텍스트이(가) 표시된 사진

자동 생성된 설명

**20194461 신혜진**

**20194538 이나혁**

**20193418 이하윤**

**20192939 이하은**

**20193574 정설희**

**20193878 최은서**

소프트웨어공학

김명호

소프트웨어학과

**-Table of Content**

1. **UC-1**

Scenario Walkthrough

Sequence Diagram

Class Diagram

1. **UC-4**

Scenario Walkthrough

Sequence Diagram

Class Diagram

1. **UC-10**

Scenario Walkthrough

Sequence Diagram

Class Diagram

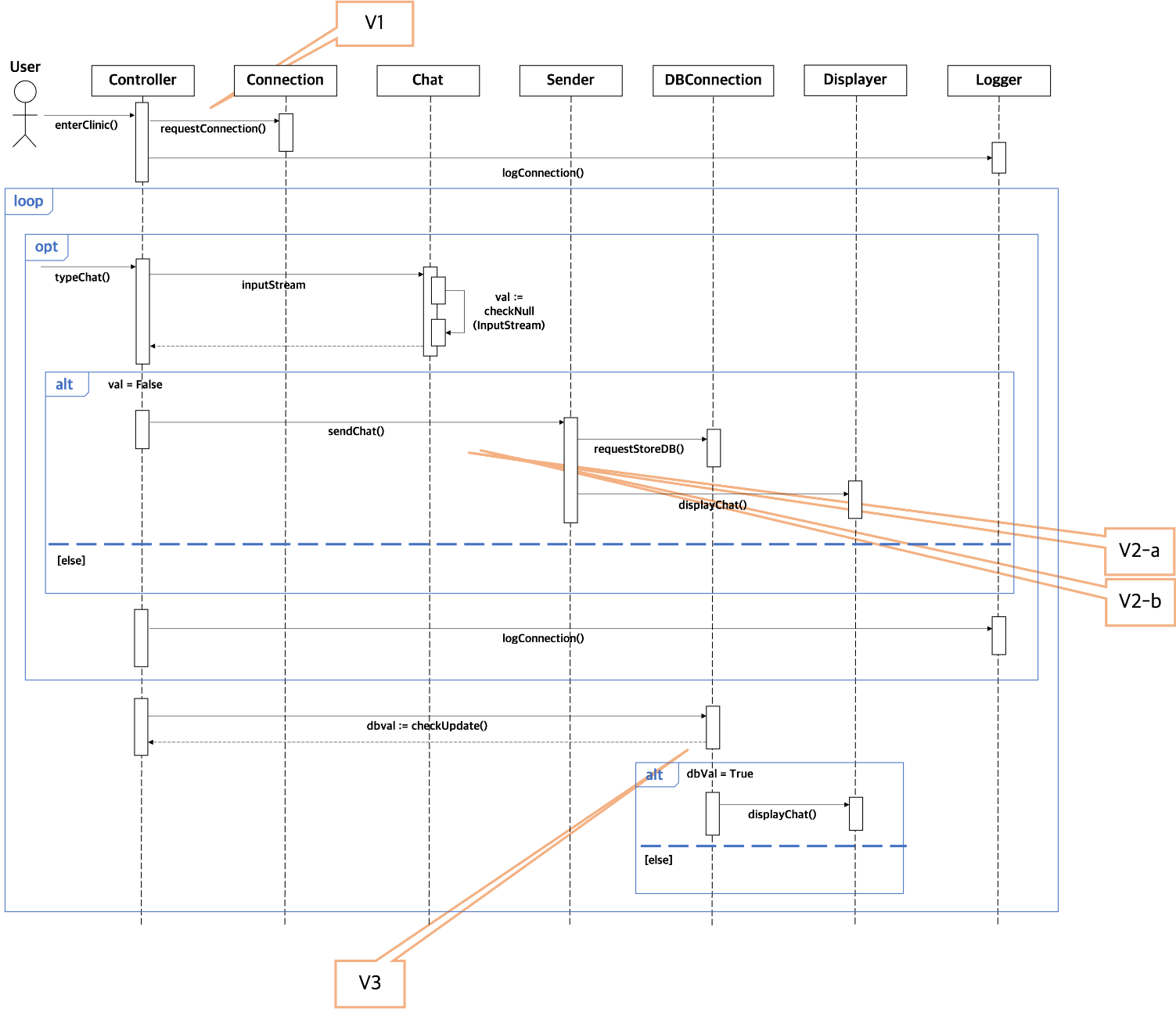
1. **UC-1**

Scenario Walkthrough

