**Cover page**

**Mô tả phiên bản**

Mục lục

I. Giới Thiệu1

1.1 Mục đích2

1.2 Phạm vi2

1.3 Tổng quan2

1.4 Tài liệu tham khảo2

1.5 Định nghĩa và viết tắt2

II. Tổng quan hệ thống4

III. Kiến trúc hệ thống4

3.1 Bản thiết kế kiến trúc hệ thống (Architectural Design)5

3.2 Mô tả phân rã (Decomposition Description)5

3.3 Lý do thiết kế (Design Rationale)5

VI. Thiết kế dữ liệu4

4.1 Mô tả dữ liệu (Data Description)5

4.2 Từ điển dự liệu (Data Dictionary)5

V. Thiết kế bộ phận (Component Design)4

VI. Thiết kế giao diện tương tác4

6.1 Tổng quan giao diện tương tác5

6.2 Hình ảnh màn hình5

6.3 Đối tượng màn hình và hành động5

VII. Ma trận yêu cầu4

VIII. Phụ lục4

I. Giới Thiệu

1.1 Mục đích

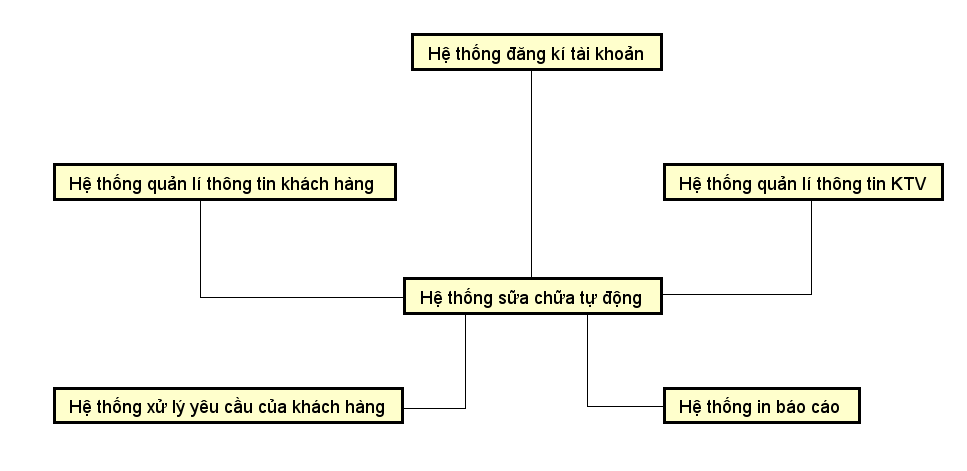
1.2 Phạm vi

1.3 Tổng quan

1.4 Tài liệu tham khảo

1.5 Định nghĩa và viết tắt

II. Tổng quan hệ thống

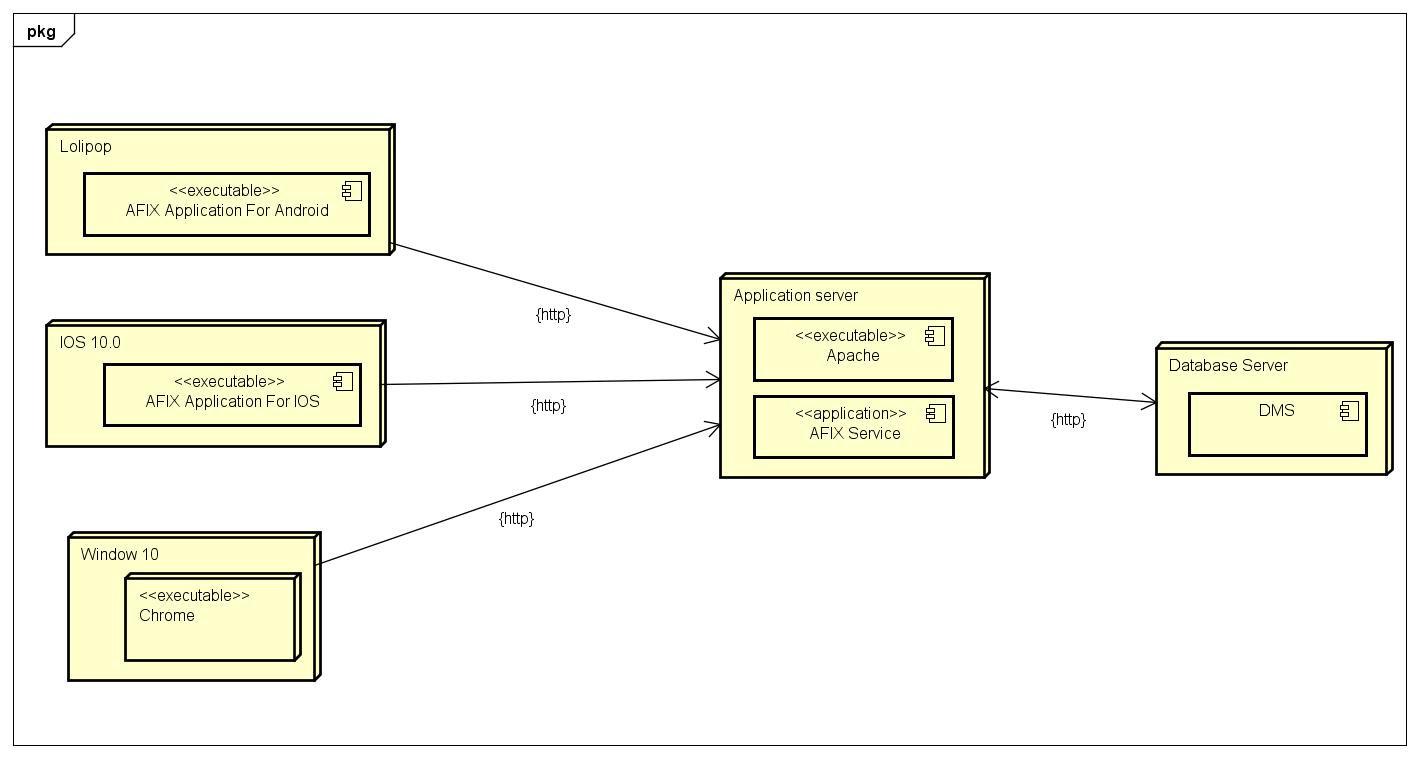


Hệ thống đăng kí tài khoản sẽ nhận yêu cầu đăng kí thông tin từ người dùng rồi từ đó sẽ đưa ra form đăng kí phù hợp. Sau khi dữ liệu của người dùng được lưu vào hệ thống quản lí thông tin khách hàng hoặc hệ thống quản lý thông tin KTV thì sẽ có thông báo xác nhận đăng kí thành công. Khi đó, người dùng có thể dùng hệ thống để yêu cầu sửa chữa - đối với khách hàng hoặc nhận yêu cầu sửa chữa - đối với KTV. Khi khách hàng gửi yêu cầu lên hệ thống thì yêu cầu sẽ được chuyển sang hệ thống xử lý yêu cầu của khách hàng. Hệ thống xử lý yêu cầu của khách hàng sẽ tìm KTV phù hợp để giải quyết yêu cầu của khách hàng. Sau đó, các thông tin sẽ được in ra định kì hoặc theo yêu cầu bằng hệ thống in báo cáo.

