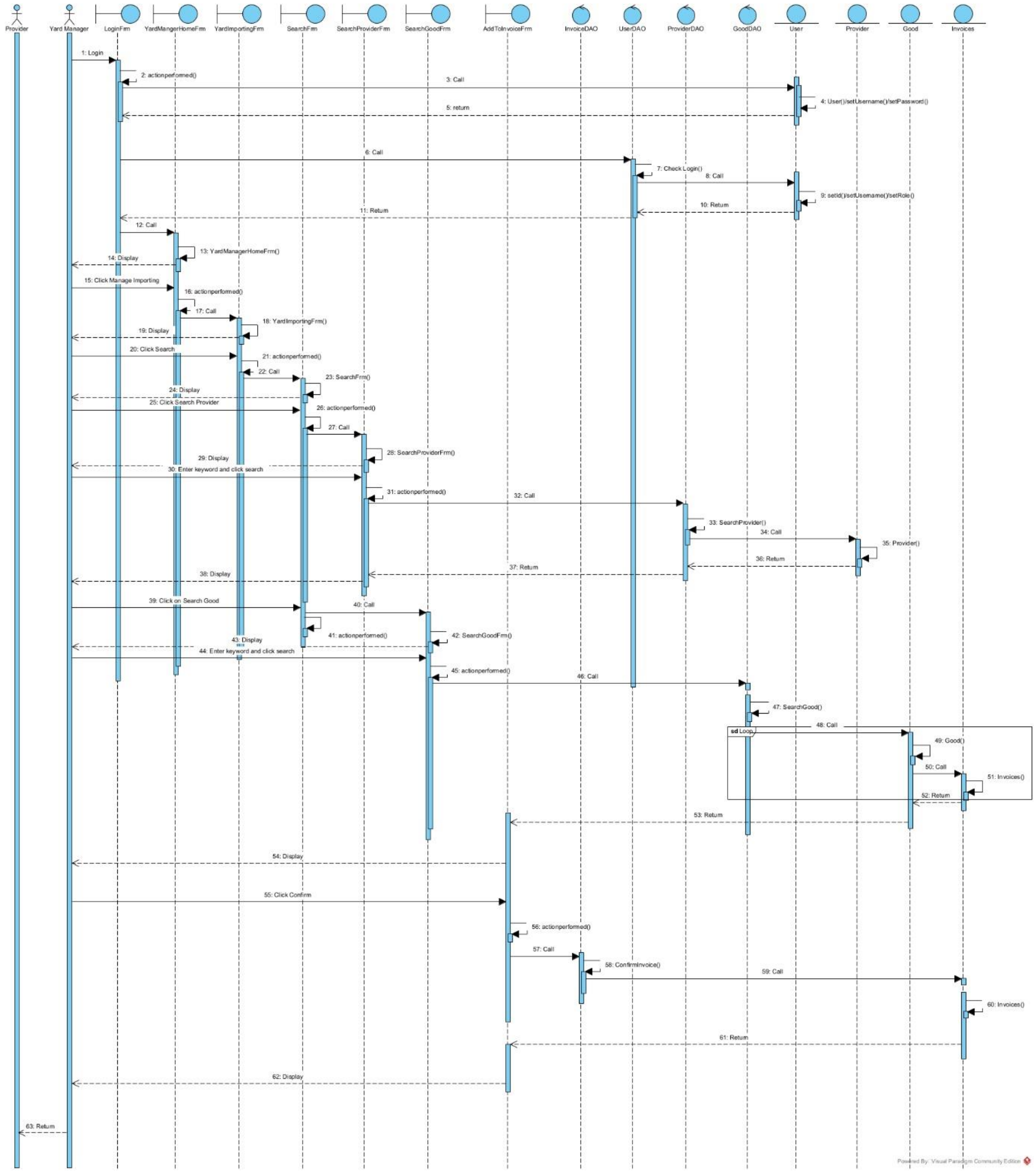
59. The method ConfirmInvoices() calls the class Invoice to pack the result.