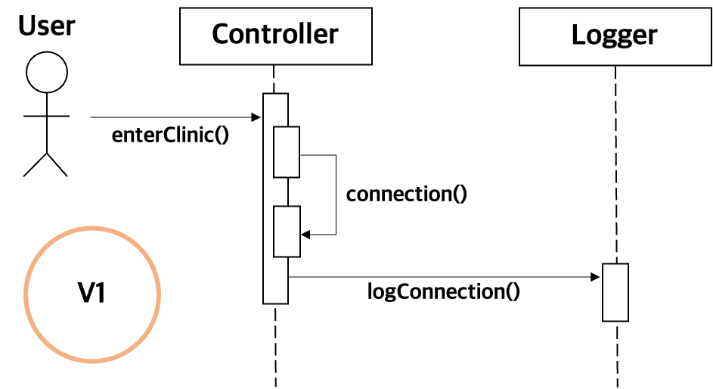
텍스트이(가) 표시된 사진

자동 생성된 설명

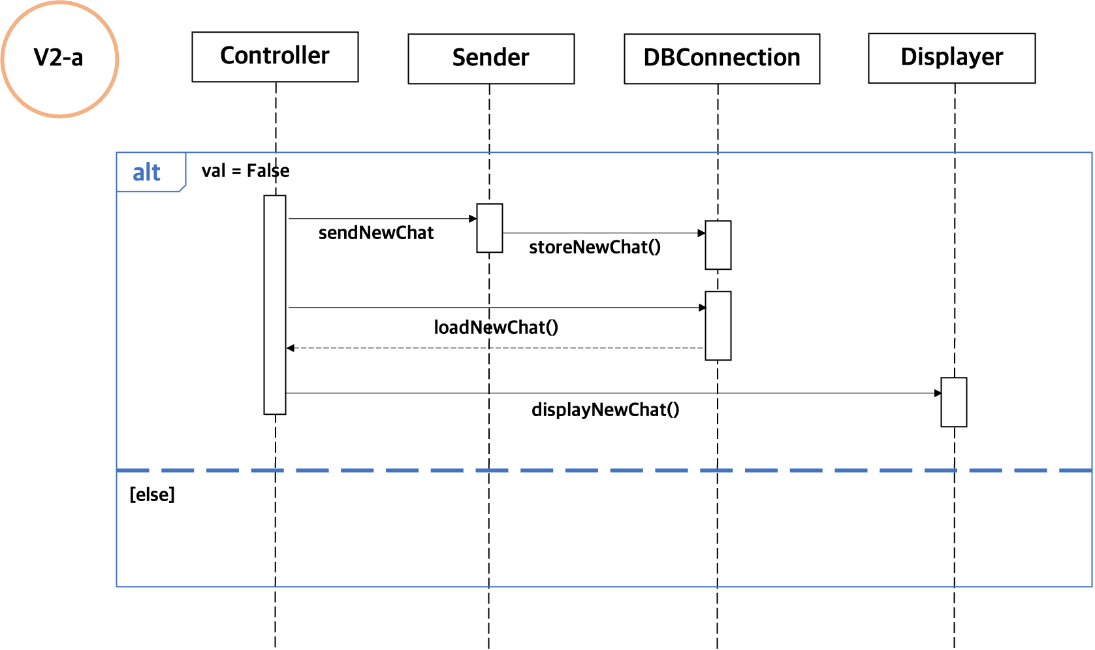
Sequence Diagram



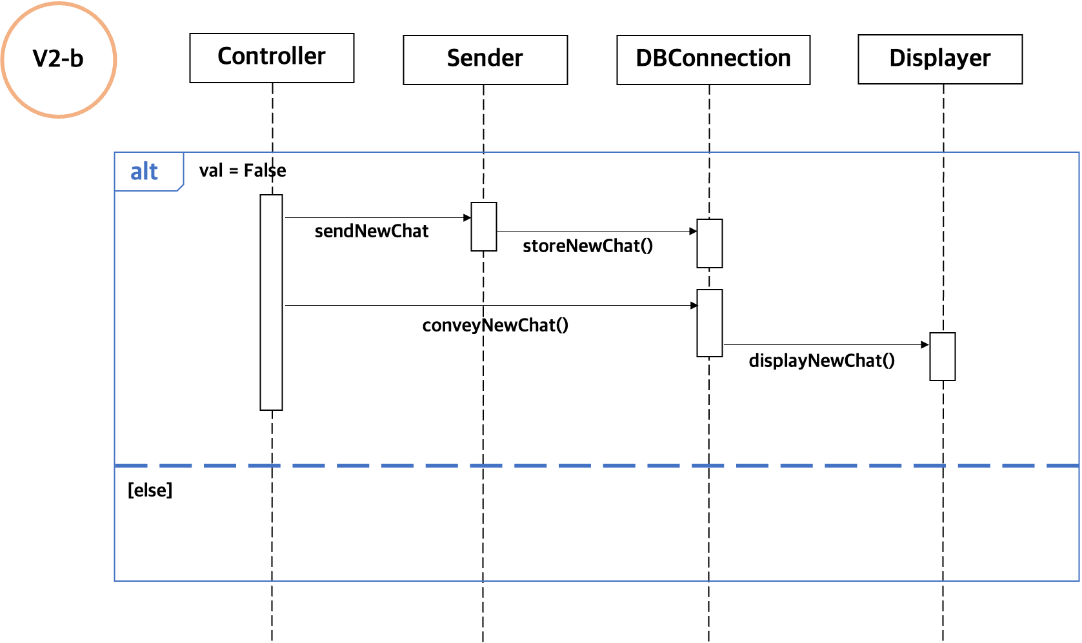
* Variation 1
* Instead of requesting connection to the Connection as a return value for the user’s connection request, the controller directly connects the user.
* **Advantage:** Less coupling(Contorller - Connection), Less Class needed
* **Disadvantage:** Extra responsibility for Controller (low cohesion)



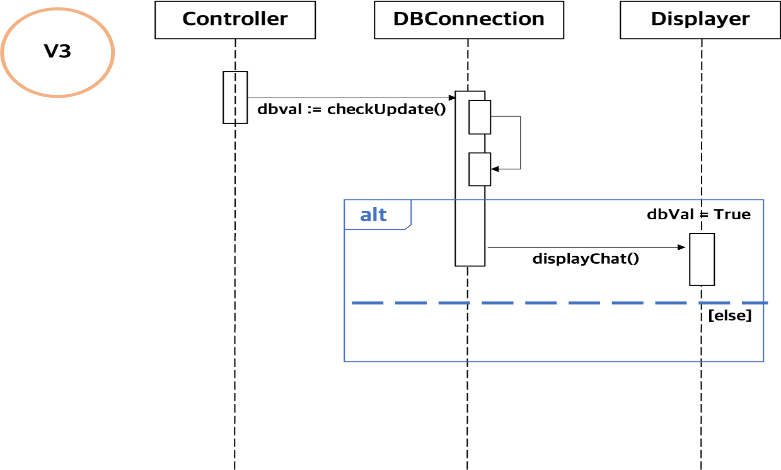
* Variation 2-a
* Instead of returning a request from the Sender to the DBConnection and the display, respectively, for a request to send a chat, the controller plays the role.
* **Advantage:** Clear role(Sender)
* **Disadvantage:** Longer communication chain, Extra responsibility for Controller(low cohesion)



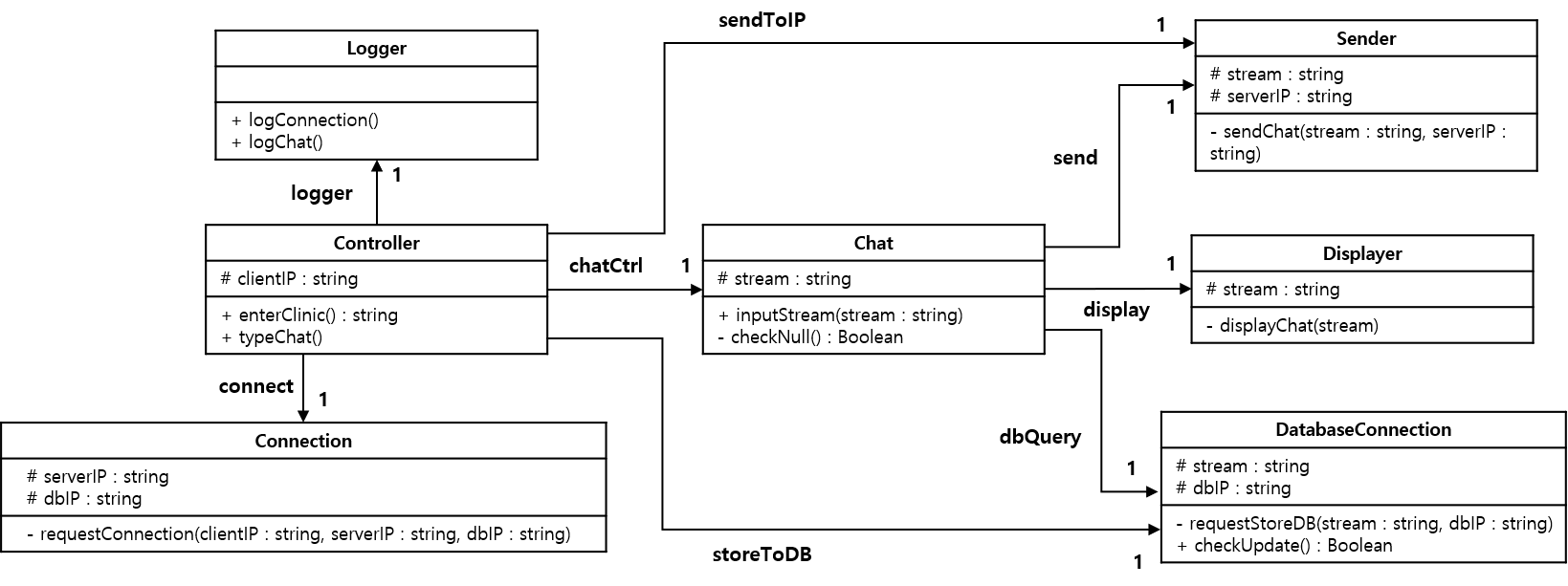
* Variation 2-b
* Instead of the DBConnector returning Chat data, it passes the Chat to the Displayer to request display.
* **Advantage:** Shorter communication chain, less coupling(DBConnector - Controller)
* **Disadvantage:** Extra responsibility for DBConnector(low cohesion)