III. Kiến trúc hệ thống

3.1 Bản thiết kế kiến trúc hệ thống (Architectural Design)

3.1.1 Bản thiết kế hệ thống (System Architecture)



Mô tả:

Hệ thống được phát triển dựa trên mô hình Client-Server. Trong đó , trên các thiết bị di động ứng dụng được thiết kế riêng để chạy trên các hệ điều hành Android, IOS tương ứng. Giao thức truyền dữ liệu được chọn là HTTP để đảm bảo tốc độ truyền tải dữ liệu. Về phía Server, ứng dụng AFIX service và hệ cơ sở dữ liệu sẽ được tách riêng ra và sử dụng giao thức HTTP để truyền dự liệu, nhằm mục đích bảo mật, và thuận tiện trong việc nâng cấp vào bảo trì hệ thống. Trên Application Server, chúng ta sẽ chọn Apache để tiếp nhận và xử lý các yêu cầu tới.

Mô tả tổng quát chức năng của từng bộ phận:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Nền tảng | Chức năng |
| 1 | AFIX Application For Android | Android (Lolipop) | Tiếp nhận yêu cầu của khách hàng thông qua giao diện tương tác. Sau đó gửi về Server. |
| 2 | AFIX Application For IOS | IOS( IOS 10.0) | Tiếp nhận yêu cầu của khách hàng thông qua giao diện tương tác. Sau đó gửi về Server. |
| 3 | Chrome | Window 10 | Tiếp nhận yêu cầu của khách hàng thông qua giao diện tương tác. Sau đó gửi về Server. |
| 4 | Apache | Window Server 2003 | Tiếp nhận yêu cầu từ client và gửi cho AFIX Service xử lý. |
| 5 | AFIX Service | Window Server 2003 | Xử lý yêu cầu và hồi đáp kết quả cho Client. |
| 6 | DMS | Window Server 2003 | Tiếp nhận yêu cầu từ AFIX Service và quản lý hệ cơ sở dữ liệu. |

3.1.2 Mô hình lớp (Class diagram)

Mỗi chức năng được trình bày theo mẫu sau:

**Mục đích:** Trình bày chức năng của thiết kế này

**Prototype:** Trình bày mã giả chức năng

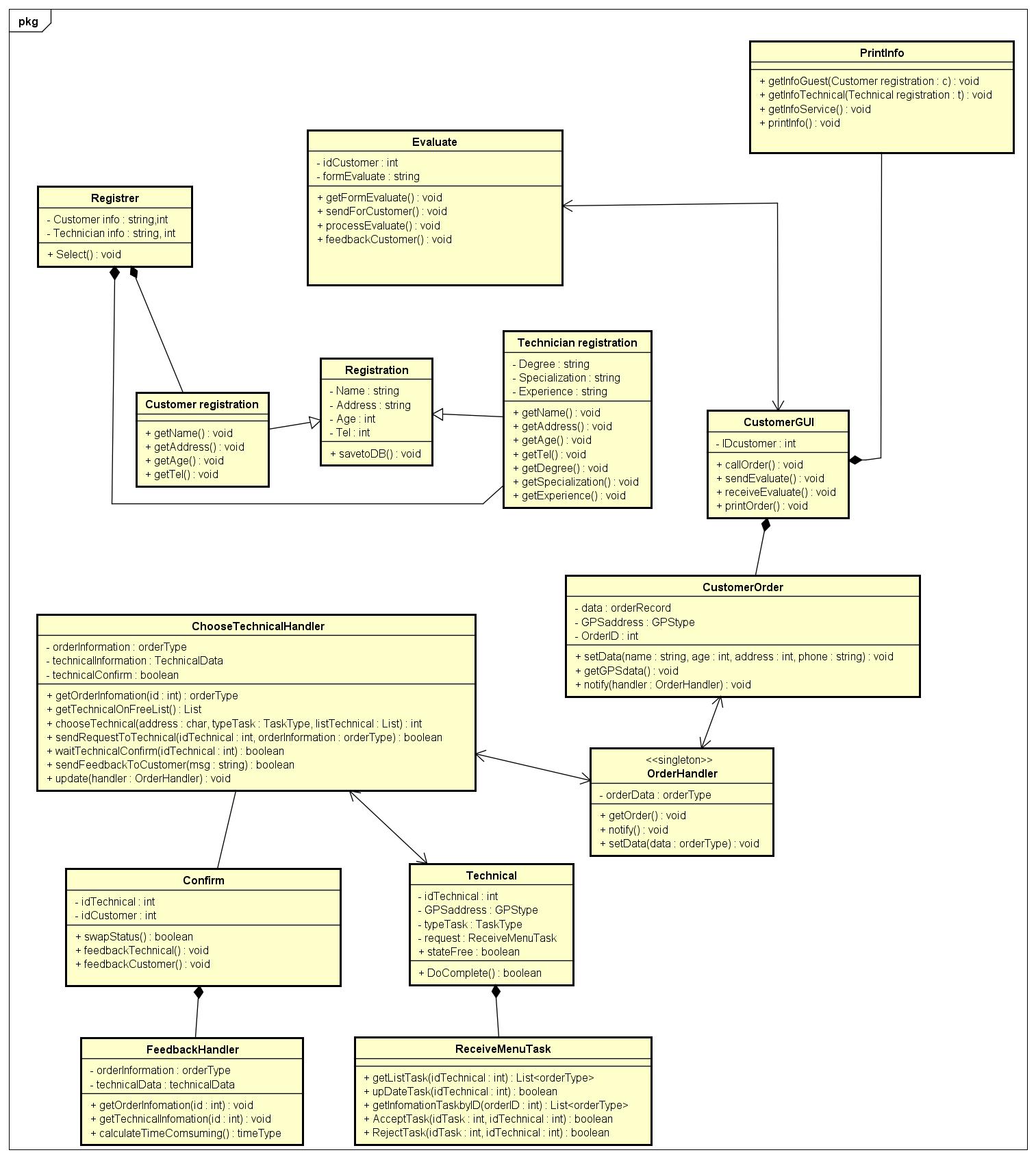
**Ngõ vào:** Trình bày ngõ vào (input) của chức năng

**Ngõ xuất:** Trình bày ngõ ra (output) của chức năng

**Gọi bởi:** Trình bày chức năng bị gởi đối tượng khác

**Gọi:** Trình bày chức năng gọi đối tượng khác

**Giải thuật:** Trình bày giải thuật thực hiện



3.2 Mô tả phân rã (Decomposition Description)

3.2.1 Register

**Mục đích:** Cho người dùng lựa chọn đăng kí là khách hàng hay đăng ký là KTV

**Prototype:** seclet CustomerInfo, select TechnicianInfo

**Ngõ vào:** Không

**Ngõ xuất:** Không

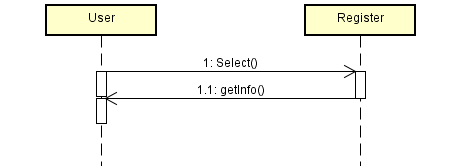
**Gọi bởi:** Không

**Gọi:** Không

**Giải thuật:**

1. Nhận yêu cầu của người dùng.
2. Xuất ra form đăng kí phù hợp.

**Biểu đồ tuần tự (Squence diagram):**

****

3.3.2 Customer registration

**Mục đích:** Xuất ra form đăng kí khách hàng.

**Prototype:** get CustomerInfo

**Ngõ vào:** Name-string, Address-string, Age-int,Tel-int

**Ngõ xuất:** Không

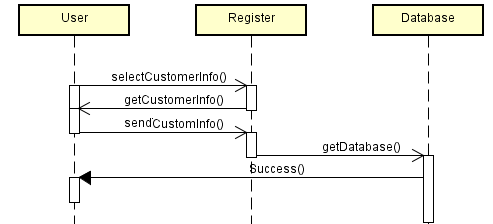
**Gọi bởi:** Register

**Gọi:** Không

**Giải thuật:**

1. Nhận tham số truyền vào
2. Lưu thành dữ liệu khách hàng

**Biểu đồ tuần tự (Squence diagram):**



3.3.3 Technician registration

**Mục đích:** Xuất ra form đăng kí KTV.

**Prototype:** get TechnicianInfo

**Ngõ vào:** Name-string, Address-string, Age-int,Tel-int, Degree-string, Specialization-string, Experience-string

**Ngõ xuất:** Không

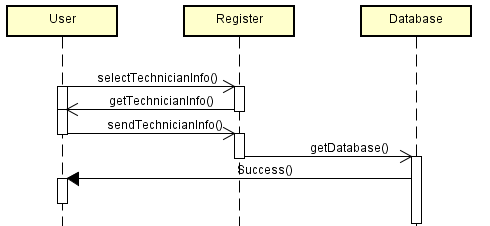
**Gọi bởi:** Register

**Gọi:** Không

**Giải thuật:**

1. Nhận tham số truyền vào.
2. Lưu thành dữ liệu KTV.

**Biểu đồ tuần tự (Squence diagram):**



3.3.4 Evaluate

3.3.4.1 getFormEvaluate

**Mục đích:** Lấy mẫu đánh giá KTV

**Prototype:** public void getFormEvaluate()

**Ngõ vào:** formEvaluate : string

**Ngõ xuất:** Mẫu đánh giá kiểu string

**Gọi bởi:** Technical

**Gọi:** Không

**Giải thuật:**

1. Truy cập hệ cơ sở dữ liệu
2. Lấy ra mẫu đánh giá

3.3.4.2 sendForCustomer

**Mục đích:** gửi mẫu đánh giá cho khách hàng

**Prototype:** public void sendForCustomer ()

**Ngõ vào:** idCustomer : int

**Ngõ xuất:** Không

**Gọi bởi:** Không

**Gọi:** Không

**Giải thuật:**

1. Nhận tham số truyền
2. Gửi dữ liệu

3.3.4.3 processEvaluate

**Mục đích:** Xử lí bản đánh giá từ khách hàng và lưu vào cơ sở dữ liệu

**Prototype:** public void processEvaluate()

**Ngõ vào:**

**Ngõ xuất:** Không

**Gọi bởi:** Không

**Gọi: Không**

**Giải thuật:**

1. Nhận tham số truyền
2. Xử lí và lưu dữ liệu

**Biểu đồ tuần tự (Squence diagram):**

3.3.4.4 feedbackCustomer

**Mục đích:** Phản hồi đánh giá từ khách hàng

**Prototype:** public void feedbackCustomer ()

**Ngõ vào:** idCustomer : int

**Ngõ xuất:** Không

**Gọi bởi:** Không

**Gọi:** Không

**Giải thuật:**

1. Nhận tham số truyền
2. Gửi phản hồi

**Biểu đồ tuần tự (Squence diagram):**

3.3.5 CustomerGUI

**Mục đích:**

**Prototype:**

**Ngõ vào:**

**Ngõ xuất:**

**Gọi bởi:**

**Gọi:**

**Giải thuật:**

**Biểu đồ tuần tự (Squence diagram):**

3.3.6 Printinfo

**Mục đích:**

**Prototype:**

**Ngõ vào:**

**Ngõ xuất:**

**Gọi bởi:**

**Gọi:**

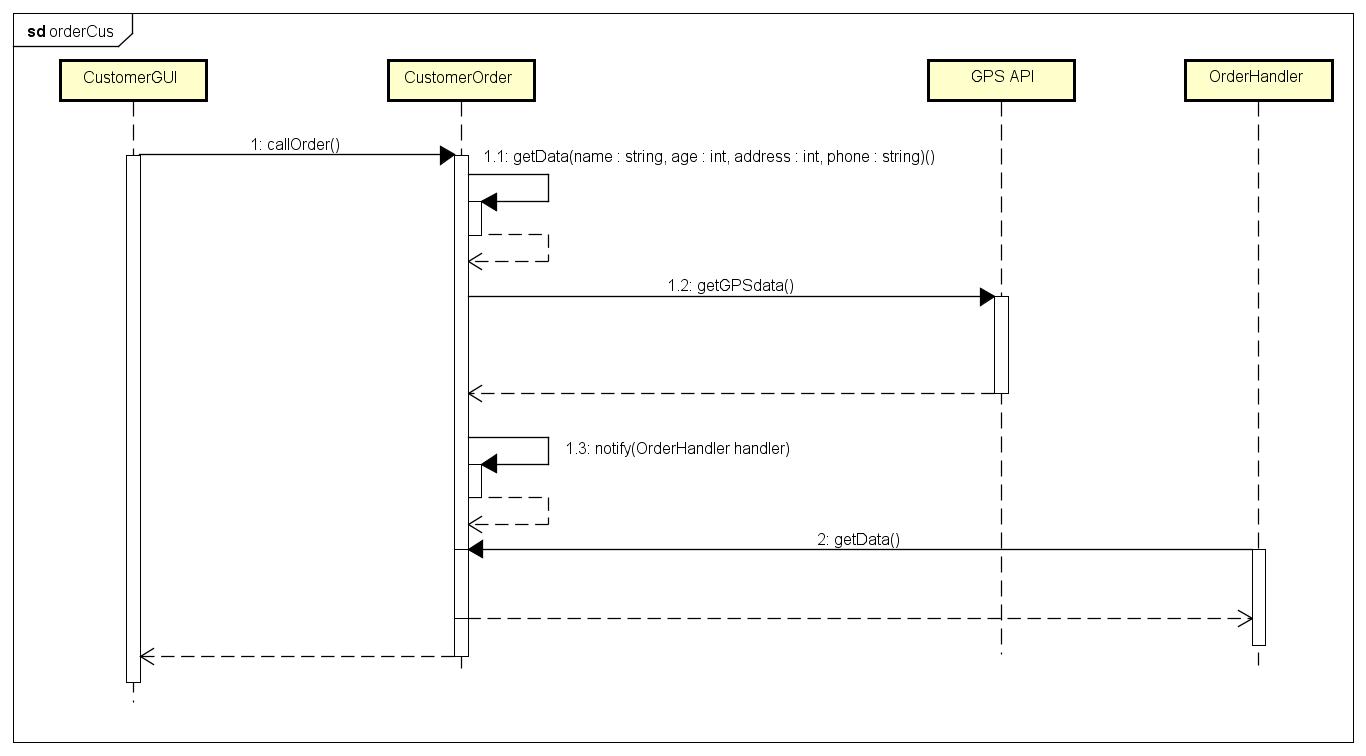
**Giải thuật:**

**Biểu đồ tuần tự (Squence diagram):**

3.3.7 CustomerOrder

Đối tượng của class này sẽ được tạo ra trong class Customer GUI. Class này chứa thông tin về đơn hàng của khách hàng. Class này chứa những phương thức để thiết đặt dữ liệu đơn hàng, lấy địa chỉ của khách hàng thông qua GPS, đóng gói dữ liệu khách hàng.

**Biểu đồ tuần tự (Squence diagram):**



3.3.7.1 setData

**Mục đích:** Thiết đặt dữ liệu của khách hàng

**Prototype:** public void setData(string name, int age, int address, string phone)

**Ngõ vào:** name - string, age - integer, address - integer, phone – string

**Ngõ xuất: Không**

**Gọi bởi: CustomerGUI**

**Gọi: Không**

**Giải thuật:**

1. Nhận tham số truyền vào và khởi tạo giá trị cho đối tượng CustomerOrder.

3.3.7.2 getGPSdata

**Mục đích:** Lấy dữ liệu vị trí của khách hàng

**Prototype:** public boolean getGPSdata()

**Ngõ vào:** Không

**Ngõ xuất:** Không

**Gọi bởi: CustomerGUI**

**Gọi:** Không

**Giải thuật:**

1. Gọi API tương tác để lấy dữ liệu từ GPS (kinh độ, vĩ độ)
2. Dùng API GoogleMap để tra cứu kinh độ và vĩ độ
3. Trả về kết quả tìm được và lưu vào biến private GPSaddress

3.3.7.3 notify

**Mục đích:** Đóng gói và chuyển dữ liệu khi có yêu cầu từ đối tượng OrderHandler

**Prototype:** public void notify (OrderHandler handler)

**Ngõ vào:** handler -OrderHandler

**Ngõ xuất:** Không

**Gọi bởi:** CustomerGUI

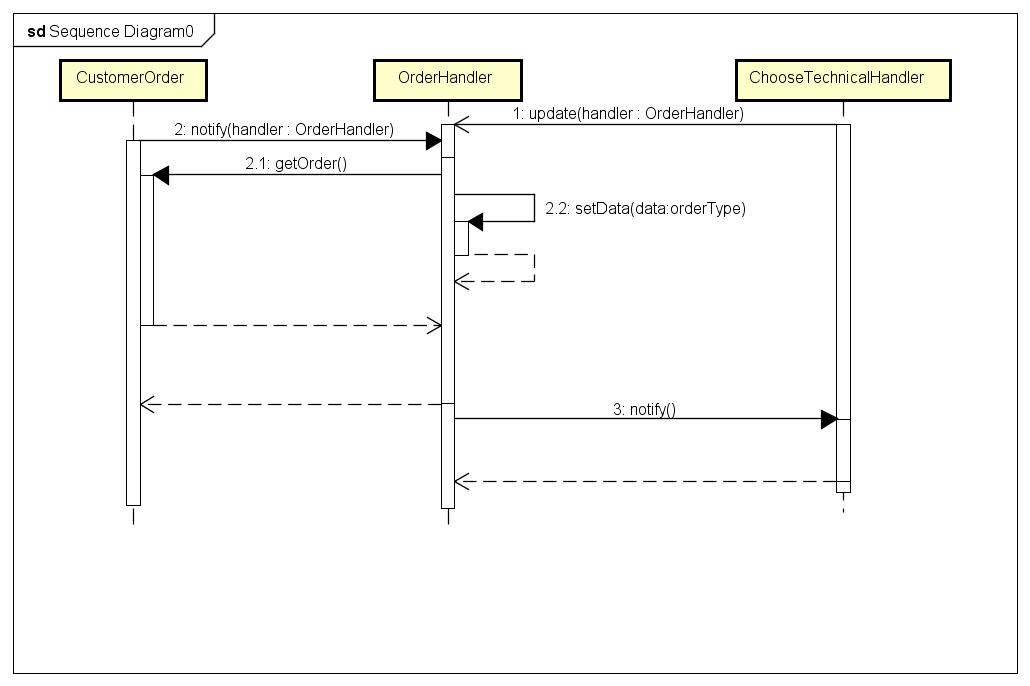
**Gọi:** OrderHandler

**Giải thuật:**

1. Khởi tạo biến kiểu orderType
2. Lấy dữ liệu từ các biến private.
3. Truyền dữ liệu từ các biến private sang biến orderType.
4. Gửi dữ liệu xuống thành phần yêu cầu.
   * 1. OrderHandler

Đối tượng của class là một đối tượng duy nhất (Singleton) chịu trách nhiệm xử lý đơn hàng khi có đơn hàng tới.

