Software Design Description  
*<Product Name>*

# Summary

This document describes architectural design for *<product>* via 4+1 View approach

The intended audiences are software development team and deployment team.

## Reference

* *<Name of Product Description document>*

# Architecture Style Selection

*Provide justification for choosing architecture style, relevant to FRs, NFRs known in product description*

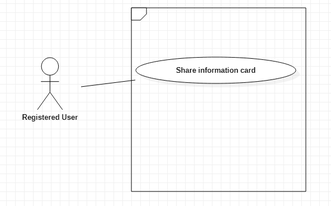
| Selected Architecture Style | Justification of Selection | Affected FRs | Affected NFRs |
| --- | --- | --- | --- |
| Master - Slave | This application is developed to serve some customers in the first moments. However,This app will be upgraded for more customers to have access in the near future . so Master-slave will be applied to this database for future serve. | FR\_02, FR\_01 | Performance |
| Batch | When a user creates an account the system gets information from the user and saves it in the database. After that user must login and the system will get username and password to authen. After login user must create a card by system get information and return information of that user in card | FR\_16, FR\_15, FR\_02 | Performance |
|  |  |  |  |
|  |  |  |  |

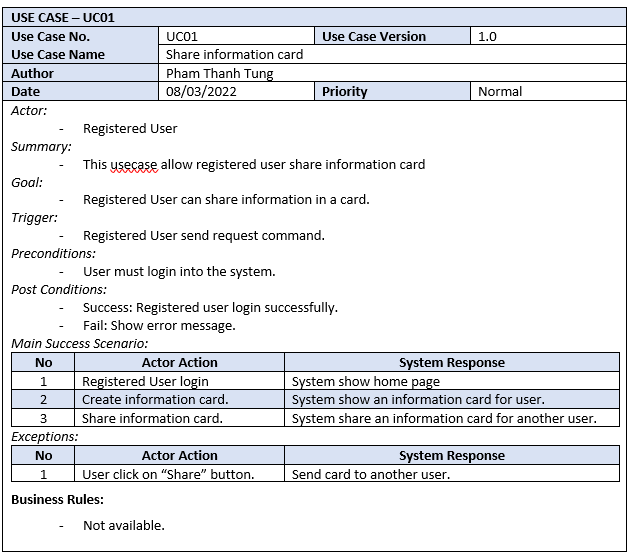
# Architectural Design

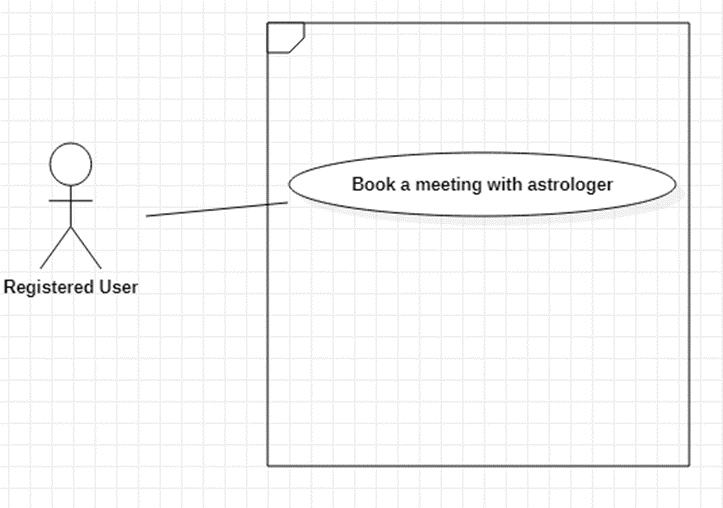
## Scenario View

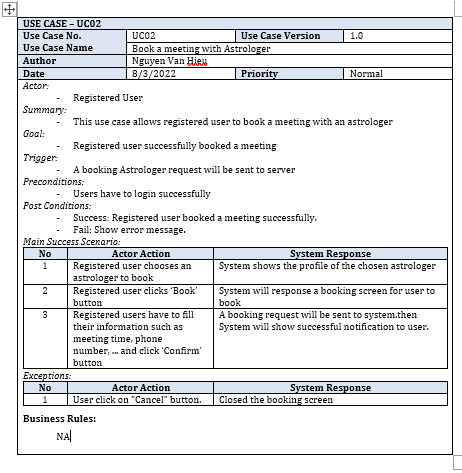
*[Usecase diagram and use case description of key (not all) use cases]*