* Variation 3
* Instead of the DBConnector returning the dbval to the controller, it determines the variable in the DBConnector and requests it directly to the Displayer.
* **Advantage:** Shorter communication chain
* **Disadvantage:** Extra responsibility for DBConnector(low cohesion)



Class Diagram



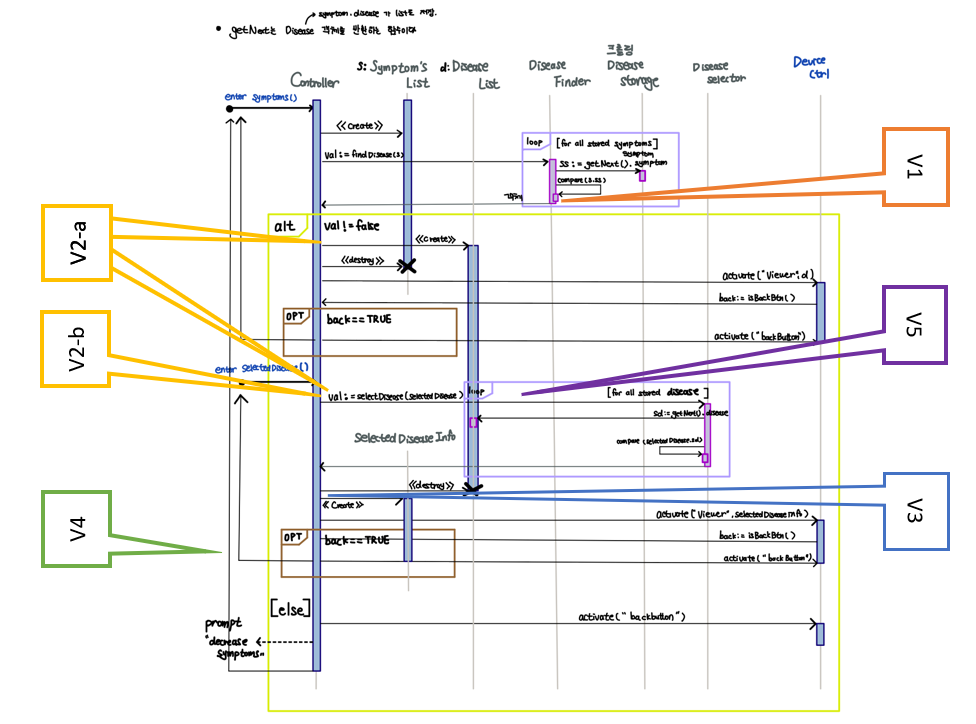
1. UC-4

Scenario Walkthrough

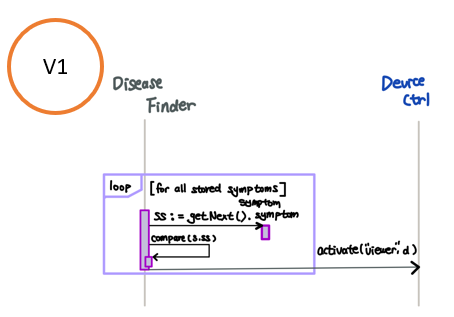
텍스트이(가) 표시된 사진

자동 생성된 설명

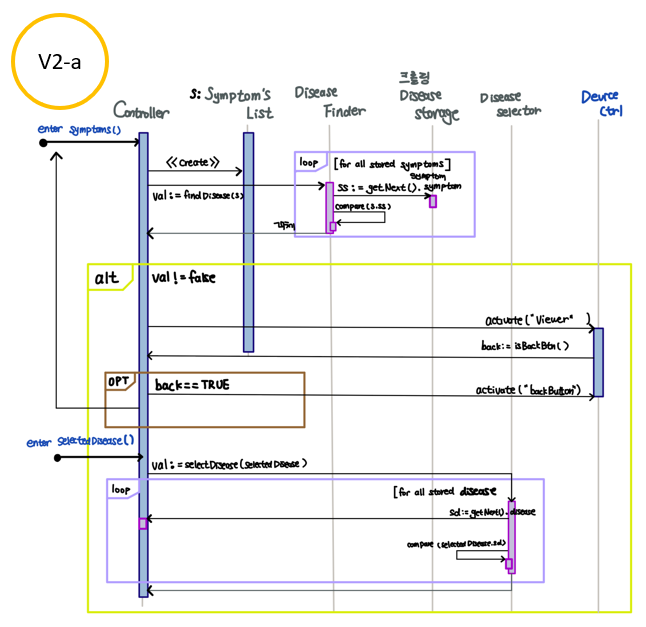
Sequence Diagram



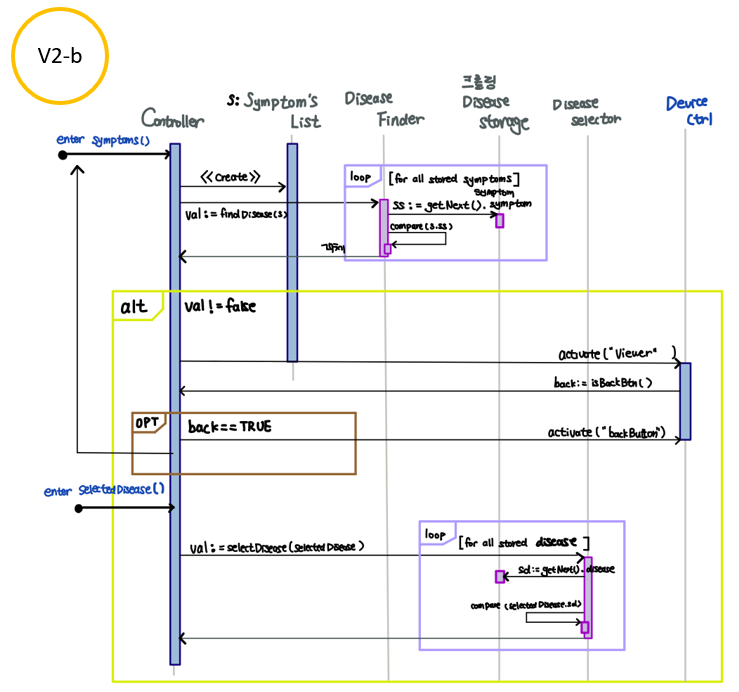
* Variation 1
* Instead of original solution where the controller activates the viewer with the return value of disease finder, diseases finder activates viewer directly with related diseases
* Advantage: Shorter communication chain, less coupling(disease finder - disease list)
* Disadvantage: Extra responsibility for disease finder (low cohesion)



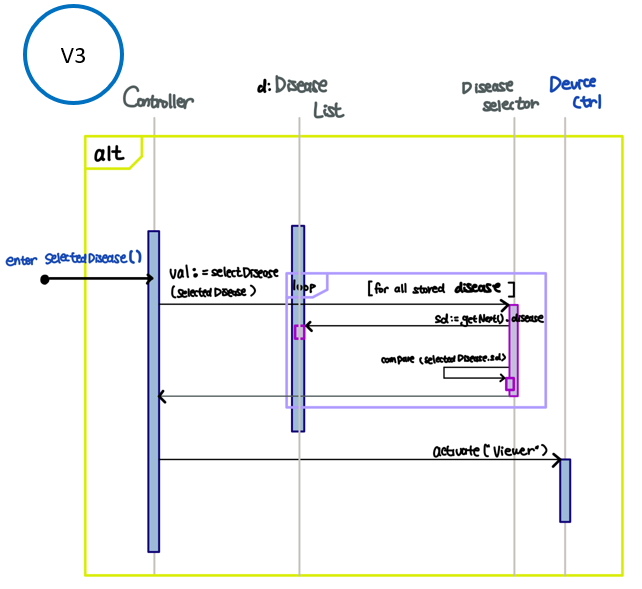
* Variation 2-a
* Instead of original solution where the symptoms-related diseases are stored in disease list, it stores diseases in controller