**Biểu đồ tuần tự (Squence diagram):**



3.3.8.1 getOrderData

**Mục đích:** Yêu cầu dữ liệu của CustomerOrder

**Prototype:** public boolean getOrderData(CustomerOrder inputData)

**Ngõ vào:** inputData - CustomerOrder

**Ngõ xuất:** Không

**Gọi bởi:** Không

**Gọi:** this.setData

**Giải thuật:**

1. Nhận tham số truyền vào
2. Lưu dữ liệu vào dữ liệu private

3.3.8.2 notify

**Mục đích:** Cập nhập dữ liệu cho các thành phần đăng kí

**Prototype:** public string notify ()

**Ngõ vào:** Không

**Ngõ xuất:** Không

**Gọi bởi:** Không

**Gọi:** Không

**Giải thuật:**

1. Chờ cho tới khi có yêu cầu dữ liệu từ thành phần đăng kí
2. Khi có thành phần đăng ký yêu cầu dữ liệu, trả về dữ liệu private

3.3.8.3 setData

**Mục đích:** Lưu dữ liệu vào biến private ( orderData)

**Prototype:** public string setData (orderType : data)

**Ngõ vào:** data - orderType

**Ngõ xuất:** Không

**Gọi bởi:** this.getOrderData

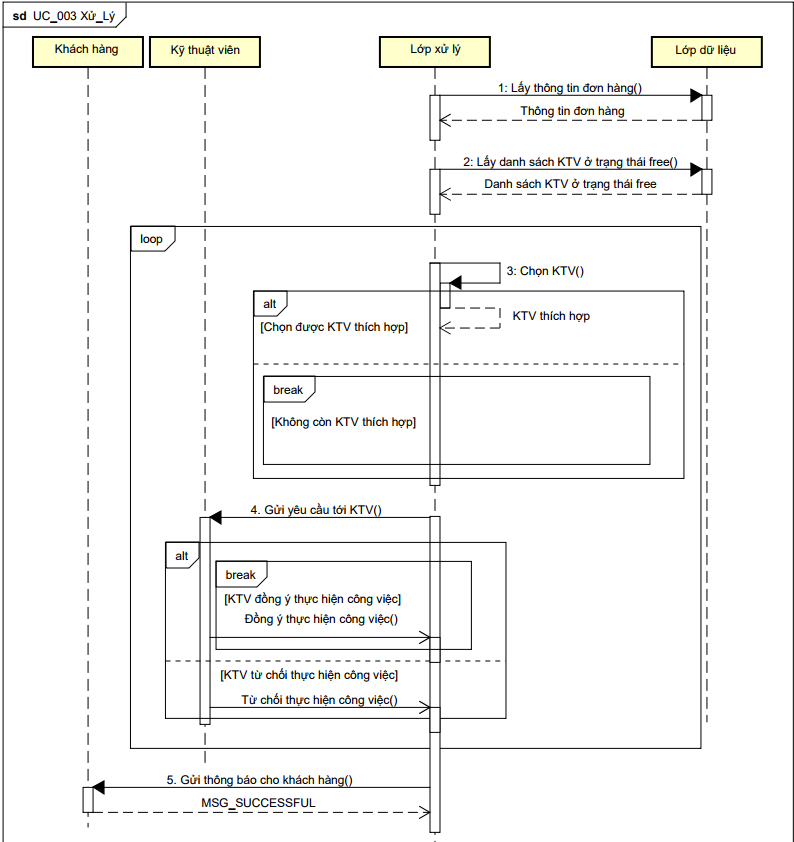
**Gọi:** Không

**Giải thuật:**

1. Chờ cho tới khi có yêu cầu dữ liệu từ thành phần đăng kí
2. Khi có thành phần đăng ký yêu cầu dữ liệu, trả về dữ liệu private

3.3.9 ChooseTechnicalHandler

**Biểu đồ tuần tự (Squence diagram):**

******

3.3.9.1 getOrderInfomation

**Mục đích:** Lấy thông tin đơn hàng

**Prototype:** public orderType getOrderInfomation (int id)

**Ngõ vào:** id - integer

**Ngõ xuất:** Đối tượng kiểu orderType

**Gọi bởi:** ChooseTechnicalHandler

**Gọi:** Không

**Giải thuật:**

1. Truy cập vào hệ cơ sở dữ liệu
2. Lấy thông tin đơn hàng từ hệ cơ sở dữ liệu dữa vào id của đơn hàng

3.3.9.2 getTechnicalOnFreeList

**Mục đích:** Lấy danh sách các kỹ thuật viên đang ở trạng thái free

**Prototype:** private List<Technical> getTechnicalOnFreeList ()

**Ngõ vào:** Không

**Ngõ xuất:** Danh sách các kỹ thuật viên

**Gọi bởi:** ChooseTechnicalHandler

**Gọi:** Không

**Giải thuật:**

1. Truy cập vào hệ cơ sở dữ liệu
2. Lấy danh sách các kỹ thuật viên ở trạng thái free

3.3.9.3 chooseTechnical

**Mục đích:** Chọn kỹ thuật viên phù hợp nhất với đơn hàng

**Prototype:** private int chooseTechnical(char address, TaskType typeTask, List<Technical> listTechnical)

**Ngõ vào:** address kiểu char, typeTask kiểu TaskType, listTechnical kiểu List<Technical>

**Ngõ xuất:** id của kỹ thuật viên được chọn – kiểu integer

**Gọi bởi:** ChooseTechnicalHandler

**Gọi:** Không

**Giải thuật:**

1. Lọc danh sách kỹ thuật viên có thể thực hiện công việc trong đơn hàng
2. Lặp qua danh sách các kỹ thuật viên để so sách vị trí của kỹ thuật viên hiện tại với địa chỉ trong đơn hàng
3. Chọn kỹ thuật viên có địa chỉ hiện tại gần với địa chỉ trong đơn hàng nhất
4. Trả về id của kỹ thuật viên được chọn

3.3.9.4 sendRequestToTechnical

**Mục đích:** Gửi yên cầu thực hiện công việc tới kỹ thuật viên được chọn

**Prototype:** private Boolean sendRequestToTechnical(int idTechnical, orderType orderInformation)

**Ngõ vào:** idTechnical kiểu int, orderInformation kiểu orderType

**Ngõ xuất:** Trả về true nếu gửi thành công và false nếu thất bại

**Gọi bởi:** ChooseTechnicalHandler

**Gọi:** waitTechnicalConfirm, chooseTechnical

**Giải thuật:**

1. Gửi thông tin đơn hàng tới kỹ thuật viên
2. Gọi hàm waitTechnicalConfirm
3. Nếu hàm waitTechnicalConfirm trả về false thì gọi lấy danh sách kỹ thuật viên ở trạng thái free còn lại.
4. Nếu danh sách rỗng thì gọi sendFeedbackToCustomer với message “Không có kỹ thuật viên nào phù hợp”. Ngược lại, gọi chooseTechnical để chọn kỹ thuật viên mới

3.3.9.5 waitTechnicalConfirm

**Mục đích:** Chờ kỹ thuật viên xác nhận thực hiện công việc

**Prototype:** private Boolean waitTechnicalConfirm(int idTechnical)

**Ngõ vào:** idTechnical kiểu integer

**Ngõ xuất:** Trả về true nếu kỹ thuật viên đồng ý thực hiện công việc và trả về false nếu kỹ thuật viên từ chối thực hiện công việc hoặc hết thời gian chờ 10p

**Gọi bởi:** ChooseTechnicalHandler

**Gọi:** Không

**Giải thuật:**

1. Dùng vòng lặp để lấy trạng thái xác nhận của kỹ thuật viên sau mỗi 1 giây (tương tự như Thread.sleep(1000) trong Java) với điều kiện thời gian chờ nhỏ hơn 10 phút
2. Nếu kỹ thuật viên đồng ý thì break ra khỏi vòng lặp, gọi sendFeedbackToCustomer với message `Kỹ thuật viên: technicalInformation đang đến thực hiện công việc` và trả về true; Nếu kỹ thuật viên từ chối thì break ra khỏi vòng lặp và trả về false.
3. Nếu hết thời gian chờ trả về false.