60. The class Invoice pack each Invoice object.

61. The class Invoice return the results to AddToInvoiceFrm

62. The class AddToInvoiceFrm display itself to the Yard Manager.

63. The Yard Manager return the result to the Provider.



8. Black box-testing

Test plan

No	Module	Testcase

1	Goods Importing	Enter a valid invoice (provider and goods exist, valid quantity and unit price)
2	Goods Importing	Import goods with a non-existent provider
3	Goods Importing	Import goods with a negative unit price
4	Goods Importing	Import goods with a quantity equal to zero
5	Goods Importing	Import goods with missing information (missing goods ID or unit price)
6	Goods Importing	Import the same goods twice consecutively with the same unit price
7	Goods Importing	Import the same goods twice consecutively with different unit prices
8	Goods Importing	Import goods with a non-existent user (person performing the import)

Testcase No1

Database before testing

tblUser

id	**username**	**password**
1	manager	ymanager@

tblProvider

id	providerID	name	Address	Email	Phone	Description
1	P002	revive	Đồng Nai	revive@gmail.com	0122311231	Drinking
2	P002	acecook	Hải Phòng	acecook@gmail.com	0123451233	Eating
3	P003	cocacola	Đà Nẵng	cocacola@gmail.com	0123451243	Drinking
4	P004	a	Hà Nội	a@gmail.com	0981233246	Eating

tblGood

id	Goodsid	name	providerID
1	G001	Mì acecook	P002
2	G002	Nước Coca-Cola	P003
3	G003	Bánh gạo	P004
4	G004	Nước 7up	P001

5	G005	Nước Coca-Cola-Zero	P003

tblInvoices

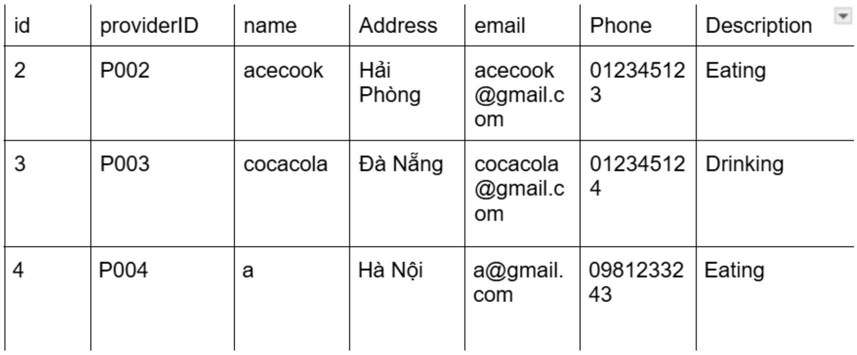
id	invoicesID	userID
1	IV001	1

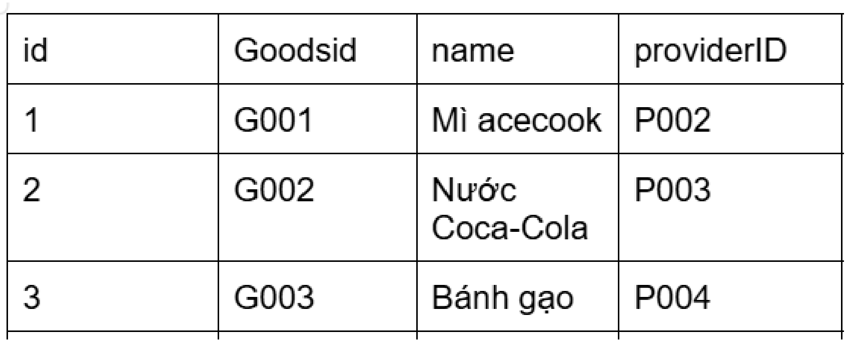
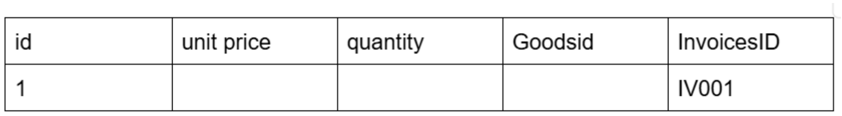
tbInvoicesGood

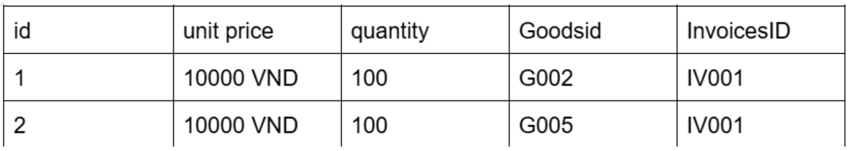
id	unit price	quantity	Goodsid	InvoicesID
1				IV001