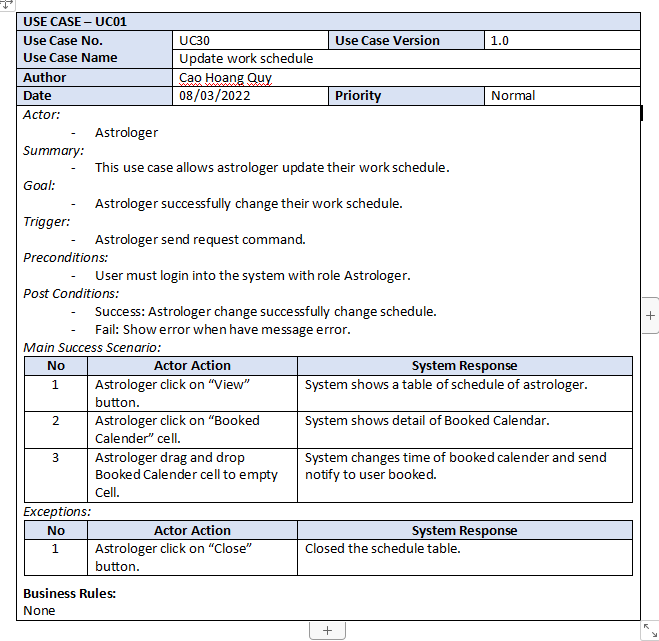
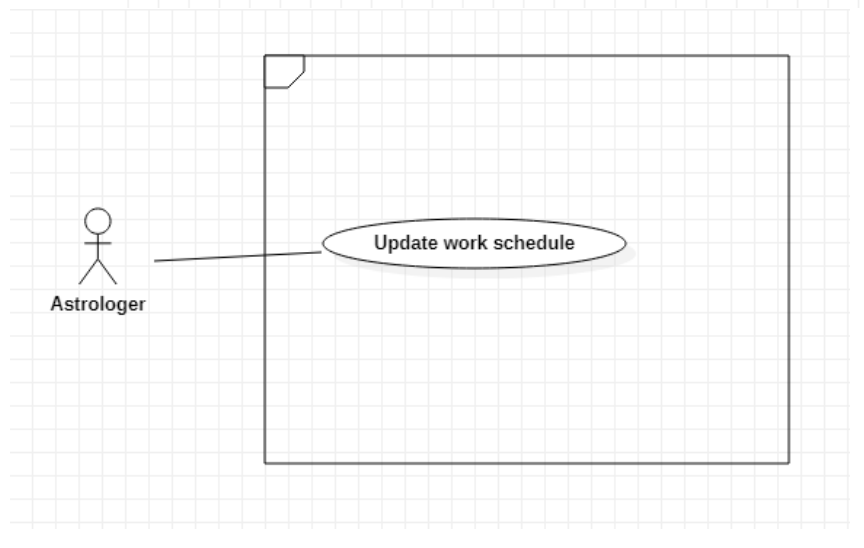
| UC ID | Use case | Actors | Pre-condition | Post-condition | Logical Processing |
| --- | --- | --- | --- | --- | --- |
| UC1 | Share information card | Registered User | User must login into the system | Success: User will have a card with information to share.  Fail: Show message.   |  | | --- | | 1. Users must register an account. 2. User Login. 3. Users create an information card. 4. Users share information cards. 5. Other users take information in cards and view. |
| UC2 | Book a meeting with Astrologer | Registered User | User must login in the system | Success:Registered User booked a meeting successfully  Fail:Show error message | 1.Users must login in the application  2.Users click on Astrologer screen  3.Users choose an Astrologer  4.Users click on ‘Book’ button in Astrologer Detail Screen  5.Users fill in their information in booking form  6.Users click on ‘Confirm’ button to book a meeting |
| UC3 | Update work schedule | Astrologer | User must login into the system with role Astrologer. | Success: Astrologer change successfully change schedule.  Fail: Show error when have message error. | 1. Astrologer click on “View” button. 2. Astrologer click on “Booked Calendar” cell. 3. Astrologer drag and drop Booked Calendar cell to empty Cell. |
| UC14 | Chat | Registered User, Astrologer | User must login into the system | Success: A message will be sent to another user.  Fail: Show error when message not be sent. | 1. Registered User or Astrologer click on another user avatar in the contact list. 2. Registered User or Astrologer type message and click the “Send” button. |
| UC5 | Write blog | Astrologer | User must login into the system with role Astrologer. | Success: Astrologer create successfully blog.  Fail: Show error when have message error. | 1. Astrologer click on “Write blog” button. 2. Astrologer input new blog data and click button “Create” |
| UC6 | Accept customer booking | Astrologer | The Astrologer must login to the system with an account that has the authorized user | Success:Show successfully message  Fail: Show error message. | 1. Astrologer click on “accept” button in the notification panel include:  * Name customer * Work time * Accept button * Cancel button |