* Advantage: Less coupling(disease finder - disease list)
* Disadvantage: Extra storage space required for controller, controller should not be involved in device-control details (low cohesion, low specialty)



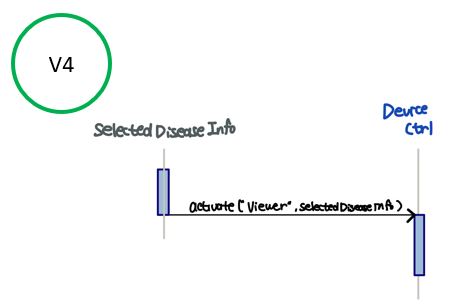
* Variation 2-b
* Instead of original solution where the symptoms-related diseases are stored in disease list, it doesn't store related disease
* Advantage: No extra storage needed, less coupling(disease finder - disease list)
* Disadvantage: When user selects the disease, disease selector should find its information in disease storage instead of disease list and it is not efficient



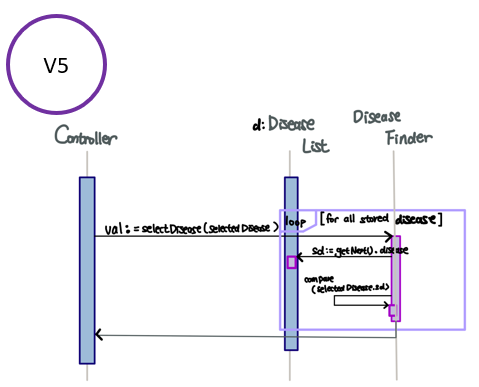
* Variation 3
* Instead of original solution where the user-selected diseases are stored in selected disease info, it stores diseases in controller
* Advantage: Less coupling(disease selector - selected disease info)
* Disadvantage: Extra storage space required for controller, controller should not be involved in device-control details (low cohesion)