3.3.9.6 sendFeedbackToCustomer

**Mục đích:** Gửi message tới khách hàng với nội dung có tìm được kỹ thuật viên phù hợp hay không với đơn hàng của khách hàng

**Prototype:** private Boolean sendFeedbackToCustomer(String msg)

**Ngõ vào:** msg kiểu String

**Ngõ xuất:** Trả về true nếu gửi message thành công và false nếu thật bại

**Gọi bởi:** sendRequestToTechnical,

**Gọi:** getOrderInfomation

**Giải thuật:**

1. Lấy id của customer từ getOrderInfomation
2. Gửi msg tới customer.

3.3.9.7 update

**Mục đích:** Nhận notify khi có đơn hàng mới từ OrderHanler

**Prototype:** public void update(OrderHandler handler)

**Ngõ vào:** handler kiểu OrderHandler

**Ngõ xuất:** Không

**Gọi bởi:** ChooseTechnicalHandler

**Gọi:** Không

**Giải thuật:**

1. Cập nhật xem hệ thống có đơn hàng mới hay không
2. Nhận notify từ OrderHanler

3.3.10 Confirm

3.3.10.1 swapStatus

**Mục đích:** Đổi trạng thái của KTV

**Prototype:** public boolean swapstatus()

**Ngõ vào:** idTechnical : int

**Ngõ xuất:** boolean

**Gọi bởi:** Không

**Gọi:** Không

**Giải thuật:**

1. Nhận tham số truyền

2. Đổi trạng thái

**Biểu đồ tuần tự (Squence diagram):**

3.3.10.2 feedbackTechnical

**Mục đích:** Gửi phản hồi cho KTV

**Prototype:** public void feedbackTechnical()

**Ngõ vào:** idTechnical : int

**Ngõ xuất:** Không

**Gọi bởi:** Không

**Gọi:** Không

**Giải thuật:**

1. Nhận tham số truyền

2. Gửi phản hồi

**Biểu đồ tuần tự (Squence diagram):**

3.3.10.3 feedbackCustomer

**Mục đích:** Gửi phản hồi cho khách hàng

**Prototype:** public void feedbackCustomer()

**Ngõ vào:** idCustomer : int

**Ngõ xuất:** Không

**Gọi bởi:** Không

**Gọi:** Không

**Giải thuật:**

1. Nhận tham số truyền

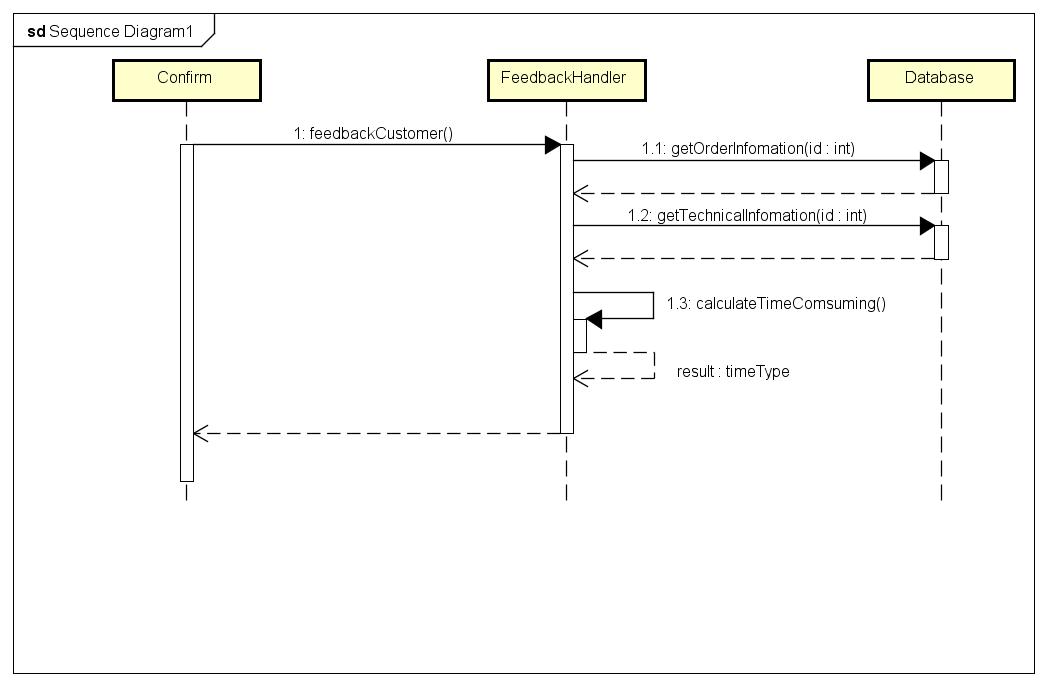
2. Gửi phản hồi

**Biểu đồ tuần tự (Squence diagram):**

3.3.11 FeedbackHandler

Đối tượng Class chịu trách nhiệm thu thập thông tin và phản hồi lại cho khách hàng.