Testing scenario and expected results

Scenario	Expected Results
1. Start the application	The login interface appeared with text fields for entering the username, password, and a login button
2. Enter username = manager password =	The home interface appears with a button: Manage Importing

<table>
<tr><td>ymanager@ and click on login button</td><td></td></tr>
<tr><td>3. Click on Manage Importing</td><td>The manage importing interface appears with a button: Search</td></tr>
<tr><td>4. Click on the search Button</td><td>The search interface appears with 2 buttons: Search provider and search good.</td></tr>
<tr><td>5. Click on the search provider button</td><td>The search provider interface has a text field to enter the keyword and a search button.</td></tr>
<tr><td>6. Enter keyword = “a” and click search button</td><td>There are three providers in the results and a back button

<table>
<tr><th>id</th><th>providerID</th><th>name</th><th>Address</th><th>email</th><th>Phone</th><th>Description</th></tr>
<tr><td>2</td><td>P002</td><td>acecook</td><td>Hải Phòng</td><td>acecook@gmail.com</td><td>012345123</td><td>Eating</td></tr>
<tr><td>3</td><td>P003</td><td>cocacola</td><td>Đà Nẵng</td><td>cocacola@gmail.com</td><td>012345124</td><td>Drinking</td></tr>
<tr><td>4</td><td>P004</td><td>a</td><td>Hà Nội</td><td>a@gmail.com</td><td>0981233243</td><td>Eating</td></tr>
</table>
Back</td></tr>
<tr><td>7. Click on the row has the name = ‘a’</td><td>The search provider interface appeared with the information of name = ‘a’:
id = 4
providerID = P004
name = a
address = “Hà Nội”
email = a@gmail.com</td></tr>
</table>

<table>
<tr><td></td><td>phone = 0981233243
Description = Eating and a back button</td></tr>
<tr><td>8. Click on back button</td><td>Return to the search interface</td></tr>
<tr><td>9. Click on search good button</td><td>The good interface appears with a textfield to enter the keyword and a search button.</td></tr>
<tr><td>10. Enter the keyword = “a” and click on search button</td><td>There are three goods in the results, an addToInvoice button and a back button:

<table>
<tr><th>id</th><th>Goodsid</th><th>name</th><th>providerID</th></tr>
<tr><td>1</td><td>G001</td><td>Mì acecook</td><td>P002</td></tr>
<tr><td>2</td><td>G002</td><td>Nước Coca-Cola</td><td>P003</td></tr>
<tr><td>3</td><td>G003</td><td>Bánh gạo</td><td>P004</td></tr>
</table>
<table><tr><td>Back</td><td>AddtoInvoice</td></tr></table></td></tr>
<tr><td>11. Click on AddtoInvoice button</td><td>The interface InvoiceGood appears with a table and a confirm button.

<table>
<tr><th>id</th><th>unit price</th><th>quantity</th><th>Goodsid</th><th>InvoicesID</th></tr>
<tr><td>1</td><td></td><td></td><td></td><td>IV001</td></tr>
</table>
<table><tr><td>Confirm</td></tr></table></td></tr>
</table>

<table>
<tr><td>12. Enter the unit price, quantity, and Goodsid belonging of all the goods belonging to a provider to the table.</td><td>The table will have all the information of the goods belonging to the provider:

<table><tr><th>id</th><th>unit price</th><th>quantity</th><th>Goodsid</th><th>InvoicesID</th></tr><tr><td>1</td><td>10000 VND</td><td>100</td><td>G002</td><td>IV001</td></tr><tr><td>2</td><td>10000 VND</td><td>100</td><td>G005</td><td>IV001</td></tr></table>
Confirm</td></tr>
<tr><td>13. Click on the confirm button</td><td>A message appeared with: “The Invoice is confirmed successfully”.</td></tr>
<tr><td>14. Click ok on the message</td><td>Return to the home interface of the yard manager.</td></tr>
</table>

Database after testing

Only change in the table of **InvoicesGood:**

id	unit price	quantity	Goodsid	InvoicesID
1	10000 VND	100	G002	IV001
2	10000 VND	100	G005	IV001