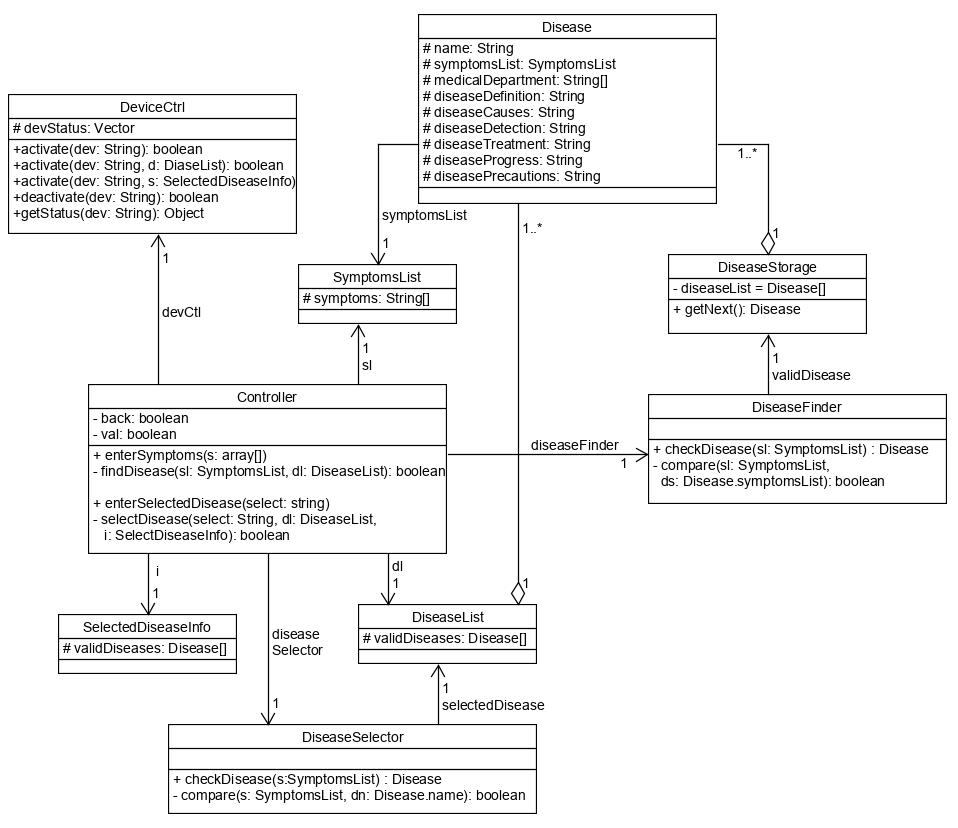
* Variation 4
* Instead of original solution where the controller activates the viewer with the information of selected disease info, diseases selector activates viewer directly
* Advantage: Shorter communication chain, less coupling(disease selector - selected disease info)
* Disadvantage: Extra responsibility for disease selector (low cohesion, low specialty), selected disease info have to be stored anyway



* Variation 5
* Instead of original solution where the disease selector searches diseases, disease finder does that job.
* Advantage: Less class needed
* Disadvantage: Extra responsibility for disease finder (low cohesion, low specialty)



Class Diagram



1. UC-10

Use Cases

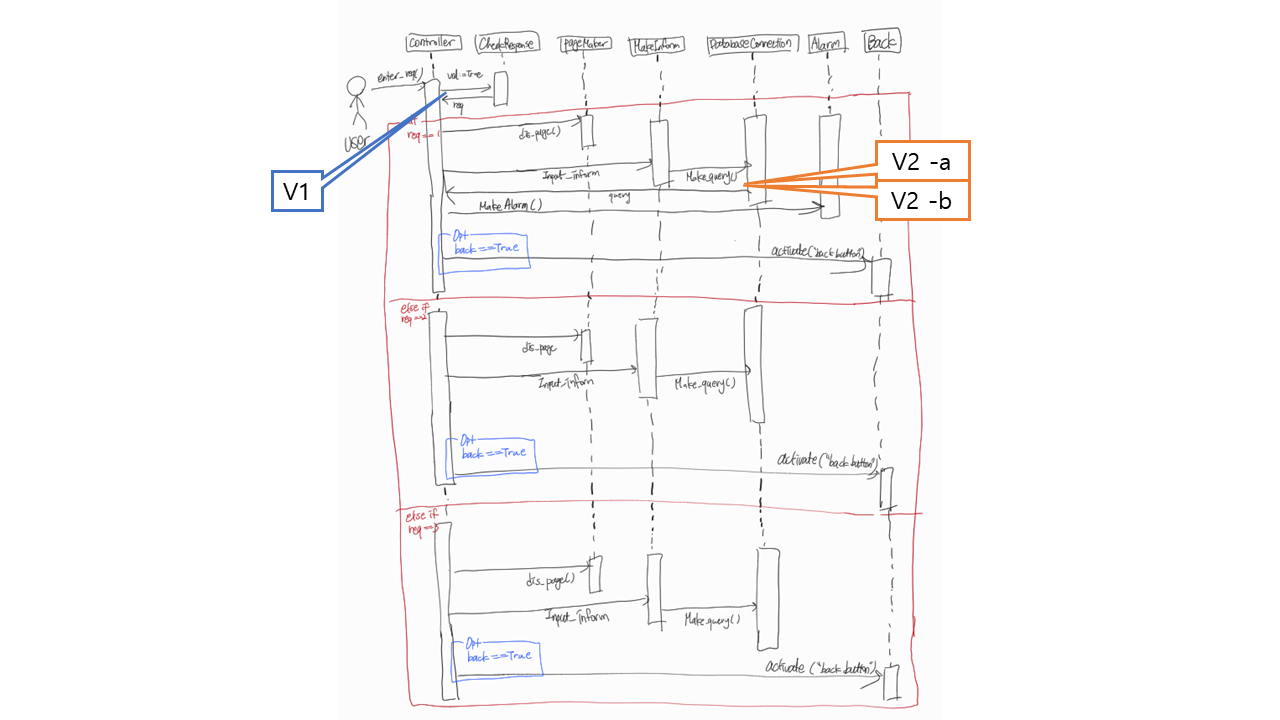
테이블이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

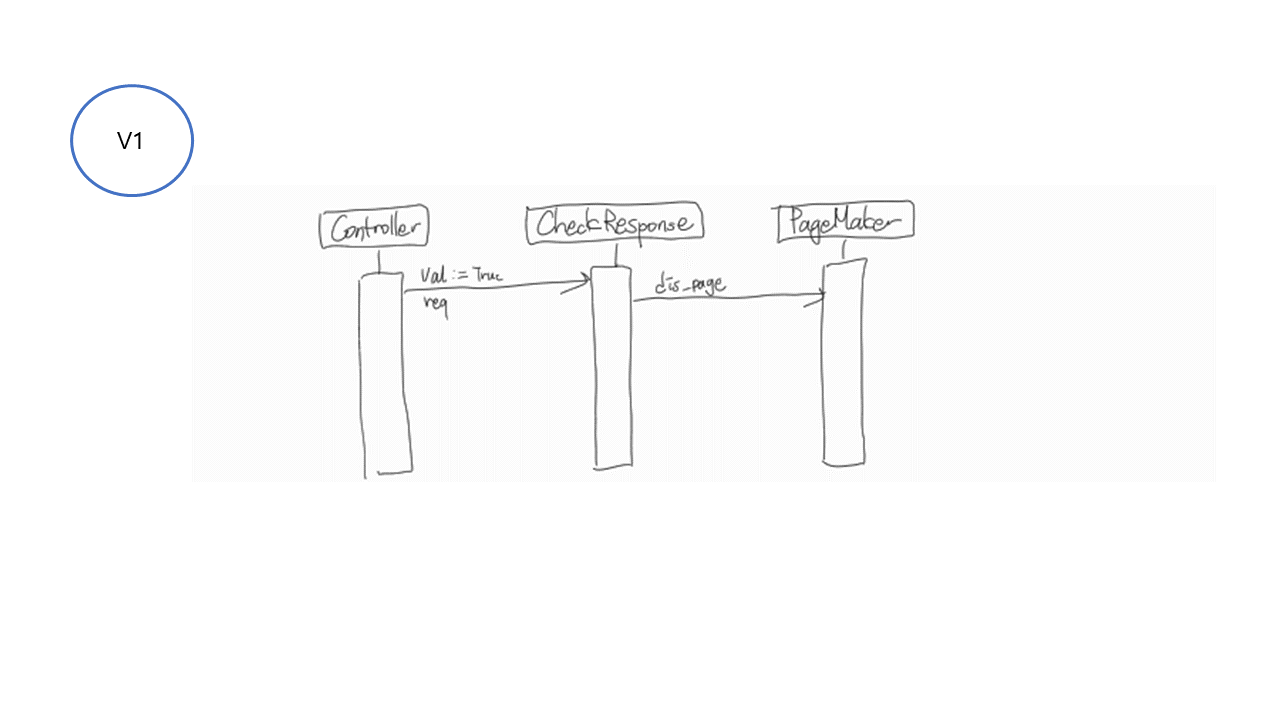
Sequence Diagram



* Variation 1

Instead of CheckResponse sends a req value to Controller and Controller sends a signal to PageMaker depending on the req value, CheckResponse sends a signal directly to PageMaker.

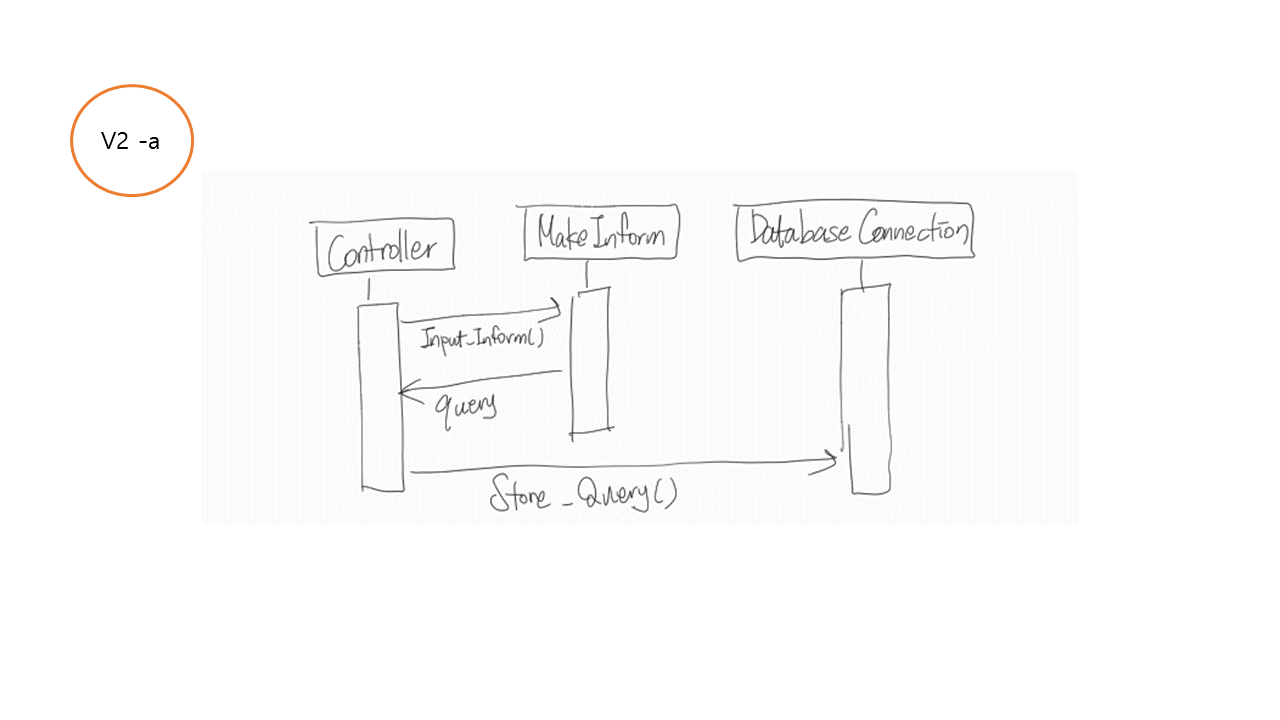
* Advantage : shorter communication chain
* Disadvantage : Extra respoonsibility for CheckResponse



* Variation 2-a

Instead of store query directly from MakeInform to DatabaseConnection, save the query via the Controller.

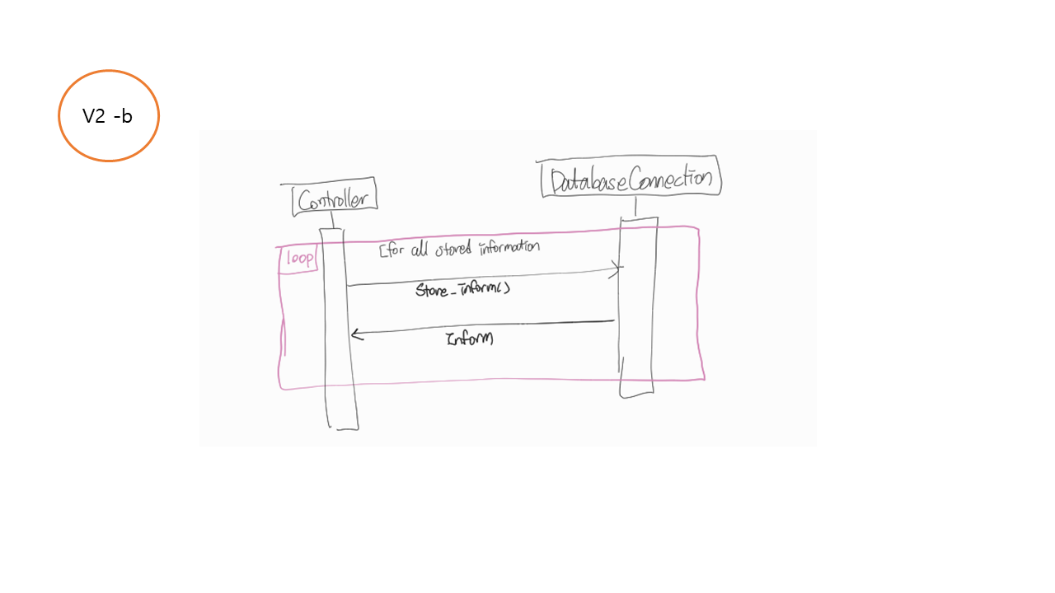
* Advantage : shorter MakeAlarm chain (Controller knows the query)
* Disadvantage : longer query save chain



* Variation 2-b

Instead of converting multiple information into a single query, each information is stored separately.

* Advantage : can skip MakeQuery course
* Disdvantage : send multiple data when referencing data from DB (loop occurs)



Class Diagram