**Biểu đồ tuần tự (Squence diagram):**



3.3.11.1 getOrderInfomation

**Mục đích:** Lấy thông tin đơn hàng từ hệ cơ sở dữ liệu

**Prototype:** public Boolean getOrderInfomation(int id)

**Ngõ vào:**  id kiểu integer

**Ngõ xuất:** không

**Gọi bởi:** Confirm

**Gọi:** Không

**Giải thuật:**

1. Truy cập vào hệ cơ sở dữ liệu
2. Lưu dữ liệu lấy được từ cơ sở dữ liệu và lưu vào dữ liệu private (orderInformation)

3.3.11.2 getTechnicalInfomation

**Mục đích:** Lấy thông tin kỹ thuật viên được chọn từ hệ cơ sở dữ liệu

**Prototype:** public Boolean getTechnicalInfomation (int id)

**Ngõ vào:**  id kiểu integer

**Ngõ xuất:** Không

**Gọi bởi:** Confirm

**Gọi:** Không

**Giải thuật:**

1. Truy cập vào hệ cơ sở dữ liệu
2. Lưu dữ liệu lấy được từ cơ sở dữ liệu và lưu vào dữ liệu private (technicalData)

3.3.11.3 calculateTimeComsuming

**Mục đích:** Nhận thông tin đơn hàng

**Prototype:** public timeType calculateTimeComsuming ()

**Ngõ vào:** Không

**Ngõ xuất:** Đối tượng kiểu timeType

**Gọi bởi:** Confirm

**Gọi:** Không

**Giải thuật:**

1. Lấy dữ liệu từ biến private (orderInformation, technicalData)
2. Tính toán khoảng cách và thời giant rung bình đi lại giữa khách hàng và kĩ thuật viên.
3. Hồi đáp thông tin cho khách hàng

3.3.11.4 feedbackForCustomer

**Mục đích:** Gửi thông tin hồi đáp cho khách hàng

**Prototype:** public Boolean feedbackForCustomer (int id)

**Ngõ vào:**  id kiểu integer

**Ngõ xuất:** không

**Gọi bởi:** Confirm

**Gọi:** Không

**Giải thuật:**

1. Lấy thông tin từ dữ liệu private
2. Đóng gói dữ liệu
3. Gửi dữ liệu được đóng gói tới khách hàng theo ID

3.3.12 Technical

Class này chứa thông tin về Kỹ thuật viên ( idTechnical, GPSaddress, stateFree). Class được tạo khi KTV login vào hệ thống và đang trong trạng thái onjob

*3.3.12.1: DoComplete*

**Mục đích**: xác nhận hoàn thành công việc

**Prototype:** public boolean DoComplete()

**Ngõ vào:** không

**Ngõ xuất:** boolean

**Gọi bởi:** Technical

**Gọi:** không

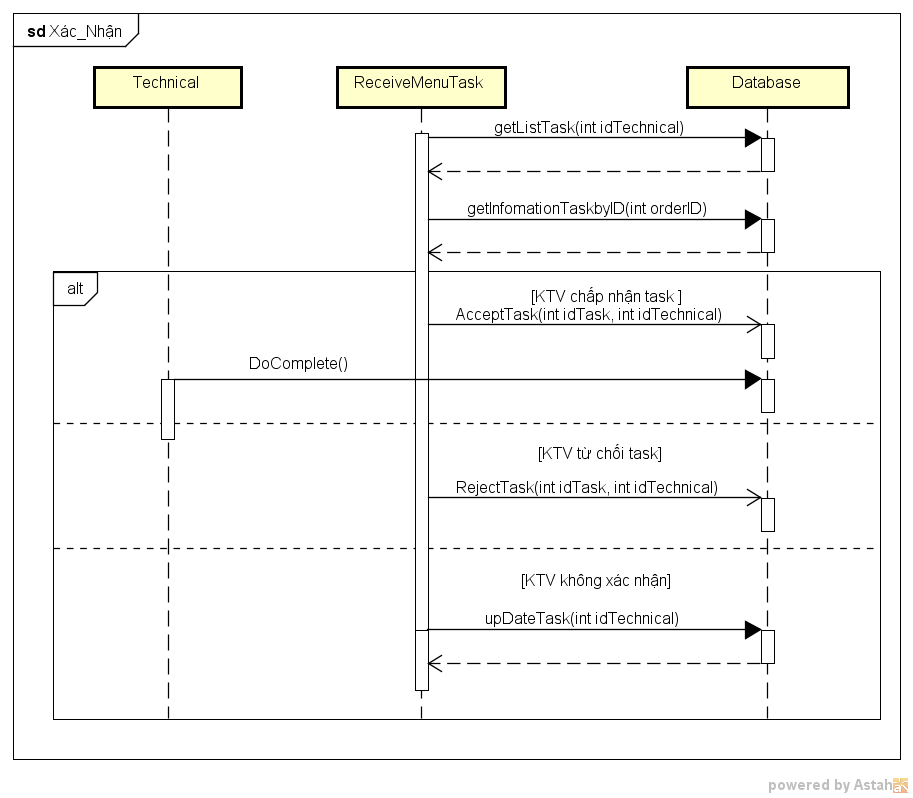
**Giải thuật:**

1. Set giá trị trả về của hàm thành “true”

3.3.13 ReceiveMenuTask

Class này được gọi khi Technical được tạo

**Biểu đồ tuần tự (Sequence Diagram) :**



*3.3.13.1: getListTask*

**Mục đích:** nhận danh sách công việc mà hệ thống đã ứng với ID định danh của KTV

**Prototype:** public List<orderType> getListTask (int idTechnical)

**Ngõ vào:** int idTechnical (ID định danh của KTV)

**Ngõ xuất:** List<orderType> (gồm 1 menuTask)

**Gọi bởi****:** ReceiveMenuTask

**Gọi:** không

**Giải thuật:**

1. Gọi list orderType ứng với ID technical từ cơ sở dữ liệu
2. Hiển thị ListView chứa số lượng, số thứ tự các task

*3.3.13.2: upDateTask*

**Mục đích:** cập nhật lại danh sách công việc sau khoảng thời gian nhất định (10s), hoặc khi công việc đã được nhận bởi kĩ thuật viên khác

**Prototype:** public boolean upDateTask(int idTechnical)

**Ngõ vào:** int idTechnical (ID định danh của KTV)

**Ngõ xuất:** boolean

**Gọi bởi:** ReceiveMenuTask

**Gọi:** getListTask

**Giải thuật:**

1. Gọi getListTask sau khoảng thời gian 10s.
2. Trả về true nếu đã update.

*3.3.14.3: getInfomationTaskbyID*

**Mục đích:** mỗi Task sẽ có 1 ID riêng, hàm này sẽ lấy thông tin task được KTV chọn trong menuTask

**Prototype:** public List<orderType> getInfomationTaskbyID (int orderID)

**Ngõ vào**: intorderID

**Ngõ xuất:** List<orderType>

**Gọi bởi:** ReceiveMenuTask

**Gọi:** không

**Giải thuật:**

1. Lấy thông tin orderType ứng với ID order (ID của đơn hàng)
2. Hiển thị ListView chứa thông tin name, số điện thoại, địa chỉ, giá của đơn hàng

*3.3.13.4 :* *AcceptTask*

**Mục đích:** hàm được gọi khi KTV chấp nhận task (chọn “Accept”) hàm này sẽ được gọi

**Prototype:** public boolean AcceptTask (int idTask, int idTechnical)

**Ngõ vào:** int idTask, int idTechnical

**Ngõ xuất:** boolean

**Gọi bởi:** ReceiveMenuTask

**Gọi:** không

**Giải thuật:**

1. thông tin về đơn hàng và KTV đó sẽ được lưu vào hệ thống

2. đặt lại--> stateFree = false

*3.3.13.5 :* *RejectTask*

**Mục đích:** khi KTV từ chối task hàm này sẽ được gọi

**Prototype:** public boolean RejectTask (int idTask, int idTechnical)

**Ngõ vào:** int idTask, int idTechnical

**Ngõ xuất:** boolean

**Gọi bởi:** ReceiveMenuTask

**Gọi:** không

**Giải thuật:**

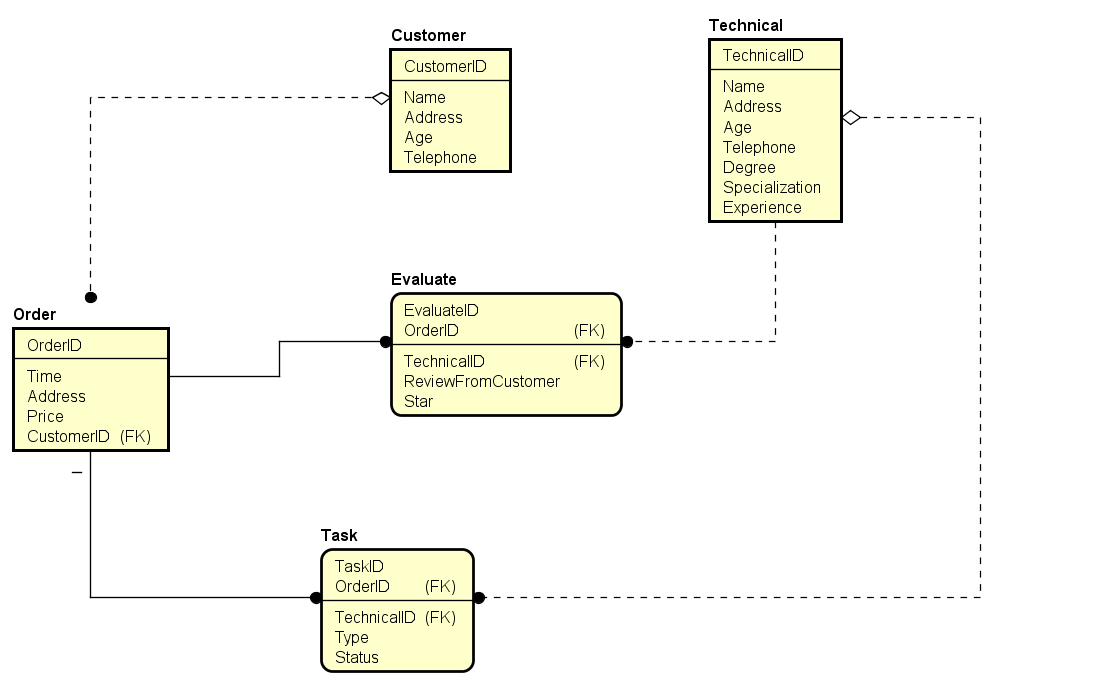
1. gọi TextBox yêu cầu KTV nêu lý do

2. stateFree = true

3.Lý do thiết kế (Design Rationale)

VI. Thiết kế dữ liệu

* 1. Mô tả dữ liệu (Data Description)



* 1. Từ điển dự liệu (Data Dictionary)

Client\_id : Là thông số định danh của khách hàng. Các thông số này không giống nhau và là duy nhất trong hệ thống. Thông số này sẽ được hệ thống tạo ra khi có khách hàng đăng kí dịch vụ. Thông số này có kiểu là Int(10).

Client\_name: Là thông số chứa tên khách hàng. Thông số này có kiểu là Varchar (20).

Password: Là thông số chứa mật khẩu truy cập vào hệ thống của khách hàng. Thông số này có kiểu là Varchar(30).

Contact\_no: Là thông số chứa số điện thoại liên lạc của khách hàng. Thông số này có kiểu là Int(15).

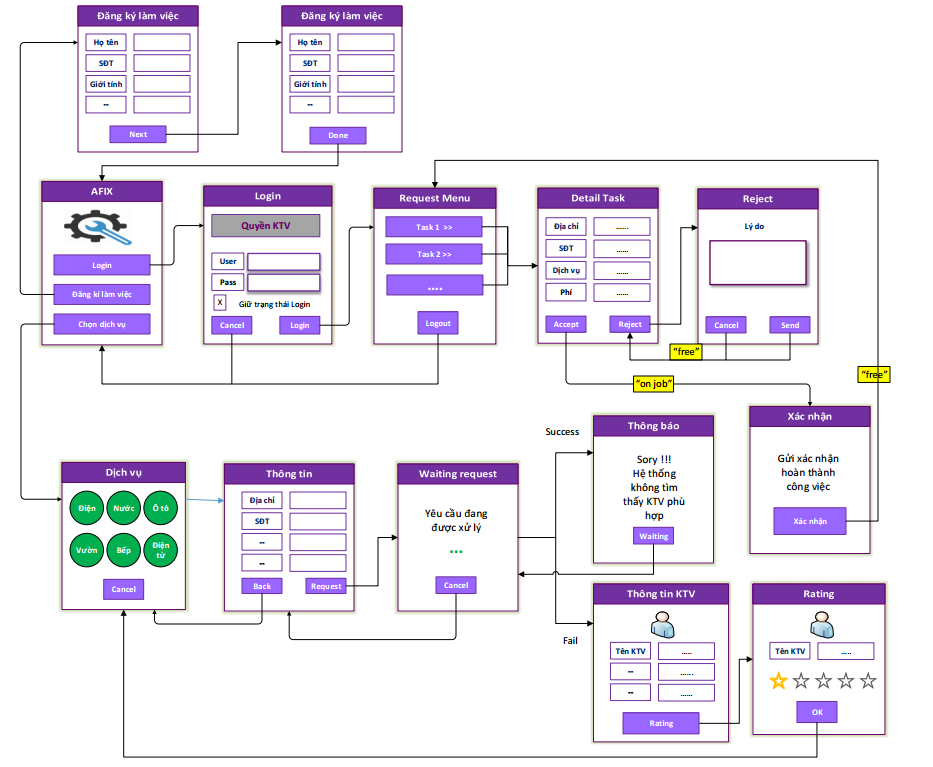
Email\_id: Là thông số chứa địa chỉ email của khách hàng. Thông số này có kiểu là Varchar(30).

Max\_Users: Là thông số chứa số lượng khách hàng tối đa của hệ thống. Thông số này có kiểu là Int(10).

Current\_user: Là thông số chứa số lượng khách hàng đang sử dụng hệ thống. Thông số này có kiểu là Int(10).

V. Thiết kế bộ phận (Component Design)

VI. Thiết kế giao diện tương tác



VII. Ma trận yêu cầu

VIII. Phụ lục