Chef For Everyone - CFE

\

Use-Case Specification: - Managing chef's service.

**- Booking chef service.**

**- Rating chef.**

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 17/Jul/21 | 1.0 | Show 3 use case represent main values of CFE.  Architecture styles chosen and explained reasons why.  Define UML model for the selected use cases, applying the architecture styles chosen. | Nguyễn Phan Khôi Nguyên  Nguyễn Thanh Tân  Nguyễn Tấn Sang (K14)  Nguyễn Tấn Sang (K13) |

Table of Contents

[1. Use-case Model 4](#_Toc77529128)

[2. Use-case Specifications 5](#_Toc77529129)

[2.1 Use-case: Managing chef’s service 5](#_Toc77529130)

[2.2 Use-case: Booking chef service 5](#_Toc77529131)

[2.3 Use-case: Rating chef 6](#_Toc77529132)

[3. Evaluation architecture style 7](#_Toc77529133)

[4. UML design 7](#_Toc77529134)

# Use-case Model



List of use-case will be shown in this document:

- Managing chef’s service:

Description: As a chef, I want to add new service, dish, food to show on my profile for booking.

- Booking chef service:

Description: As a customer, I want to search and after that book favorite chef, dish, food, and quantity for event.

- Rating chef:

Description: As a customer, I want to rate chef and leave comment about quality of chef after I have been served.

# Use-case Specifications

## Use-case: Managing chef’s service

|  |  |
| --- | --- |
| Use case Name | Managing chef’s service |
| Brief description | This use case describes how chef can provide services for Customer. |
| Actors  (Actors) | Chef |
| Basic Flow | 1. At the homepage, the chef goes to “Services” section. 2. Chef clicks on “Add service” button. 3. System displays Add service screen. 4. Chef fills in service’s information. 5. Chef clicks on “Add” button and system sends success message on page. 6. System renders to home page and adding service done. |
| Alternative Flows | **Alternative flow 1: Information on the form not full fill**   1. From step 4 of the basic flow, system will not allow chef click “Add” button and send text alert on the screen. 2. Continue step 4 until fill full. |
| Pre-conditions | Chef have to login at <https://www.cfe.chef-for-everyone.com>. |
| Post-conditions | Chef successfully adding services and get success notification on screen.  The homepage will display number of services had added. |

## Use-case: Booking chef service

|  |  |
| --- | --- |
| Use case Name | Booking chef service. |
| Brief description | This use case describes how the Customer can booking a chef and Service to serve at home. |
| Actors  (Actors) | Customer |
| Basic Flow | 1. At the homepage, the user enters keywords on the ‘Search’ field. 2. Users clicks on ‘Search’ button to start searching a service. 3. System displays the service found. 4. User clicks ‘Detail Information’ button on the expected service of chef 5. System renders a detail information services of chef. 6. User clicks “Booking” button to book chef. 7. User fills out form, choose information and payment method. 8. Click “Finish” button and system send success message on page. 9. System renders to home page and booking chef done. |
| Alternative Flows | **Alternative flow 1: User cannot find products searched.**   1. From step 1 of the basic flow, user enters another term. 2. Continue step 2 in the basic flow.   **Alternative flow 2: Information on the form does not fill full**   1. From 7 of the basic flow, system will not allow customer click “Finish” button and send text alert on the screen. 2. Continue step 7 until fill full. |
| Pre-conditions | User have to login before booking chef |
| Post-conditions | The user successfully booking chef and get success notification on screen.  The cart will display number of service user had booked on home screen. |

## Use-case: Rating chef

|  |  |
| --- | --- |
| Use case Name | Rating chef |
| Brief description | This use case describes how the Customer can rating a chef |
| Actors  (Actors) | Customer |
| Basic Flow | 1. After chef have had clicked “done service” button in chef app, the rating screen will display in customer app. 2. Users choose number of stars they want to rate and input comment. 3. Users click on “Submit” button. 4. System renders to home page. |
| Alternative Flows | **Alternative flow 1: Chef rented but not rated yet.**   1. On homepage, click on rented chef. 2. System will display rented chef information screen. 3. Continue from step 2 of basic flow. |
| Pre-conditions | User have to login at homepage <https://www.cfe.chef-for-everyone.com>.  User have to booking chef. |
| Post-conditions | The user successfully booking chef and get success notification on screen.  The rating screen will display. |

# Evaluation architecture style

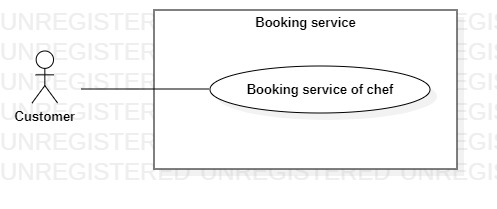
The microservice architecture is chosen as main design approach by following purpose:

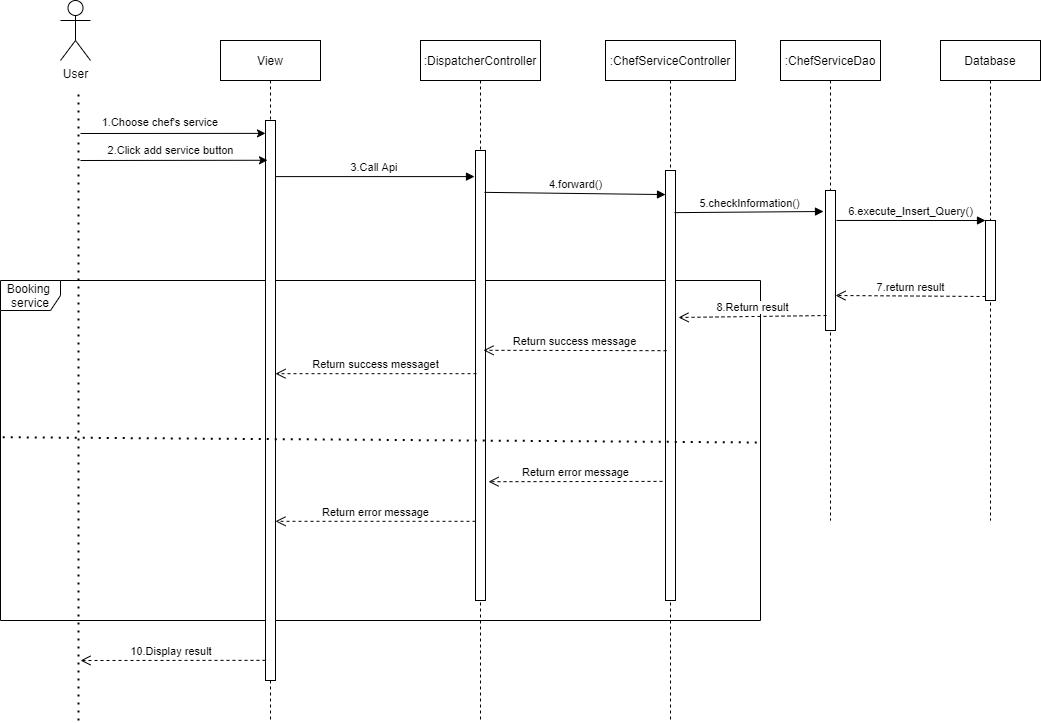
* + - Because microservices allows scale services up or down independently, easily and the cost of is dramatically less than in a monolithic system. Adding new capabilities usually means adding discrete new microservices, not redoing the entire application, which increases both development speed and application stability.
    - Highly maintainable and testable.
    - Loosely coupled.
    - Organized around business capabilities.
    - Development teams can collaborate more easily to speed up time-to-market.

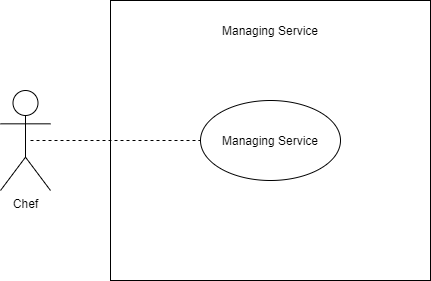
If one microservice fails, all the others will likely continue to work.

# UML design

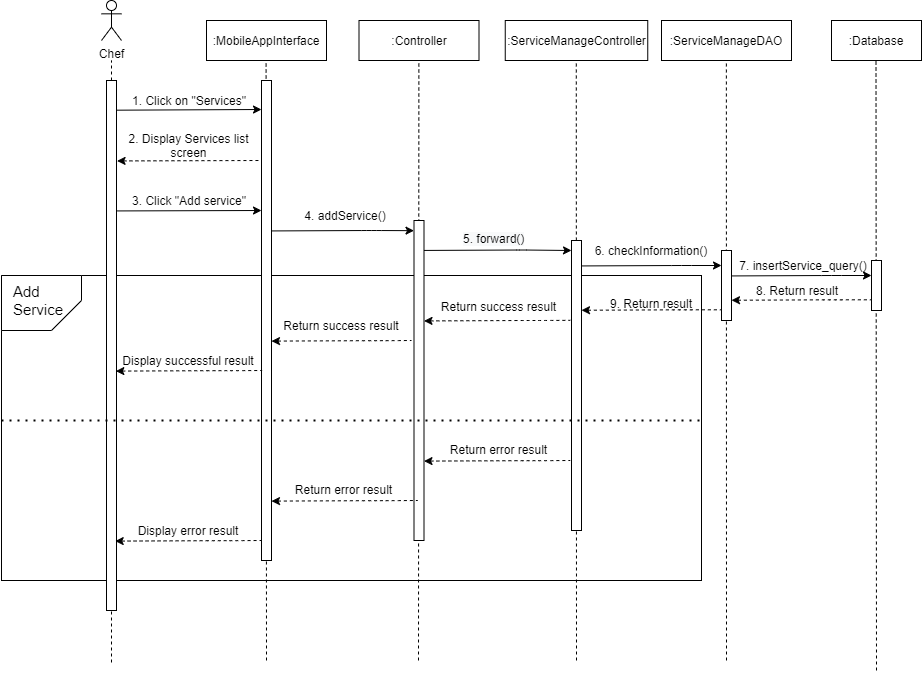
* Booking chef service use case diagram



* Booking chef service sequence diagram
* Managing chef’s service use case diagram



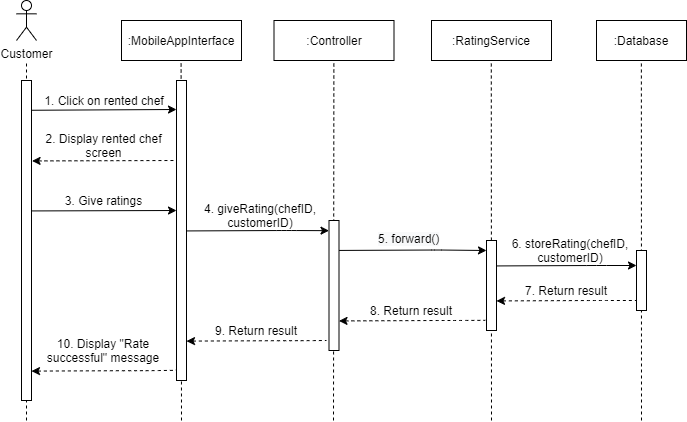
* Managing chef’s service sequence diagram



* Rating chef use case diagram

****

* Rating chef sequence diagram



* CFE ERD