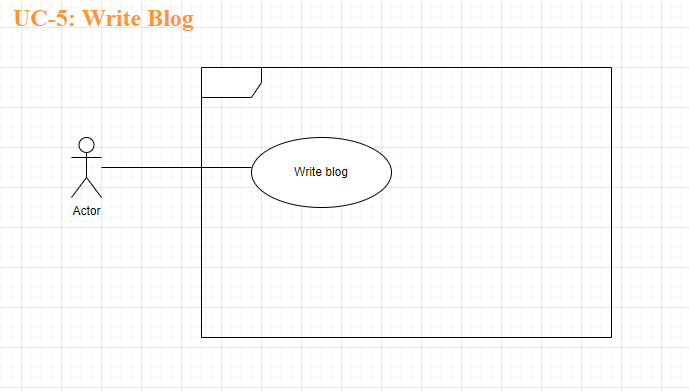


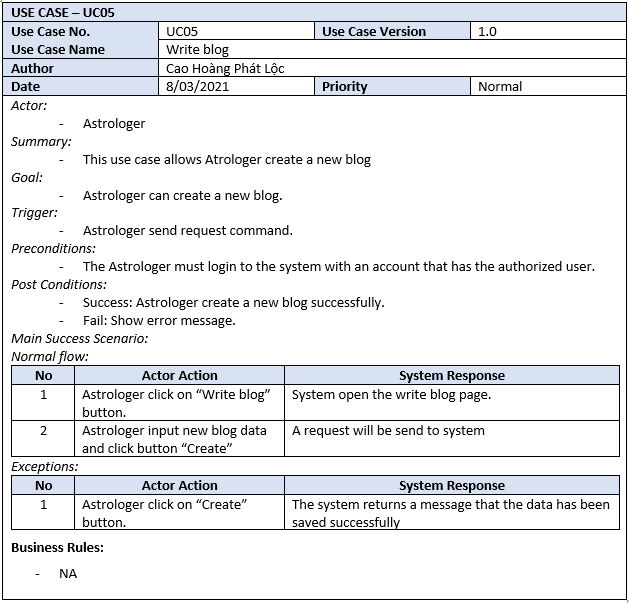


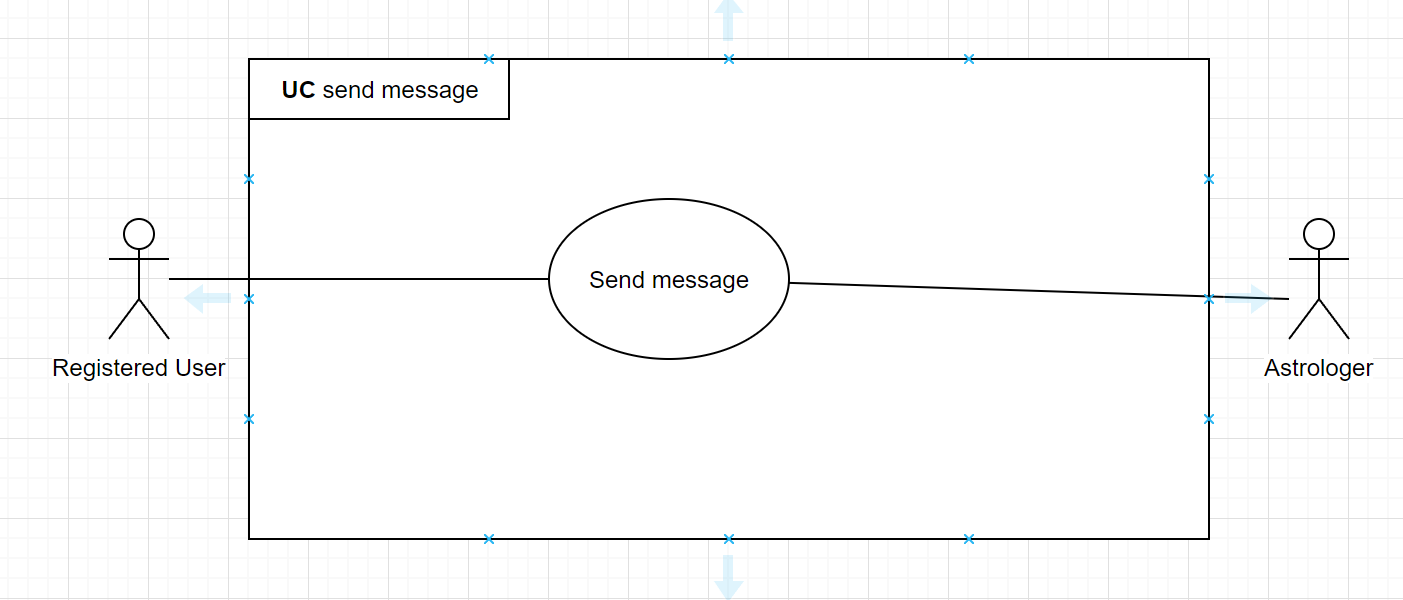




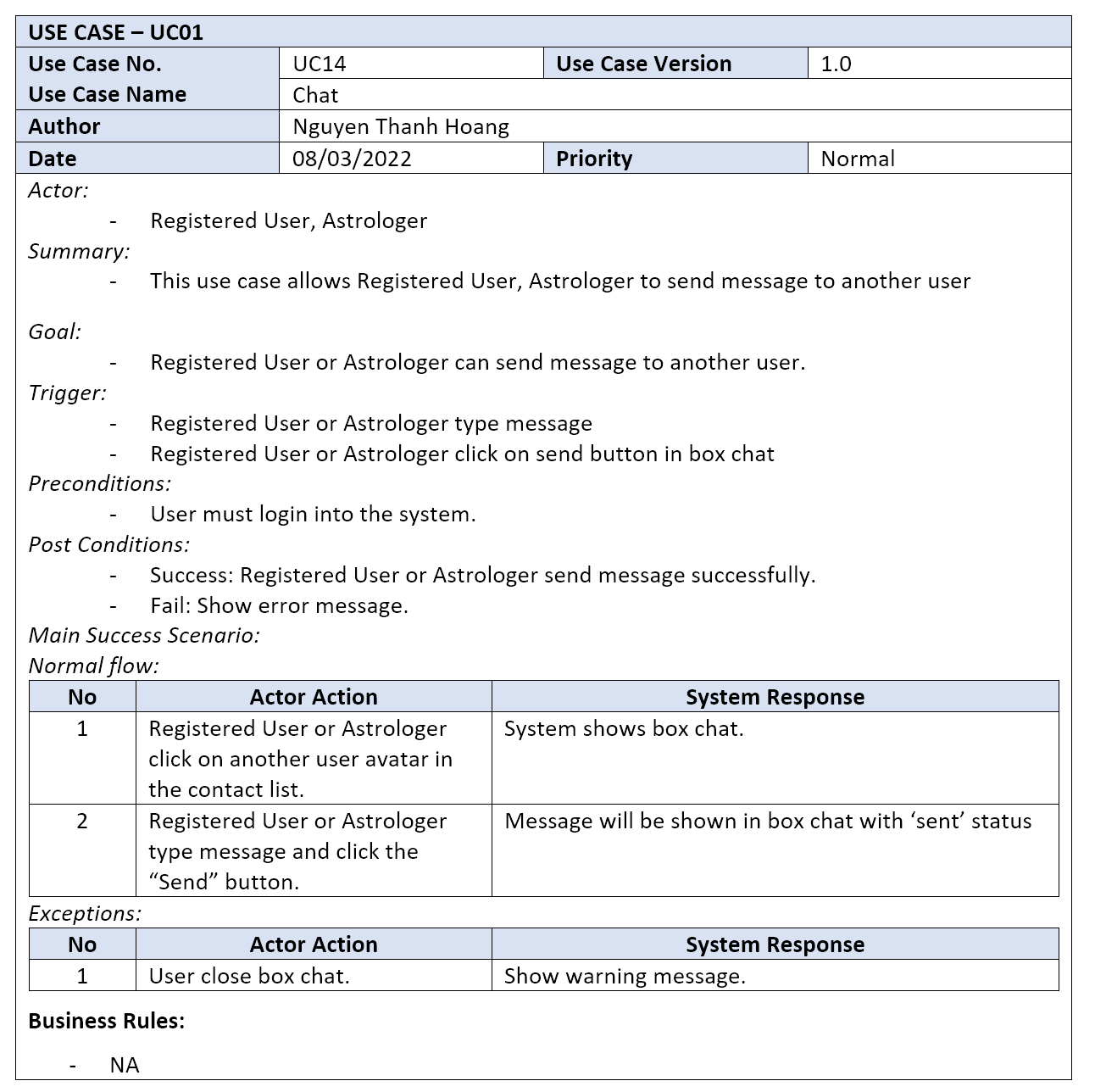


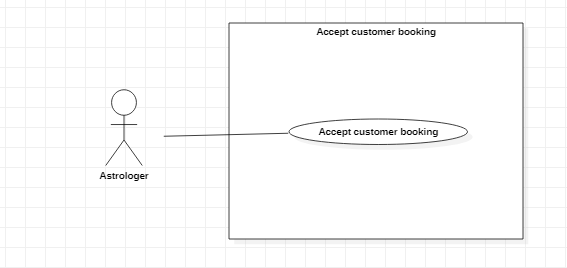


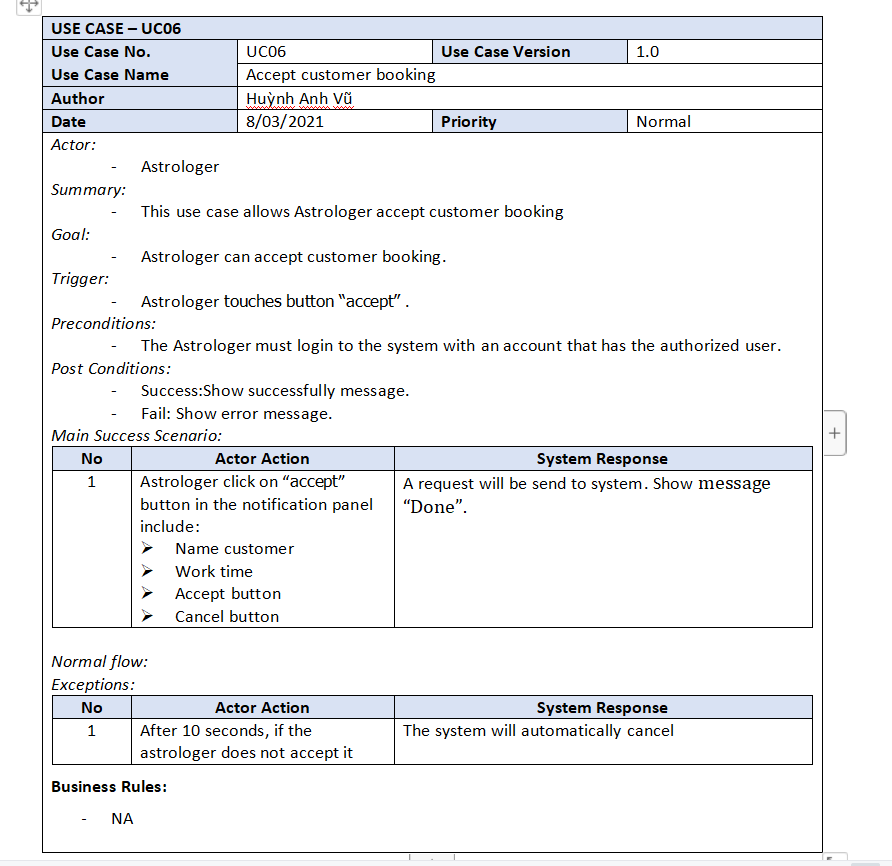




**Use Case Specification**

****





## Logical View

*Show static diagrams: class, object*

## Process View

* 1. *Dynamic diagrams: state, sequence, activity, communication,…  
     Each diagram must have a note to tell which use cases are related*
  2. *If using implicit asynchronous communication styles, must include descriptions for each type of message/event*

## Development View (optional)

*Component diagrams*

## Physical View

*deployment diagrams: server nodes, server connections*

# Design Traceability

| Use case ID | Logical View | Process View | Development View | Physical View |
| --- | --- | --- | --- | --- |
| UC1 | LV2 | PV2 |  |  |
| Uc2 | LV3, 4 | PV1 |  |  |
| UC3 | LV1 | PV3 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |