텍스트이(가) 표시된 사진

자동 생성된 설명

**20194461 신혜진**

**20194538 이나혁**

**20193418 이하윤**

**20192939 이하은**

**20193574 정설희**

**20193878 최은서**

소프트웨어공학

김명호

소프트웨어학과

* Table of Contents

1. Project Subject (p.3)
2. Problem Statement (p.3~5)

* The problem, Scenario

1. Gathered Requirements (p.6~9)

* User stories(Functional and Nonfunctional)

1. System Model (p.10~74)

* Use Case model (w/ Traceability Matrix)
* Domain model (E-C-B) (w/ Traceability Matrix)
* User Interface mockup

1. Project Subject

Smart Health Consulting

1. Problem Statement

-The Problem

*얼마 전부터 머리가 아프고 기침이 납니다. 혹시 제가 심각한 병에 걸린 걸까요? 하지만 저는 병원에 갈 시간적 여유가 없는걸요... 병원을 간다 해도 어느 과를 가야 할지도 잘 모르겠고요. 제 증상이 어떤 병인지 알고 싶고, 혹시 심각한 병일 가능성이 있다면 의사 선생님께 정확한 진단을 받고싶어요. 비대면으로 하면 좋겠지만, 주변에 가까운 병원이 있다면 직접 갈 의향도 있습니다.*

*또, 저희 어머니께서 당뇨가 있으신데, 자꾸 약 먹는 것을 깜빡하셔서 걱정입니다. 저희 어머니가 쉽고 편리하게 약 먹는 시간, 식단, 그리고 운동 등을 관리할 수 있게 옆에서 도와드리고 싶지만, 너무 바빠서 그럴 수 없어 마음이 아픕니다…*

*이런 저와 어머니가 함께 건강한 생활 습관을 지킬 수 있도록 도와주는 서비스를 만들어 주세요 :)*

-Scenario

사용자분께서 저희 웹사이트를 처음 사용하시는 경우, 회원가입을 진행하게 됩니다. 자신의 이름을 비롯한 간단한 개인 정보를 기입해주시면 되는데, 이 때, 일반 사용자와 특별 사용자로 구분됩니다. 현재 지속적으로 케어가 필요한 질환(당뇨, 건망증 등)을 가지고 계신 사용자분의 어머니께서는 특별 사용자로, 위 질환에 해당되지 않는 사용자분께서는 일반 사용자로 등록되어 다른 서비스를 제공받게 됩니다.

사용자 등록을 마친 후, 일반 사용자분께서는 평소 자신의 생활 패턴 (수면시간, 식단, 운동량)등을 기재하고 이에 따라 더욱 질 높은 생활을 하기 위한 생활방식을 추천받게 됩니다. 운동량이 어느정도 부족한지, 식단 구성에서 어떤 부분이 부족하니 어떻게 보충할지 등에 대해서 말이죠. 그리고 특별 사용자분께서는 질환에 따라 다른 서비스가 제공될 것입니다. 만일 사용자분의 어머니와 같이 당뇨병을 앓고 계신다면 인슐린 투여 시간을 알람을 통해 알려주거나, 당뇨병 환자에게 알맞은 식단을 추천하는 등의 서비스가 제공됩니다.

또한 사용자분께서 특정 통증을 겪고 해당 통증에 관한 치료를 원하신다면, 증상을 통해 관련 진료를 받을 수 있는 병원을 추천하는 시스템 또한 제공되고 있습니다. 진료를 받을 수 있는 시스템은 두 가지 방식으로 제공되는데, 비대면 시스템과 대면 시스템이 있습니다. 비대면 시스템의 경우 관련 질병 치료가 가능한 의사와 환자분을 채팅을 통해 직접 연결시켜 굳이 병원에 찾아가지 않더라도 진료가 가능하게합니다. 또한 대면의 경우 환자분의 위치를 중심으로 근처의 어떤 병원이 해당 질병 관련 진료를 하는지 파악 후 그 병원의 위치와 영업 시간을 알려드립니다.

1. Gathered Requirements (user stories)

| **Identifier** | **User Story** | **Size** |
| --- | --- | --- |
| REQ-1 | As patient and doctor, In order to protect the patient's privacy, the chatting should be visible only to the participants in the chatting. | 10 pts |
| REQ-2 | As patient and doctor, chatting must being in real time to provide real-world care experience. | 10 pts |
| REQ-3 | As patient, In order to receive treatment from a real doctor, the doctor must be a registered user in the system. | 8 pts |
| REQ-4 | As patient and doctor, to communicate clearly, chatting must support the input case of Hangul, English, numbers(digits), and special symbols. | 10 pts |
| REQ-5 | As patient and doctor, If a face-to-face diagnosis is required, the doctor must be able to transmit his/her hospital address. | 2 pts |
| REQ-6 | As patient and doctor, transfer button or designated key for transmission must exist for transmission of chattings. | 8 pts |
| REQ-7 | As patient and doctor, there must be designated key that supports changing lines for legible chatting. | 4 pts |
| REQ-8 | As patient and doctor, can refuse to talk to a specific patient or doctor in order to provide customized services. | 3 pts |
| REQ-9 | As patient and doctor, can check past chatting history. | 3 pts |
| REQ-10 | As patient, be able to check the doctor's profile to talk to the doctor of your choice. | 3 pts |
| REQ-11 | As a general/patient user, can choose between general user and patient user when signing up. | 10 pts |
| REQ-12 | As a general/patient user, be able to reposition themselves depending on the situation (general user ↔ patient user) | 1 pts |
| REQ-13 | As a general/paitent user, for daily life management, can record my own life patterns(sleeping patterns, diet, exercise record) | 5 pts |
| REQ-14 | As a general/patient user, can check my past life patterns | 4 pts |
| REQ-15 | As a system, for users to record their own diet, provide food-specific classifications. | 2 pts |
| REQ-16 | As a general/patient user, can record my physical condition through the checklist. | 5 pts |
| REQ-17 | As a general/patient user, the names of all possible diseases associated with my symptoms can be found on the Disease Search page. | 6 pts |
| REQ-18 | As a general/patient user, click on the name of the disease to view the definition, cause, symptoms, diagnosis, treatment, progress, precautions, and medical information of the disease. | 8 pts |
| REQ-19 | The system can retrieve all disease information from the database and filter only the diseases associated with the symptoms. | 16 pts |
| REQ-20 | As a general/patient user, I can choose whether to receive medical treatment for the disease. | 1 pt |
| REQ-21 | As a general/patient user, you can choose whether to receive treatment face-to-face or non-face-to-face. | 1 pt |
| REQ-22 | As a general/patient user, can receive recommendations from nearby clinics based on your current location. | 10 pts |
| REQ-23 | The system can retrieve map information to search for clinics and locations. | 10 pts |
| REQ-24 | As a general/patient user, may be recommended to be a disease-related doctor among doctors who have joined the app. | 6 pts |
| REQ-25 | As a general/patient user, view the recommended doctors' profiles and apply for medical chat. | 4 pts |
| REQ-26 | The system retrieves information from the subscribed doctors from the database, filters it for the department, and sorts it randomly. | 8 pts |
| REQ-27 | As a general/patient user, can go straight to the previous page when you click the Back button. | 1 pt |
| REQ-28 | As a general/patient user, can be recommended symptom analysis through my own checklist. | 3 pts |
| REQ-29 | As a patient user, can register my own disease. | 8 pts |
| REQ-30 | As a paitent user, can record symptoms related to my own disease. | 4 pts |
| REQ-31 | As a patient user, can record actions to manage disease. | 4 pts |
| REQ-32 | As a patient user, can receive special services for the disease (ex. notification of medication administration time, diet recommendations). | 3 pts |
| NFR-1 | Server and database must be designed to be safe from external attacks. | 6 pts |
| NFR-2 | Server and database must be fast enough. | 5 pts |
| NFR-3 | Duplication of DB should be minimized to manage DB efficiently. | 5 pts |
| NFR-4 | Do not take too long to move to next stage | 5 pts |
| NFR-5 | User's personal information should not be leaked | 8 pts |

1. System Model

–Use Case model

Deriving use cases from system requirements

| **Actor** | **Actor's Goal** | **Use Case Name** |
| --- | --- | --- |
| Patient | Write a chat to send to doctor. And also select doctors/conversation which want to show. | UC-1: Clinic UC-2: manageclinic |
| Doctor | Write a chat to send to patient. And also select patients/conversation which want to show. | UC-1: Clinic UC-2: manageclinic |
| Server | Connect patient and doctor in on-line. | UC-1: Clinic |
| Interface | Show the chatting room in Real-Time on the participant's screen. And also some buttons. | UC-1: Clinic UC-2: manageclinic UC-3: matchclinic |
| Database | Save doctor's information, conversation history(chatting). | UC-1: Clinic UC-2: manageclinic UC-3: matchclinic |
| Database | Have blacklist information and update on every request. | UC-3: matchclinic |
| Interface | Have some interface elements such as buttons to add someone in blacklist. | UC-3: matchclinic |
| System | Filter all disease names associated with symptoms | UC-4: searchdieases |
| Patient and general user | Choose a disease user want to see more about | UC-4: searchdieases |
| System | Show definitions, causes, symptoms, diagnosis, treatment, progress, precautions, medical information | UC-4: searchdieases |
| Patient and general user | Choosing whether to get medical treatment | UC-5: findhospital |
| Patient and general user | Choosing whether to get treatment offline or online | UC-5: findhospital |
| Locationsensor | Get user's current location | UC-5: findhospital |
| System | Search and show the hospital based on your location and medical department on the map | UC-5: findhospital |
| System | Filter from doctor's database based on medical department | UC-6: findclinic |
| System | Show profile based on filtered data of doctor's list | UC-6: findclinic |
| Citizen | Sign Up and select type of User | UC-7: adduser |
| System | Provide pages by user type | UC-7: adduser |
| Patient | Enter user's own disease information | UC-7: adduser |
| Doctor | Enter doctor's license and information about hospital | UC-7: adduser |
| System | Check for duplicate ID and doctor license | UC-7: adduser |
| System | Provide Symptom list | UC-8: conditioncheck |
| Citizen | Look up his own physical condition data | UC-8: conditioncheck |
| System | Recommend Serach Disease in the case of suspicion of disease | UC-8: conditioncheck |
| Citizen | Request Search Disease | UC-8: conditioncheck |
| Citizen | Choose one among Sleeping Pattern / Diet / Exercise record | UC-9: life  pattern |
| Citizen | Enter information about Sleeping Pattern / Diet / Exercise Record | UC-9: lifepattern |
| System | Provide User's Life Pattern Grade and improvement | UC-9: lifepattern |
| Patient | Record Symptom / Record Action / Enter notification registration | UC-10:  Disease Management |
| System | Send a notification at the specified time | UC-10:  Disease Management |

Detailed Use case

|  |  |
| --- | --- |
| Use Case UC-1 | Clinic |
| Related Requirement | REQ-2, REQ-4, REQ-5, REQ-6, REQ-7 |
| Initiating Actor | Patient, Doctor |
| Actor’s Goal | * For clear communication |
| Participating Actors | Server, Interface DB |
| Preconditions | * There should be an environment in which the two participants (patient and doctor) can connect through the server. * The two participants must have a device (computer, keyboard, etc.) for chat input. * Approved doctor's information should be stored in the DB. |
| Postconditions | * The string entered by the two participants (patient and doctor) and the transmission time are displayed on the user's screen. * The interfaces (input windows, buttons, etc.) required for chatting are displayed. |
| Flow of events for main success scenario:  → 1. Patient, doctor request a connection.  ← 2. The system connects the two user (patient, doctor) through a server.  → 3. Patient and doctor enter a chat and press the Send button.  ← 4. Displays the text entered by the patient and doctor through the server, and the transmission time on their screen. | |

|  |  |
| --- | --- |
| Use Case UC-2 | ManageClinic |
| Related Requirement | REQ-1, REQ-9 |
| Initiating Actor | Patient, Doctor |
| Actor’s Goal | * To check and protect existing chat |
| Participating Actors | DB, System |
| Preconditions | * Requires a DB that can store chat history. * DB's chat history should only be accessible to its participants. |
| Postconditions | Patient and doctor are given read access to the chat and can save the chat or view it. |
| Flow of events for main success scenario:  → Users access past chat records.  ← System verifies the user's rights and provides a list of past chat records.  → Users access the desired chat.  ← System provides a chat history for that chat. | |

|  |  |
| --- | --- |
| Use Case UC-2 | Match Clinic |
| Related Requirement | REQ-3, REQ-8, REQ-10 |
| Initiating Actor | Patient, Doctor |
| Actor’s Goal | * To provide reliable matching to patients |
| Participating Actors | DB, Interface |
| Preconditions | * All doctors should be registered in the system's DB. * All doctors should have a profile. |
| Postconditions | * Patients can receive customized services by checking and selecting their doctor's profile |
| Flow of events for main success scenario  ← System shows the user a list of doctors.  → Patient check the doctor's profile on the list and select the doctor they want to consult.  ← System provides the patient with a button to refuse to talk to a particular doctor.  → Patients can click the button to refuse to talk to doctor.  ← Blacklist information is stored in the DB. | |

|  |  |
| --- | --- |
| Use Case UC-4 | Search Disease |
| Related Requirement | REQ-17, REQ-18, REQ-19, REQ-27 |
| Initiating Actor | Patient, General User |
| Actor’s Goal | * To provide symptoms to search diseases. * To choose one of the disease and get information(정의, 원인, 증상, 진단, 치료, 경과, 주의사항, 진료과) about it. |
| Participating Actors | System |
| Preconditions | * The initial value of the symptoms is given by the citizen’s checklist. * The list of diseases for each symptoms is generated by the DB created by Web Crawling on 아산병원 website |
| Postconditions | Update Medical department into DB |
| Flow of events for main success scenario:  <- 1. Initialize the value of the symptoms to find a list of diseases.  <- 2. Show the list of diseases  -> 3. Citizen selects one of the disease in the diseases list  <- 4. Citizen can view the disease information(정의, 원인, 증상, 진단, 치료, 경과, 주의사항, 진료과)  <- 5. Add 진료과 to citizen’s DB. | |

|  |  |
| --- | --- |
| Use Case UC-5 | FindHospital |
| Related Requirement | REQ-20, REQ-21,REQ-22, REQ-23, REQ-27 |
| Initiating Actor | Patient, General User |
| Actor’s Goal | * To click button with mouse and choose to get treatment * To click button with mouse and choose to find clinic offline * To provide current location and view nearby hospital recommendation based on location and symptom’s department info |
| Participating Actors | Location sensor, Map api |
| Preconditions | * Citizen agreed to provide current location info * Location sensor is available * The current medical department about the disease is stored in DB * System has map information |
| Postconditions |  |
| Flow of events for main success scenario  → 1. Citizen chooses to get treatment button  → 2. Citizen chooses to find clinic offline  → 3. Citizen agrees to provide current location  → 4. Location sensor gets citizen’s location      5. System searches nearby hospital based on current location and medical department info  ← 6. Viewer shows nearby hospital recommendation | |

|  |  |
| --- | --- |
| Use Case UC-6 | FindClinic |
| Related Requirement | REQ-20,REQ-21,REQ-24, REQ-25, REQ-26, REQ-27 |
| Initiating Actor | Patient, General User |
| Actor’s Goal | * To click button to get treatment * To click button to get clinic from doctor online * To click button and view doctor’ profiles related to current symptom’s department info * To select doctor and apply for medical treatment chat online |
| Participating Actors | Doctor database |
| Preconditions | * The list of all registered doctors and their information is provided by the DB * The medical department info of the symptom is current stored in DB |
| Postconditions | Store medical treatment application information into DB |
| Flow of events for main success scenario  → 1 .Citizen selects get treatment button  → 2. Citizen selects find doctor online button  3. System filters doctors in doctor DB based on the symptom’s medical department information  ← 4. Viewer shows the profiles of filtered doctor | |

|  |  |
| --- | --- |
| Use Case UC-7 | AddUser |
| Related Requirement | REQ-11, REQ-29 |
| Initiating Actor | Patient, General User,Doctor |
| Actor’s Goal | * User의 ID / password 및 개인 정보(이름, 나이, 연락처, 성별 등) 서버에 등록 * User를 general user과 patient user 로 구분 - patient user 일 시, User의 own disease 서버에 등록 * Doctor user 일 시, 의사 여부 확인 및 병원 등록 - Register user's ID / password and personal information (name, age, contact information, gender, etc.) on the server * Classify users into general users and patient users * For patient users, register the user's own disease on the server * For doctor users, check doctor license and register hospital information |
| Participating Actors | DB, server |
| Preconditions | * 다른 User들과 ID 값 달라야 함 * 특정 질병 리스트 DB에 등록되어 있어야함 * ID value must not be duplicated with other users. * Specific disease list must be registered in DB |
| Postconditions | * 해당 User의 정보 DB에 등록 * Register the user's information in the DB |
| Flow of events for main success scenario:  → 1. User가 회원가입 버튼 클릭  ← 2. 회원가입 창 보이기  → 3. User가 ID / password 및 개인정보(이름, 나이, 연락처, 성별) 입력  ← 4. 기존 User들과 ID값 같으면 ID 재입력 요구 메세지 보내기  → 5. User에게 general User / patient user / doctor 중 선택 받기  → 6. patient User시 user의 own disease 정보 입력받기  → 7. Doctor시 병원 및 의사 정보 입력받기  ← 8. User의 정보 DB에 등록  → 9. User clicks the sign up button  ← 10. Show sign up page  → 11. User enters ID / password and personal information (name, age, contact, gender)  ← 12. Send ID re-input request message if the ID value already exists in the DB  → 13. User selects between general user / patient user / doctor  → 14. Patient User enters his own disease information  → 15 Doctor enters his own doctor license and hospital information  ← 16. Register user information in DB | |

|  |  |
| --- | --- |
| Use Case UC-8 | ConditionCheck |
| Related Requirement | REQ-16, REQ-28 |
| Initiating Actor | Patient, General User |
| Actor’s Goal | * 일상 생활 속 physical condition check * physical condition을 토대로 질병 의심시 Search Disease 추천 * Physical condition check in daily life * Recommend Search Disease in case of suspected disease based on physical condition |
| Participating Actors | DB |
| Preconditions | * 여러 질병에 관한 증상 정보가 충분히 등록되어 있어야함 * Symptom information on various diseases must be sufficiently registered |
| Postconditions | User가 동의할 시 Search Disease로 연결  - Link to Search Disease when user agrees |
| Flow of events for main success scenario  ← 1. symptom 리스트 제공 (두통, 복통 등등)  → 2. User가 수시로 자신의 physical condition 을 등록  → 3. User가 지난 자신의 physical condition 확인 요청  ← 4. User에게 his own physical condition data 보여주기  ← 5. physical condition 분석 후 질병 의심시 Search Disease 받을 것을 추천  → 6. User가 Search Disease 받는다고 요청  ← 7. User를 Search Disease로 연결 ([Use Case UC-4](https://www.notion.so/Use-Case-UC-4-3b448dfa06234f00adc5bf24c8ebd43b))  ← 8. Provide a list of symptoms (headache, colic etc)  → 9. User enters his own physical condition from time to time  → 10. User requests to confirm his past physical condition  ← 11. Show user his own physical condition data  ← 12. Recommend user to receive Search Disease in case of suspected disease based on physical condition  → 13. User requests to receive Search Disease  ← 14. Link User to Search Disease ( [Use Case UC-4](https://www.notion.so/Use-Case-UC-4-3b448dfa06234f00adc5bf24c8ebd43b)) | |

|  |  |
| --- | --- |
| Use Case UC-9 | LifePattern |
| Related Requirement | REQ-13, REQ-14,REQ-15 |
| Initiating Actor | Patient, General User |
| Actor’s Goal | * User가 Good Life Pattern 을 가질 수 있도록 도움 * Help users to have a good life pattern |
| Participating Actors | DB |
| Preconditions | * Sleeping Pattern, diet, exercise record의 좋고 나쁨의 기준이 세워져 있어야함   - Standards of good and bad for sleeping pattern, diet, and exercise record should be established. |
| Postconditions | * 기준과 비교해 User가 Good Life Pattern을 가지고 있는지 보여줌.   - User에게 그들의 Life Pattern의 개선점을 알려줌  - Shows whether the user has a good life pattern compared to the standard  - Notifying user of improvements in his life pattern |
| Flow of events for main success scenario  → 1. User가 Life Pattern 입력 요청  ← 2. User 에게 Sleeping Pattern / Diet / Exercise Record 중 무엇을 입력 받을 것인지 선택창  → 3. User가 Sleeping Pattern / Diet / Exercise Record 중 선택  → 4. User의 Life Pattern 입력  ← 5. DB에 User의 Life Pattern 저장  ← 6. User의 Life Pattern과 기준을 비교하여, 해당 User의 Life Pattern Grade및 개선점 보여줌  → 7. User requests to enter Life Pattern  ← 8. Display a page to select among Sleeping Pattern / Diet / Exercise Record  → 9. User selects one of Sleeping Pattern / Diet / Exercise Record  → 10. User inputs his life pattern  ← 11. Save User's Life Pattern to DB  ← 12. Shows whether the user has a good life pattern compared to the standard and improvements in his life pattern | |

|  |  |
| --- | --- |
| Use Case UC-10 | Disease Management |
| Related Requirement | REQ-12, REQ-29, REQ-30, REQ-31, REQ-32 |
| Initiating Actor | Patient |
| Actor’s Goal | * Patient가 자신의 질병을 관리 할 수 있도록 help   - Help patient to manage his own diseases |
| Participating Actors | DB |
| Preconditions | * Patient의 질병이 DB에 등록되어 있어야 함   - 특정 질병에 대한 special service list가 있어야함 (ex. notification of medication administration time, diet recommendations)  - Patient's disease must be registered in DB  - There must be a special service list for specific diseases (ex. notification of medication administration time, diet recommendations) |
| Postconditions | * DB에 patient의 disease symptom data 등록하기   - DB에 patient의 action to manage disease data 등록하기  - DB에 알림 설정 등록하기  - 지정된 시간에 알림 보내기  - Register patient's disease symptom data in DB  - Register patient's action to manage disease data in DB  - Registering notification settings in DB -Send notification at specified time |
| Flow of events for main success scenario  → 1. Patient가 symptom 기록 요청  ← 2. symptom record 창 띄우기  → 3. 어떤 종류의 symptom인지 / symptom의 강도는 어떠한 지 입력 받기  ← 4. DB에 symptom record 저장  → 5. Patient user requests to record his symptom  ← 6. Display symptom record page  → 7. User enters the type and intensity of symptoms  → 8. Patient가 action to manage disease 기록 요청  ← 9. action record 창 띄우기  → 10. action 종류 / 시간 입력 받기  ← 11. DB에 action record 저장  → 12. Patient user requests to record action to manage diseases  ← 13. Display action record page  → 14. User enters action type and time  ← 15. Save action record in DB  → 16. Patient가 알림 등록 요청  ← 17. 알림 등록 창 띄우기  → 18. 어떤 시간에, 어떤 행위에 대한 알림 받을 건지 등록 받기  ← 19. DB에 알림 관련 정보 저장  ← 20. 지정 시간 되면 Patient에게 알림 보내기  → 21. Patients requests notification registration  ← 22. Display notification registration page  → 23. User enters the time and type of notification  ← 24. Save notification information in DB  ← 25. Send a notification to user at the specified time | |

–Traceability Matrix

System requirements를 Use Cases에 Mapping

| **REQ** | **PW** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **UC6** | **UC7** | **UC8** | **UC9** | **UC10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 10 |  | X |  |  |  |  |  |  |  |  |
| 2 | 10 | X |  |  |  |  |  |  |  |  |  |
| 3 | 8 |  |  | X |  |  |  |  |  |  |  |
| 4 | 10 | X |  |  |  |  |  |  |  |  |  |
| 5 | 2 | X |  |  |  |  |  |  |  |  |  |
| 6 | 8 | X |  |  |  |  |  |  |  |  |  |
| 7 | 4 | X |  |  |  |  |  |  |  |  |  |
| 8 | 3 |  |  | X |  |  |  |  |  |  |  |
| 9 | 3 |  | X |  |  |  |  |  |  |  |  |
| 10 | 3 |  |  | X |  |  |  |  |  |  |  |
| 11 | 10 |  |  |  |  |  |  | X |  |  |  |
| 12 | 1 |  |  |  |  |  |  |  |  |  | X |
| 13 | 5 |  |  |  |  |  |  |  |  | X |  |
| 14 | 4 |  |  |  |  |  |  |  |  | X |  |
| 15 | 2 |  |  |  |  |  |  |  |  | X |  |
| 16 | 5 |  |  |  |  |  |  |  | X |  |  |
| 17 | 6 |  |  |  | X |  |  |  |  |  |  |
| 18 | 5 |  |  |  | X |  |  |  |  |  |  |
| 19 | 10 |  |  |  | X |  |  |  |  |  |  |
| 20 | 1 |  |  |  |  | X | X |  |  |  |  |
| 21 | 1 |  |  |  |  | X | X |  |  |  |  |
| 22 | 7 |  |  |  |  | X |  |  |  |  |  |
| 23 | 7 |  |  |  |  | X |  |  |  |  |  |
| 24 | 5 |  |  |  |  |  | X |  |  |  |  |
| 25 | 3 |  |  |  |  |  | X |  |  |  |  |
| 26 | 8 |  |  |  |  |  | X |  |  |  |  |
| 27 | 1 |  |  |  | X | X | X |  |  |  |  |
| 28 | 3 |  |  |  |  |  |  |  | X |  |  |
| 29 | 8 |  |  |  |  |  |  | X |  |  | X |
| 30 | 4 |  |  |  |  |  |  |  |  |  | X |
| 31 | 4 |  |  |  |  |  |  |  |  |  | X |
| 32 | 3 |  |  |  |  |  |  |  |  |  | X |
| **Max PW** | **--** | **10** | **10** | **8** | **10** | **7** | **8** | **10** | **5** | **5** | **8** |
| **Total PW** | **--** | **34** | **13** | **14** | **22** | **17** | **19** | **18** | **8** | **11** | **20** |

Use Cases를 Domain Model에 Mapping

|  |
| --- |
|  |
| **Use Case** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **UC6** | **UC7** | **UC8** | **UC9** | **UC10** |
| **PW** | **34** | **13** | **14** | **22** | **17** | **19** | **18** | **8** | **11** | **20** |
| Controller-SS1 | X |  |  |  |  |  |  |  |  |  |
| Request | X |  |  |  |  |  |  |  |  |  |
| ConnectOperator | X |  |  |  |  |  |  |  |  |  |
| DatabaseConnection | X | X | X |  |  |  | X | X | X | X |
| Chat | X | X | X |  |  |  |  |  |  |  |
| SendOperator | X |  |  |  |  |  |  |  |  |  |
| DisplayOperator | X | X |  |  |  |  |  |  |  |  |
| AlarmOperator | X |  |  |  |  |  |  |  |  |  |
| Logger | X |  | X |  |  |  |  |  |  |  |
| Controller-SS2 |  | X |  |  |  |  |  |  |  |  |
| CheckAuth |  | X |  |  |  |  |  |  |  |  |
| ConversationList |  | X |  |  |  |  |  |  |  |  |
| SelectConversation |  | X |  |  |  |  |  |  |  |  |
| Controller-SS3 |  |  | X |  |  |  |  |  |  |  |
| Doctor | X | X | X |  |  | X |  |  |  |  |
| CheckDoctor |  |  | X |  |  |  |  |  |  |  |
| SelectDoctor |  |  | X |  |  |  |  |  |  |  |
| Blacklist |  |  | X |  |  |  |  |  |  |  |
| BlacklistOperator |  |  | X |  |  |  |  |  |  |  |
| Controller-SS7 |  |  |  |  |  |  | X |  |  |  |
| CheckSignUp |  |  |  |  |  |  | X |  |  |  |
| PageMaker |  |  |  |  |  |  | X | X | X | X |
| MakeInform |  |  |  |  |  |  | X | X |  | X |
| CheckIDRedundancy |  |  |  |  |  |  | X |  |  |  |
| CheckDoctor |  |  |  |  |  |  | X |  |  |  |
| Controller-SS8 |  |  |  |  |  |  |  | X |  |  |
| InterfaceSymptomlist |  |  |  |  |  |  |  | X |  |  |
| CheckPhysical |  |  |  |  |  |  |  | X |  |  |
| DiseaseOperator |  |  |  |  |  |  |  | X |  |  |
| CheckResponse |  |  |  |  |  |  |  | X |  |  |
| Controller-SS8 |  |  |  |  |  |  |  |  | X |  |
| CheckLifePattern |  |  |  |  |  |  |  |  | X |  |
| DataCollectOperator |  |  |  |  |  |  |  |  | X |  |
| CheckAttribute |  |  |  |  |  |  |  |  | X |  |
| CompareWithStandard |  |  |  |  |  |  |  |  | X |  |
| GradeOperator |  |  |  |  |  |  |  |  | X |  |
| ImprovementOperator |  |  |  |  |  |  |  |  | X |  |
| Controller-SS10 |  |  |  |  |  |  |  |  |  | X |
| CheckResponse |  |  |  |  |  |  |  |  |  | X |
| Alarm |  |  |  |  |  |  |  |  |  | X |
| Controller-SS11 |  |  |  | X |  |  |  |  |  |  |
| DiseaseStorage |  |  |  | X |  |  |  |  |  |  |
| SymptomsList |  |  |  | X |  |  |  |  |  |  |
| DiseaseFinder |  |  |  | X |  |  |  |  |  |  |
| Viewer |  |  |  | X | X | X |  |  |  |  |
| DiseaseSelection |  |  |  | X |  | X |  |  |  |  |
| BackButton | X | X | X | X | X | X | X | X | X | X |
| Controller-SS12 |  |  |  |  | X |  |  |  |  |  |
| Controller-SS13 |  |  |  |  |  | X |  |  |  |  |
| ButtonPopup |  |  |  |  | X | X |  |  |  |  |
| UserSelection |  |  |  |  | X | X |  |  |  |  |
| LocationNotify |  |  |  |  | X |  |  |  |  |  |
| UserAgreement |  |  |  |  | X |  |  |  |  |  |
| GetLocation |  |  |  |  | X |  |  |  |  |  |
| UserLocation |  |  |  |  | X |  |  |  |  |  |
| FindNearHospital |  |  |  |  | X |  |  |  |  |  |
| MouseEntry |  |  |  |  | X |  |  |  |  |  |
| MapViewer |  |  |  |  | X |  |  |  |  |  |
| DoctorFinder |  |  |  |  |  | X |  |  |  |  |
| SelectedDoctor |  |  |  |  |  | X |  |  |  |  |

* Domain model (E-C-B) (w/ Traceability Matrix)
* Domain model for UC-1

1. Extracting responsibilities

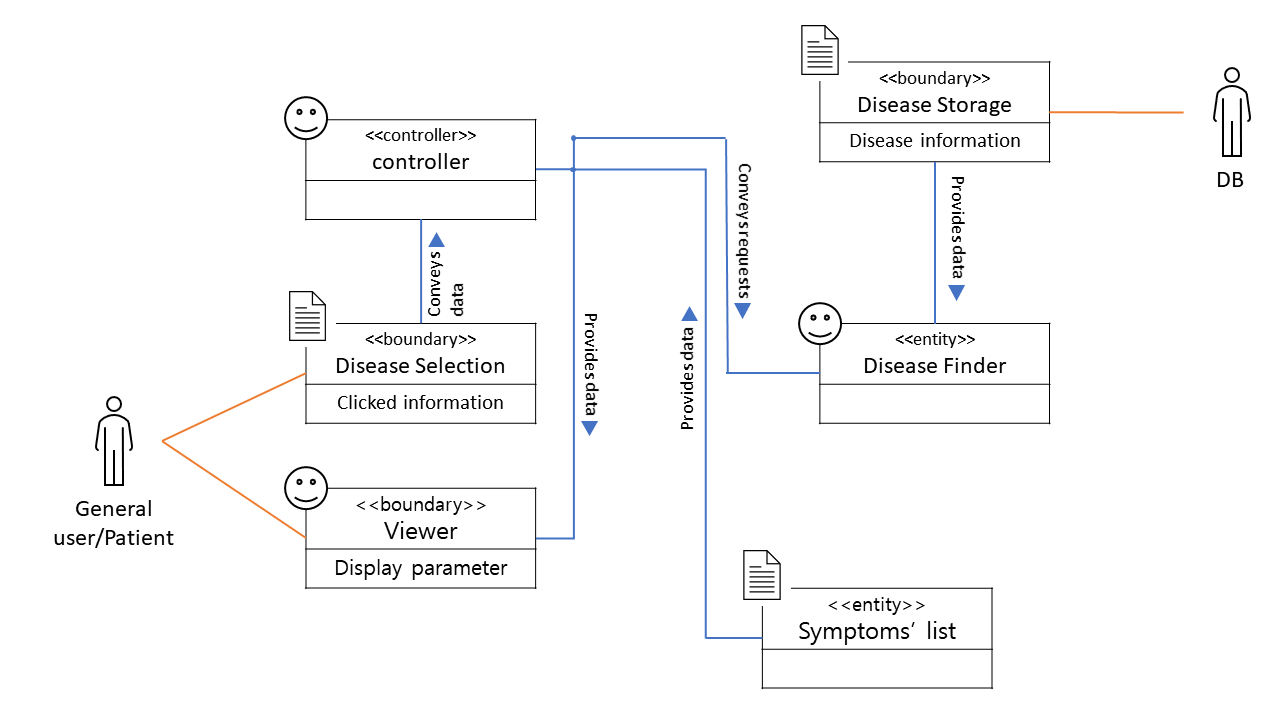
|  |  |  |
| --- | --- | --- |
| Responsibility Description | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a local grouping of use case, or the entire system and delegate the work to other concepts | D | Controller |
| Connection information for two users who request connection. | K | Request |
| Connect two users(patient and doctor). | D | ConnectOperator |
| Prepare a database query to save log about all interaction(request, conversation, etc.) | D | DatabaseConnection |
| Container for chat information(input stream, time state, send user, etc.). | K | Chat |
| Operate Send System to send user's input stream as a chat. | D | SendOperator |
| Operate Display System to display a user's chat for other user. | D | DisplayOperator |
| Operate Alarm System to inform the user that a chat is arrived, | D | AlarmOperator |
| Log all interactions with the system in persistent storage. | D | Logger |

2. Extracting the Associations

|  |  |  |
| --- | --- | --- |
| Concept Pair | Association description | Association name |
| Controller  ↔DatabaseConnection | Controller passes databases query to save logs(request information, conversation, etc.).cautions,care department to disease finder | conveys requests |
| Controller  ↔ConnectionOperator | Controller passed connection operation request to ConnectionOperator | conveys request |
| ConnectionOperator  ↔Request | Request information is passed to ConnectionOperator | conveys data |
| Controller  ↔SendOperator | Controller passed send operation request to SendOperator | conveys requests |
| SendOperator  ↔Chat | Chat information is passed to SendOperator | conveys data |
| Controller  ↔DisplayOperator | Controller passed display operation request to DisplayOperator | conveys requests |
| DisplayOperator  ↔Chat | Chat information is passed to DisplayOperator | conveys data |
| Controller  ↔AlarmOperator | Controller passed alarm operation request to AlarmOperator | conveys requests |
| DatabaseConnection  ↔Logger | Logger request DatabaseConnection to save logs about chatting information. | saves data |

* 1. Extracting the Attributes

|  |  |  |
| --- | --- | --- |
| Concept | attributes | attribute description |
| DatabaseConnection | server address | Information of server address, so that 2 user can be connected. |
| Chat | input stream | Input stream of a chat. |
|  | send information | Chat information that includes send time, sender, etc. |
| Request | user's IP information | Information of IP address, that request connection. |
|  | user information | Information of user, that contains user name, user type etc. |
| DisplayOperator | display parameters | Font type, font color, and other parameters for font, and it includes every parameters about display chat. |
| AlarmOperator | sound information | Information of alarm sound. |
|  |  |  |



* Domain model for UC-2
  1. Extracting responsibilities

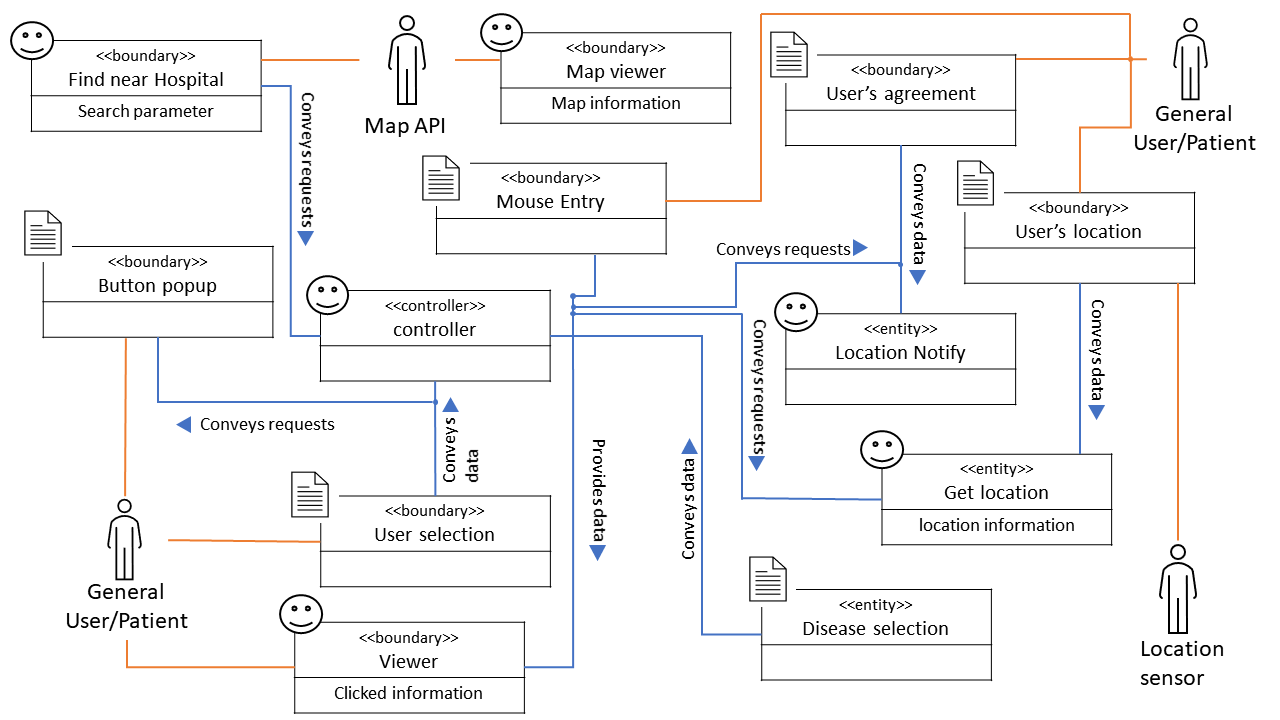
|  |  |  |
| --- | --- | --- |
| Responsibility Description | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a local grouping of use case, or the entire system and delegate the work to other concepts | D | Controller |
| Auth information for users who want to access history of previous conversations. | K | Request |
| Prepare a database query to get conversation list | D | DatabaseConnection |
| Checking the permissions of the user who want to access history of previous conversation | D | CheckAuth |
| history of previous conversation | K | ConversationList |
| The user selects the conversation wants to view from among the history of previous conversations | D | SelectConversation |
| Operate Display System to display a user's chat for user | D | DisplayOperator |

* 1. Extracting associations

|  |  |  |
| --- | --- | --- |
| Concept Pair | Association description | Association name |
| Controller  ↔CheckAuth | Auth information is passed to Controller | conveys requests |
| Controller  ↔SelectConversation | Controller passed selection operation request to SelectConversation | conveys data |
| Controller  ↔DisplayOperator | Controller passed display operation request to DisplayOperator | conveys request |
| Controlloer  ↔DatabaseConnection | Controller passes databases query to get conversation list | conveys request |
| CheckAuth  ↔Request | CheckAuth passed check operation request to Request | conveys request |
| ConversationList  ↔DisplayOperator | Conversation information is passed to DisplayOperator | conveys data |
| ConversationList  ↔DatabaseConnection | Conversation list is passed to conversation list from database. | conveys data |

* 1. Extracting attributes

|  |  |  |
| --- | --- | --- |
| Concept | attributes | attribute description |
| Request | user & conversation user | Information for determining whether the conversation participant and the current user match. |
| ConversationList | conversation participants | Information of patient, doctor |
| conversation information | Information about the conversation date |
| DisplayOperator | display parameters | Font type, font color, and other parameters for font, and it includes every parameters about display conversation |



* Domain model for UC-3
  1. Extracting responsibilities

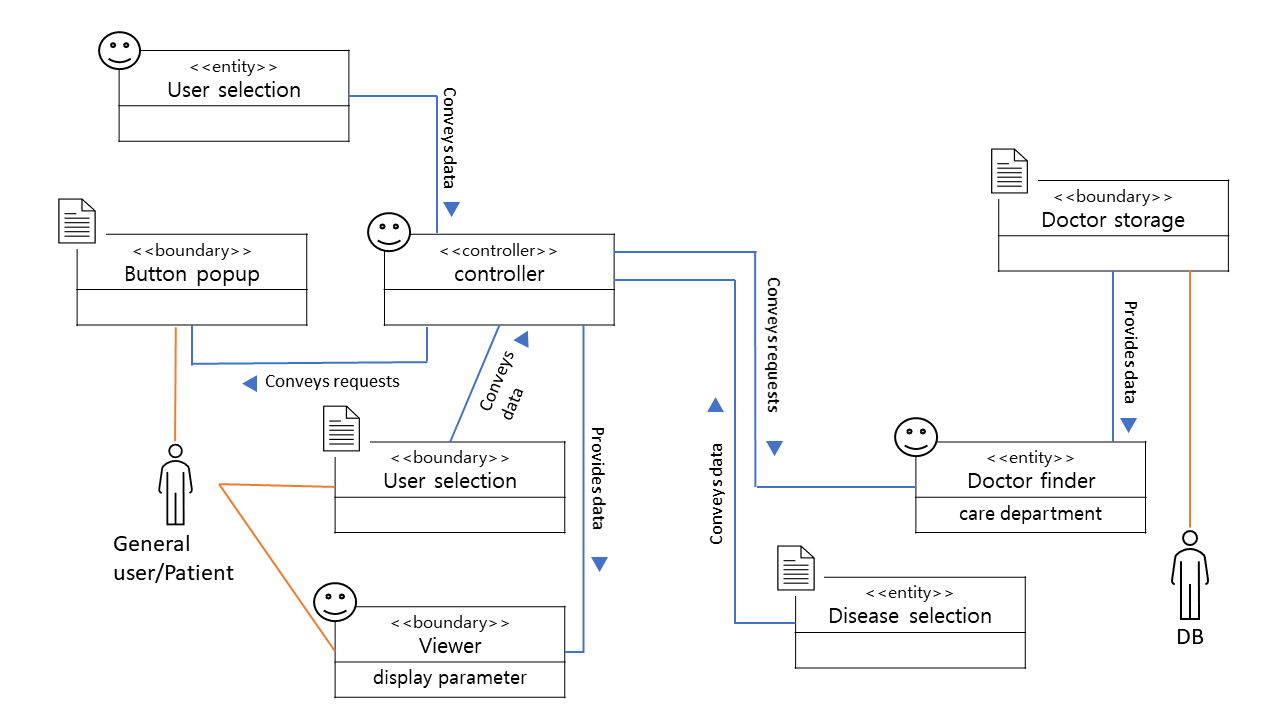
|  |  |  |
| --- | --- | --- |
| Responsibility Description | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a local grouping of use case, or the entire system and delegate the work to other concepts | D | Controller |
| Prepare a database query to save blacklist information | D | DatabaseConnection |
| Container for information about approved doctors | K | Doctor |
| Checking that the user accessed as doctor is approved doctor stored in DB. | D | CheckDoctor |
| Patient check the doctor's profile on the list and select the doctor they want to consult. | D | SelectDoctor |
| Container for information about blacklist | K | Blacklist |
| Operate blacklist system to send the user's blacklist selection information to the system | D | BlacklistOperator |
| Log adding black list information in persistent storage. | D | Logger |

* 1. Extracting associations

|  |  |  |
| --- | --- | --- |
| Concept Pair | Association description | Association name |
| Controller  ↔DatabaseConnection | Controller passes databases query to get doctor's information. | conveys request |
| DatabaseConnection  ↔Doctor | Approved doctor information is passed to Doctor from Database. | conveys data |
| DatabaseConnection  ↔Blacklist | Blacklist information is passed to Blacklist from Database. | conveys data |
| Controller  ↔CheckDoctor | Controller passed check doctor operation request to CheckDoctor. | conveys request |
| CheckDoctor  ↔Doctor | Doctor information is passed to CheckDoctor. | conveys data |
| Controller  ↔SelectDoctor | Controller passed select doctor operation request to SelectDoctor. | conveys request |
| SelectDoctor  ↔Doctor | Doctor information is passed to SelectDoctor. | conveys data |
| Controller  ↔BlacklistOperator | Controller passed blacklist operation request to BlacklistOperator. | conveys request |
| BlacklistOperator  ↔Blacklist | Blacklist information is passed to BlacklistOperator. | conveys data |
| DatabaseConnection  ↔Logger | Logger request DatabaseConnection to save logs about black list information. | saves data |

* 1. Extracting attributes

|  |  |  |
| --- | --- | --- |
| Concept | attributes | attribute description |
| SelectDoctor | select parameters | Parameter about how to show doctor's list |
| Blacklist | doctor information | Information of doctor who selected by patient. |
| patient information | Information of patient who select doctor as blacklist. |



* Domain model for UC-4
  1. Extracting responsibilities

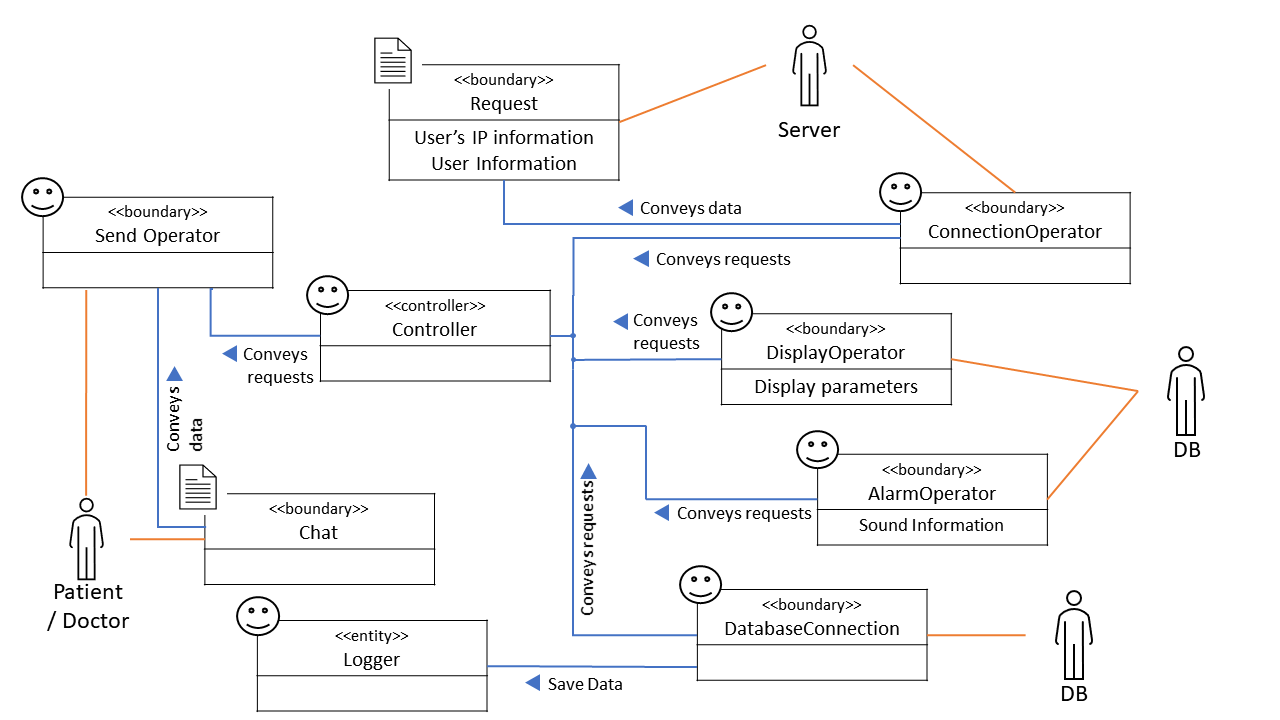
|  |  |  |
| --- | --- | --- |
| Responsibility Description | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a local grouping of use case, or the entire system and delegate the work to other concepts | D | Controller |
| Defining disease definition, causes, symptom, diagnosis treatement, progression, cautions  (정의, 원인, 증상, 진단, 치료, 경과, 주의사항, 진료과) | K | Disease storage |
| General user/Patient’s symptoms (from previous page) | K | Symptoms list |
| With a symptom list received from the pre-stage, search associated disease from Disease Storage. | D | Disease finder |
| Sends associated disease lists (from Disease finder) to viewer | D | Controller |
| View diseases found from disease finder | D | Viewer |
| General user/Patient’s disease selection | K | Disease selection |
| Pass the selection to viewer | D | Controller |
| Show detailed information of selected disease | D | Viewer |
| Go back to previous page | D | BackButton |

* 1. Extracting the Associations

|  |  |  |
| --- | --- | --- |
| Concept Pair | Association description | Association name |
| disease storage  ↔disease finder | Disease storage provides data of disease’s definition, causes, symptom, diagnosis treatment, progression, cautions,care department to disease finder | provides data |
| disease finder  ↔controller | With a symptom list received from the pre-stage, controller requests to search associated disease. Disease finder provides the disease list. | conveys request |
| controller↔ symptoms list | Symptom’s list of User is provided to controller | provides data |
| Controller  ↔viewer | Controller provides diseases list associated with symptoms list to Viewer | provides data |
| Controller  ↔disease selection | Clicked disease value is passed to Controller | conveys data |
| Controller  ↔viewer | Controller provides selected disease and its detailed information (definition, causes, symptom, diagnosis treatment, progression, cautions, care department) to Viewer | provides data |

* 1. Extracting the Attributes

|  |  |  |
| --- | --- | --- |
| Concept | attributes | attribute description |
| Disease storage | disease information | disease information are obtained by 아산병원’s website using website crawling |
| Viewer | display parameter | Page design, Table design and size, Font type, font color, etc. Includes every parameters about display disease information |
| Disease selection | clicked information | disease name, care department for next page's search (find hospital, find clinic) |



* Domain model for UC-5
  1. Extracting responsibilities

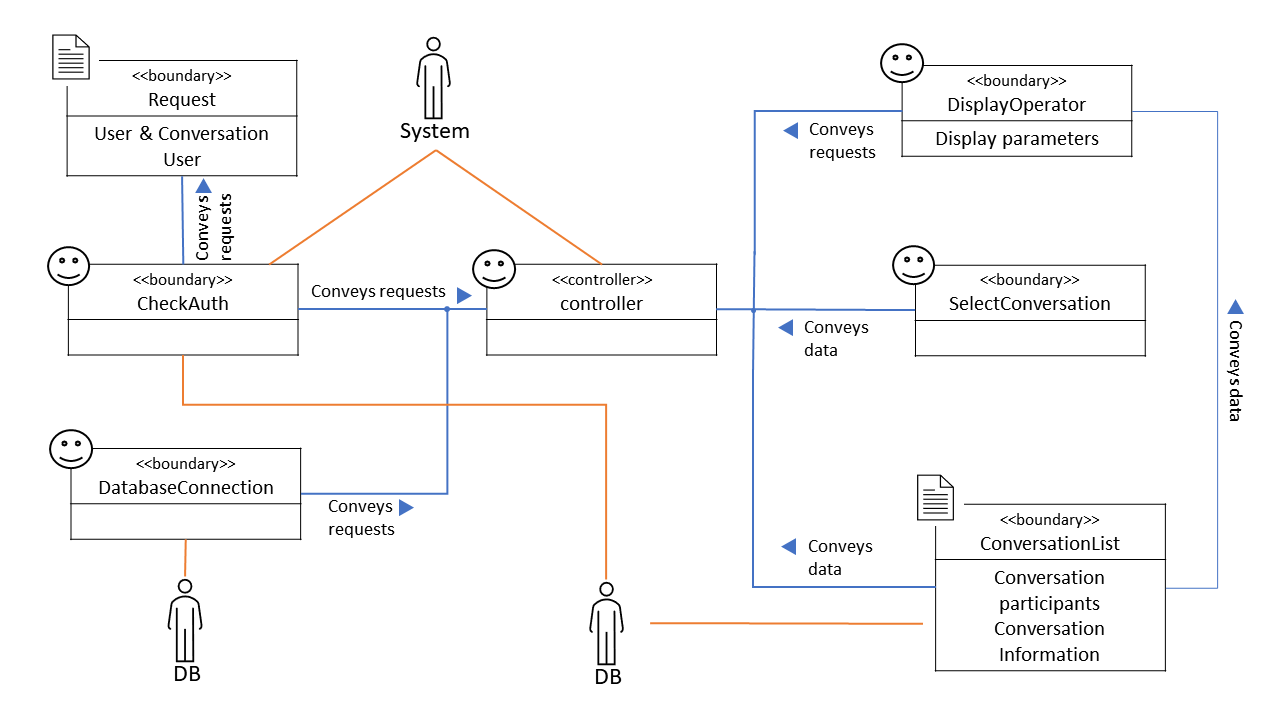
|  |  |  |
| --- | --- | --- |
| Responsibility Description | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a local grouping of use case, or the entire system and delegate the work to other concepts | D | Controller |
| General user/Patient’s disease selection | K | Disease selection |
| Move on to the next page according to the Mouse Entry(Online button, Offline button). | D | Controller |
| Online button, Offline button | K | Button popup |
| General user/Patient’s (Online/Offline) button selection | K | User selection |
| If online selected, Go to UC-6 | D | Controller |
| If offline selected, Go to the page asking if the user agrees with location information or not. | D | Controller |
| It is mandatory to find a clinic close to the user. (if user choose not then, go to previous page) | D | Location Notify |
| User agreement for providing current location | K | User’s agreement |
| Find user location | D | Location sensor |
| User location from location sensor | K | User’s location |
| Send disease information and user location to map API | D | Controller |
| Get the list of Clinic near to the user from map API | D | Find near clinic |
| Show nearby hospital recommendation | D | Viewer |
| User selection from the map | K | MouseEntry |
| Show selected hospital information | D | MapViewer |

* 1. Extracting associations

|  |  |  |
| --- | --- | --- |
| Concept Pair | Association description | Association name |
| disease selection  ↔controller | User’s diseases selection is provided to controller | conveys data |
| button popup  ↔controller | Controller requests to show offline button/ online button | conveys request |
| Controller  ↔user selection | Clicked offline value is passed to controller | conveys data |
| Controller  ↔location notify | Controller requests to show location notification(agree or not) | conveys request |
| Controller  ↔user agreement | Send if user agrees to provide current location or not | conveys data |
| Controller  ↔location sensor | Controller requests to find user’s location | conveys request |
| location sensor  ↔user’s location | Location sensor measure user’s current location and passed to user’s location | conveys data |
| Controller  ↔Find near clinic | Controller send disease information and user location to map API and requests to search and get the list of hospitals near user’s location from map API | conveys request |
| Controller  ↔Viewer | Controller provides the list of hospitals near user’s location to viewer | provides data |
| Controller  ↔MouseEntry | Clicked clinic is passed to Controller | conveys data |
| Controller  ↔MapViewer | Viewer shows selected hospital information | provides data |

* 1. Extracting attributes

|  |  |  |
| --- | --- | --- |
| Concept | attributes | attribute description |
| Get location | location information | Location sensor implemented in the current device |
| Find near Hospital | search parameter | search parameter consists of user's location and disease selection (care department) is passed to map API |
| Viewer | clicked information | Page design, Table design and size, Font type, font color, etc. Includes every parameters about display disease information |
| Map viewer | map information | map information provided by map API |



* Domain model for UC-6
  1. Extracting responsibilities

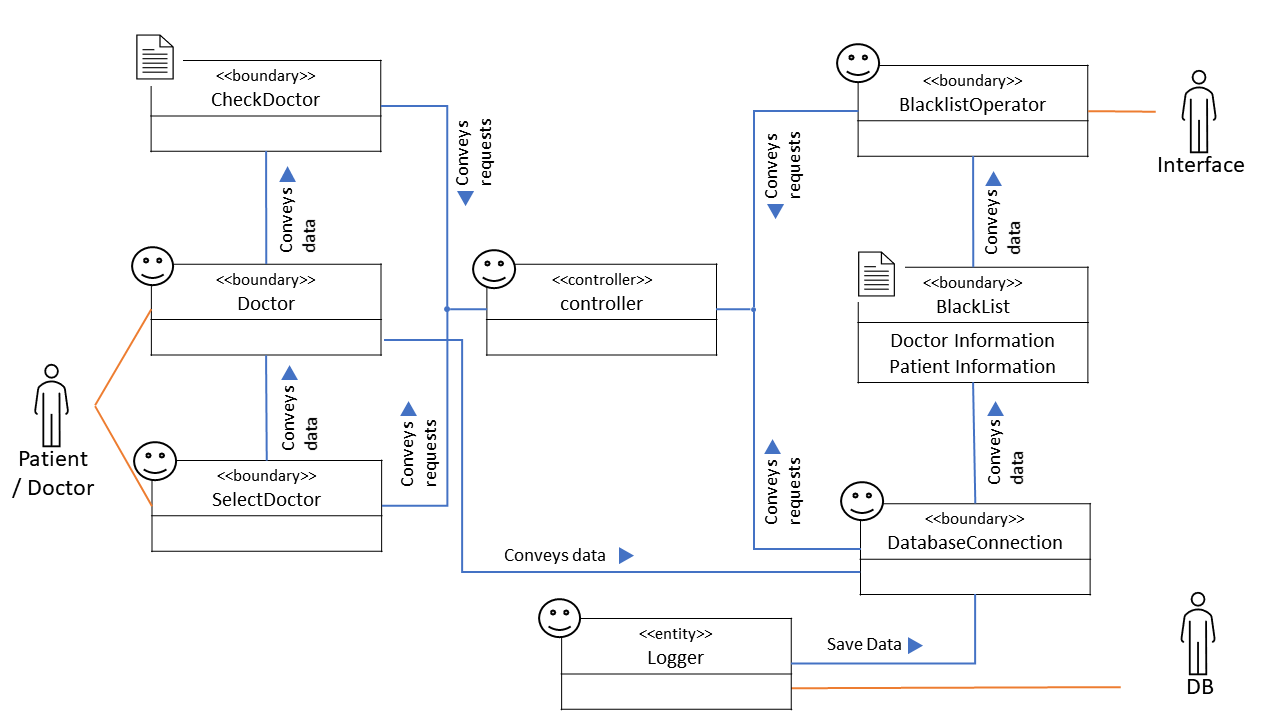
|  |  |  |
| --- | --- | --- |
| Responsibility Description | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a local grouping of use case, or the entire system and delegate the work to other concepts | D | Controller |
| General user/Patient’s disease selection | K | Disease Selection |
| Move on to the next page according to the Mouse Entry(Online button, Offline button). | D | Controller |
| Online button, Offline button | K | Button popup |
| General user/Patient’s (Online/Offline) button selection | K | User selection |
| If offline selected, Go to UC-5 | D | Controller |
| If online selected controller sends disease selection to doctor DB | D | Controller |
| Base on the Disease selection (from UC-4),  search associated doctor from doctor storage | D | Doctor finder |
| Send doctor’s list (found from Doctor finder) to Viewer | D | Controller |
| Show doctors’ list associated with Disease selection | D | Viewer |
| User selection from the doctors’ list | K | Selected Doctor |
| Send selected doctor information to match clinic page | D | Controller |

* 1. Extracting associations

|  |  |  |
| --- | --- | --- |
| Concept Pair | Association description | Association name |
| disease selection  ↔controller | User’s diseases selection is provided to controller | conveys data |
| button popup  ↔controller | Controller requests to show offline button/ online button | conveys request |
| Controller  ↔user selection | Clicked online value is passed to controller | conveys data |
| Controller  ↔doctor finder | With a disease selection received from the pre-stage(UC-4) , controller requests to find a search doctor’s list associated disease. Doctor finder provides the doctor’s list | conveys request |
| doctor finder  ↔doctor storage | Doctor storage provides every doctor’s profile data signed in the app to doctor finder | provides data |
| Viewer  ↔controller | Controller provides doctor’s list associated to the disease | provides data |
| Controller  ↔selected doctor | Clicked doctor is passed to Controller | conveys data |

1. Extracting attributes

|  |  |  |
| --- | --- | --- |
| Concept | attributes | attribute description |
| Doctor finder | care department | Care department from Disease selection and find doctors whose major is same with care department |
| Viewer | display parameter | Page design, Table design and size, Font type, font color, etc. Includes every parameters about display disease information |



* Domain model for UC-7

1. Extracting responsibilities

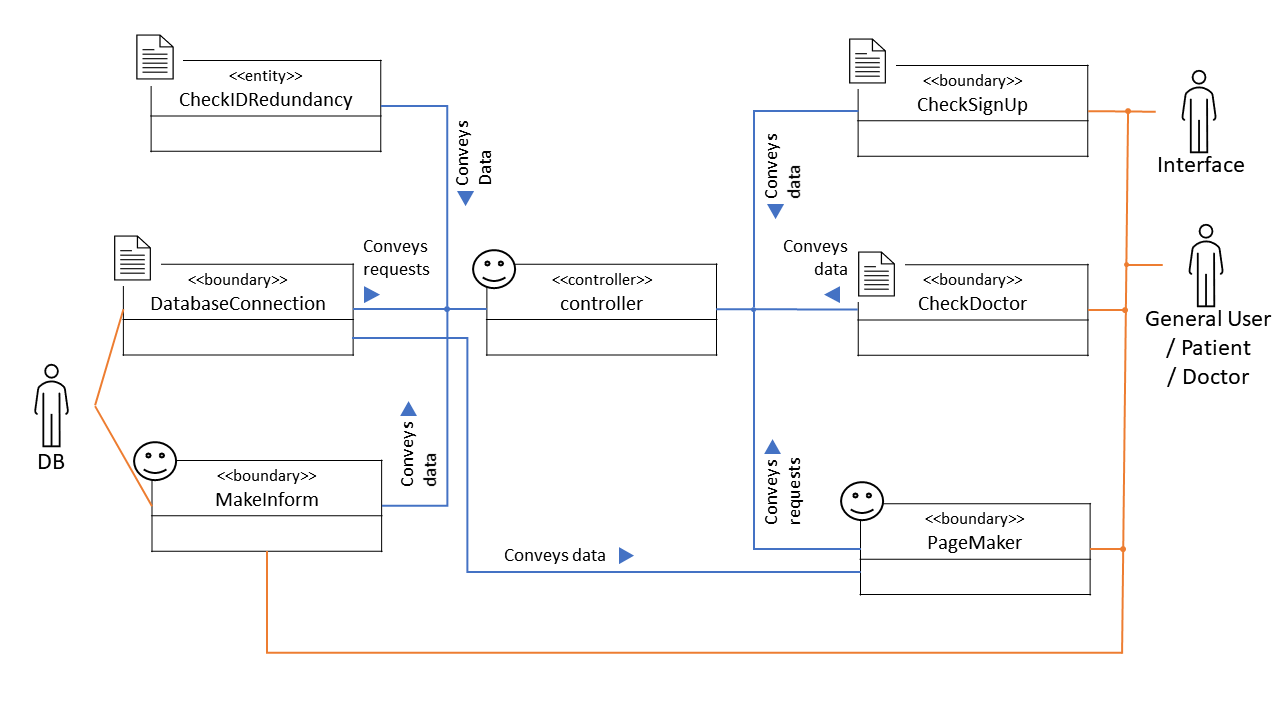
|  |  |  |
| --- | --- | --- |
| Responsibility Desciption | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a logical grouping of use cases, or the entire system and delegate the work to other concepts. | D (doing) | Controller |
| 회원가입 요청 확인 Check Sign Up Request | K (knowing) | CheckSingUp |
| 회원가입 창 띄우기 Display Sign Up Page | D (doing) | PageMaker |
| User의 입력 값 Database에 저장 Save User's input value to DB | D (doing) | MakeInform |
| User 입력 값을 DB와 비교해, ID 중복 확인 Check ID duplicate compared with DB | K (knowing) | CheckIDRedundancy |
| patient User의 own disease 정보 입력 창 띄우기 Diplay paitent user's own disease input page | D (doing) | PageMaker |
| Load speicific disease list from DB | K (knowing) | DatabaseConnection |
| patient User의 own disease 정보 DataBase에 저장 Save patient user's own disease information in DB | D (doing) | MakeInform |
| Prepare a database query to save log about all interaction(request, conversation, etc.) | D (doing) | DatabaseConnection |
| Doctor User의 의사 증빙자료 확인 Check doctor user's doctor license | K (knowing) | CheckDoctor |
| Doctor User의 병원 정보 DataBase에 저장 Save Doctor User's hospital information in the DB | D (doing) | MakeInform |

1. Extracting associations

|  |  |  |
| --- | --- | --- |
| Concept pair | Association description | Association name |
| Controller↔DatabaseConnection | Controller passes databases query to save logs(request information conversation etc.). | conveys requests |
| CheckSignup↔Controller | CheckSignup에서 회원가입 요청 여부 Controller에 전달 CheckSignup passes sign up request to Controller | conveys data |
| Controller↔PageMaker | Controller에서 회원가입 페이지 제공 / Patient 회원에게 질병에 대한 정보 및 입력 페이지 제공 / Doctor 회원에게 병원 정보 입력 페이지 제공 Controller provide SignUp page / Provide patient member disease input page / Provide hospital information input page to Doctor User | conveys requests |
| CheckIDRedundancy↔Controller | User Id 중복 여부 확인 Check for duplicate User ID | conveys data |
| Controller↔MakeInform | ID 중복 여부에 따라 User의 입력값을 저장 / Patient 회원여부에 대한 정보 제공 Save User's input value depending on whether ID is duplicated / Provide information on whether he is a patient user | conveys data |
| DatabaseConnection↔PageMaker | 특정 질병에 대한 정보를 Controller에 전달 Deliver information on specific diseases to Controller | conveys data |
| CheckDoctor↔Controller | Doctor 여부에 대한 정보 제공 Provide information on whether he is a doctor | conveys data |

1. Extracting attributes

|  |  |  |
| --- | --- | --- |
| Concept | Attiributes | Attribute Description |
| PageMaker | General User 정보 입력 Enter General User information | General User에 체크한 경우 관련 정보 입력 페이지 제공 Provides personal information input page when checking General User |
|  | Patient User 정보 입력 Enter Patient User information | Patient User에 체크한 경우 질병 관련 정보 제공 및 환자의 질병에 대한 입력 페이지 제공 Provides disease-related information and input page for patient's disease when checking Patient User |
|  | Doctor User 정보 입력 Enter Doctor User information | 의사 증빙 여부 확인 후 Doctor 회원에게 병원 정보 입력 페이지 제공 Provides hospital information input page to Doctor members after checking the evidence of doctor |



* Domain model for UC-8

1. Extracting responsibilities

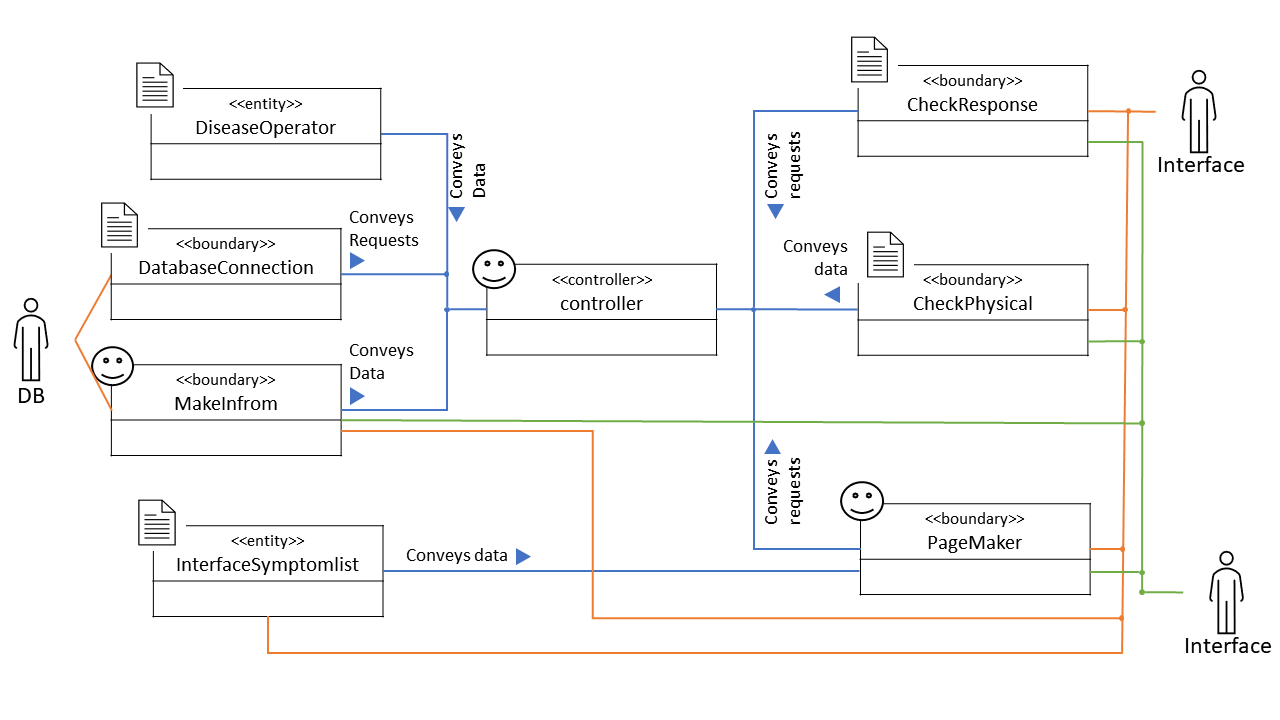
|  |  |  |
| --- | --- | --- |
| Responsibility Desciption | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a logical grouping of use cases, or the entire system and delegate the work to other concepts. | D (doing) | Controller |
| symptom list 목록 띄우기 Display symptom list | K (knowing) | InterfaceSymptomlist |
| Physical Condition 등록 요청 확인 Check Physical Condition Registration Request | K (knowing) | CheckPhysical |
| Physical Condition 등록 창 띄우기 Display Physical Condition registration page | D (doing) | PageMaker |
| Prepare a database query to save log about all interaction(request, conversation, etc.) | D (doing) | DatabaseConnection |
| User의 Physical Condition 등록 data 저장 Save User's phyical condition data in DB | D (doing) | MakeInform |
| User의 지난 Physical Condition Database에서 불러오기 Load user's last physical condition from the database | K (knowing) | DataBaseConnection |
| Check User's Physical Condition LookUp request | K (knowing) | CheckPhysical |
| Display User's past physical condition page | D (doing) | PageMaker |
| Check the list of suspicious disease in the user's physical condition | K (knowing) | DiseaseOperator |
| Display Search Disease recommendation page | D (doing) | PageMaker |
| Check Search Disease Request | K (knowing) | CheckResponse |

1. Extracting associations

|  |  |  |
| --- | --- | --- |
| Concept pair | Association description | Association name |
| Controller↔DatabaseConnection | Controller passes databases query to save logs(request information conversation etc.). | conveys requests |
| CheckPhysical↔Controller | CheckPhyiscal passes Physcial Condition request to Controller | conveys data |
| Controller↔PageMaker | Controller provides Physical Condition input page / Controller provieds Search Disease recommendation input page | conveys requests |
| Controller↔MakeInform | MakeInform passes Physical Conditon data to Controller | conveys data |
| InterfaceSymptomlist↔PageMaker | Provide Symptom List when registering Physical Condition | conveys data |
| DiseaseOperator↔Controller | DiseaseOperator checks the disease suspicion record and passes it to Controller when the appropriate amount is abnormal. | conveys data |
| CheckResponse↔Controller | CheckResponse passes Search Disease request to Controller | conveys requests |

1. Extracting attributes

|  |  |  |
| --- | --- | --- |
| Concept | Attiributes | Attribute Description |
| InterfaceSymptomList | Provide list of Symptom | Provides a list of common diseases related to headache and fever, etc. |
| DiseaseOperator | Provide list of disease | 사용자의 지난 physical condition을 확인하고, 특정 증상 지속 시 의심되는 질병 목록 확인 |



* Domain model for UC-9

1. Extracting responsibilities

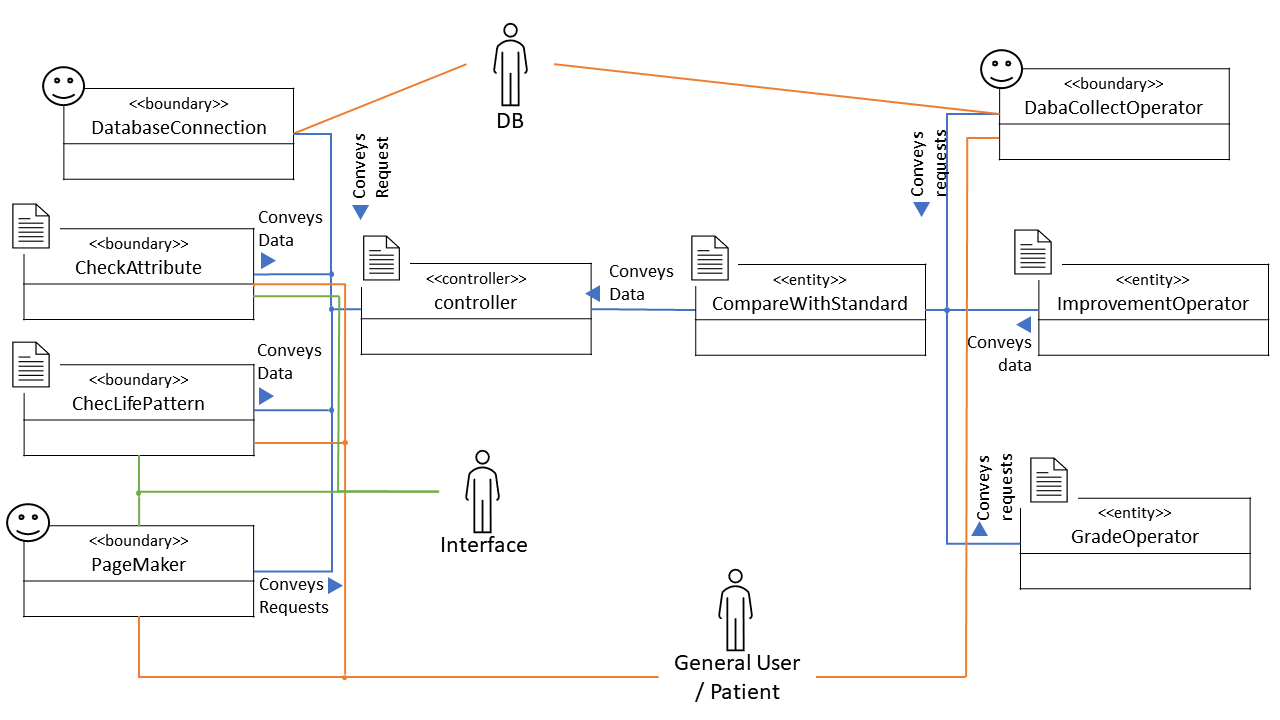
|  |  |  |
| --- | --- | --- |
| Responsibility Desciption | Type | Concept Name |
| Coordinate actions of all concepts associated with a use case, a logical grouping of use cases, or the entire system and delegate the work to other concepts. | D (doing) | Controller |
| Check LifePattern input request | K (knowing) | CheckLifePattern |
| Display selecting page (Sleeping Pattern / Diet / Exercise Record) | D (doing) | PageMaker |
| Prepare a database query to save log about all interaction(request, conversation, etc.) | D (doing) | DatabaseConnection |
| Save user's life patern data in DB | D (doing) | DataCollectOperator |
| Check which one is selected among Sleeping Pattern/Diet/Exercise Record | K (knowing) | CheckAttribute |
| Comparison of DB's Life Pattern Standard and user's life pattern | K (knowing) | CompareWithStandard |
| Display life pattern grade page | D (doing) | PageMaker |
| check user's life pattern grade | K (knowing) | GradeOperator |
| check user's improvements in his life pattern | K (knowing) | ImprovementOperator |

1. Extracting associations

|  |  |  |
| --- | --- | --- |
| Concept pair | Association description | Association name |
| Controller↔DatabaseConnection | Controller passes databases query to save logs(request information, conversation, etc.). | conveys requests |
| CheckLifePatter↔Controller | CheckLifePatter passes LifePattern input request to Controller | conveys data |
| Controller↔PageMaker | Controller provides selection page (Sleeping Pattern/Data/Exercise Record) / PageMaker provides page about selected one among Sleeping Pattern, Diet, and Exercise Record according to information passed by Controller | conveys requests |
| CheckAttribute↔Controller | CheckAttribute passes selected one among Sleeping Pattern, Diet, and Exercise Record to Controller | conveys data |
| DataCollectOperator↔CompareWithStandard | DataCollectOperatr passes information about user's life pattern to ComparWithStandard to compare it with life pattern's standard | conveys data |
| CompareWithStandard↔GradeOperator | Grade calculation through user's life pattern and standard | conveys data |
| CompareWithStandard↔ImprovementOperator | Check improvement of user's life pattern by comparing user's life pattern and standard | conveys data |
| Controller↔CompareWithStandard | Controller passes standard information to CompareWithStandard | conveys data |

1. Extracting attributes

|  |  |  |
| --- | --- | --- |
| Concept | Attiributes | Attribute Description |
| CompareWithStandard | compare user's life pattern and standard | Save standard life pattern values, and compare it with user's life pattern by sleeping pattern / diet / exercise record |
| GradeOperator | grade life pattern | Calculate total score after checking in percentage how much user's life pattern meets the standard life pattern for each sleeping pattern / diet / exercise record |
| ImprovementOperator | give improvement in user's life pattern | Provide improvement of life pattern which idoes not meet the standard by sleeping pattern / diet / exercise record |



* Domain model for UC-10

1. Extracting responsibilities

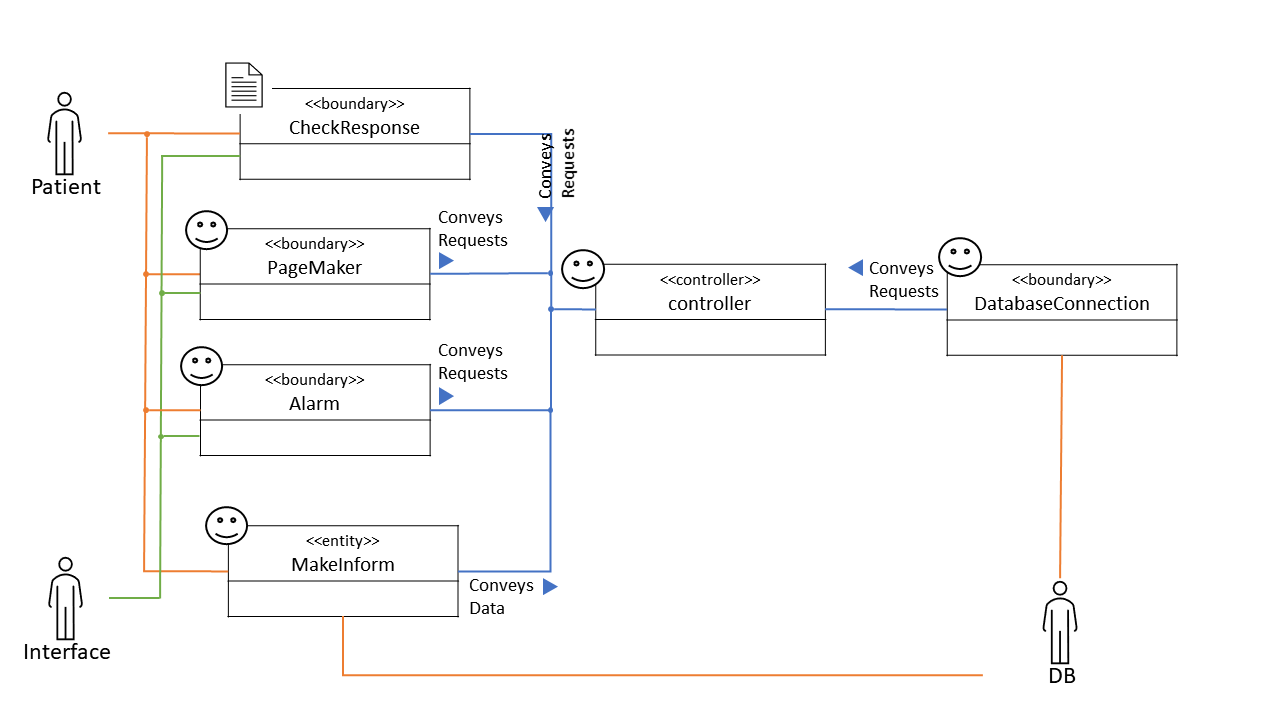
|  |  |  |
| --- | --- | --- |
| Concept | Attiributes | Attribute Description |
| CompareWithStandard | compare user's life pattern and standard | Save standard life pattern values, and compare it with user's life pattern by sleeping pattern / diet / exercise record |
| GradeOperator | grade life pattern | Calculate total score after checking in percentage how much user's life pattern meets the standard life pattern for each sleeping pattern / diet / exercise record |
| ImprovementOperator | give improvement in user's life pattern | Provide improvement of life pattern which idoes not meet the standard by sleeping pattern / diet / exercise record |
| Concept | Attiributes | Attribute Description |
| CompareWithStandard | compare user's life pattern and standard | Save standard life pattern values, and compare it with user's life pattern by sleeping pattern / diet / exercise record |
| GradeOperator | grade life pattern | Calculate total score after checking in percentage how much user's life pattern meets the standard life pattern for each sleeping pattern / diet / exercise record |
| ImprovementOperator | give improvement in user's life pattern | Provide improvement of life pattern which idoes not meet the standard by sleeping pattern / diet / exercise record |
| Concept | Attiributes | Attribute Description |
| CompareWithStandard | compare user's life pattern and standard | Save standard life pattern values, and compare it with user's life pattern by sleeping pattern / diet / exercise record |
| GradeOperator | grade life pattern | Calculate total score after checking in percentage how much user's life pattern meets the standard life pattern for each sleeping pattern / diet / exercise record |
| ImprovementOperator | give improvement in user's life pattern | Provide improvement of life pattern which idoes not meet the standard by sleeping pattern / diet / exercise record |
| Concept | Attiributes | Attribute Description |

1. Extracting associations

|  |  |  |
| --- | --- | --- |
| Concept pair | Association description | Association name |
| Controller↔DatabaseConnection | Controller passes databases query to save logs(request information conversation etc.). | conveys requests |
| Controller↔CheckResponse | CheckResponse provides Symptom/action/notification registration request to Controller | conveys requests |
| Controller↔MakeInform | MakeInform passes User's input to Controller | conveys data |
| Controller↔PageMaker | Controller provides symptom record / action record / notification registration page | conveys requests |
| Controller↔Alarm | Send notification to User at the speicified time | conveys requests |

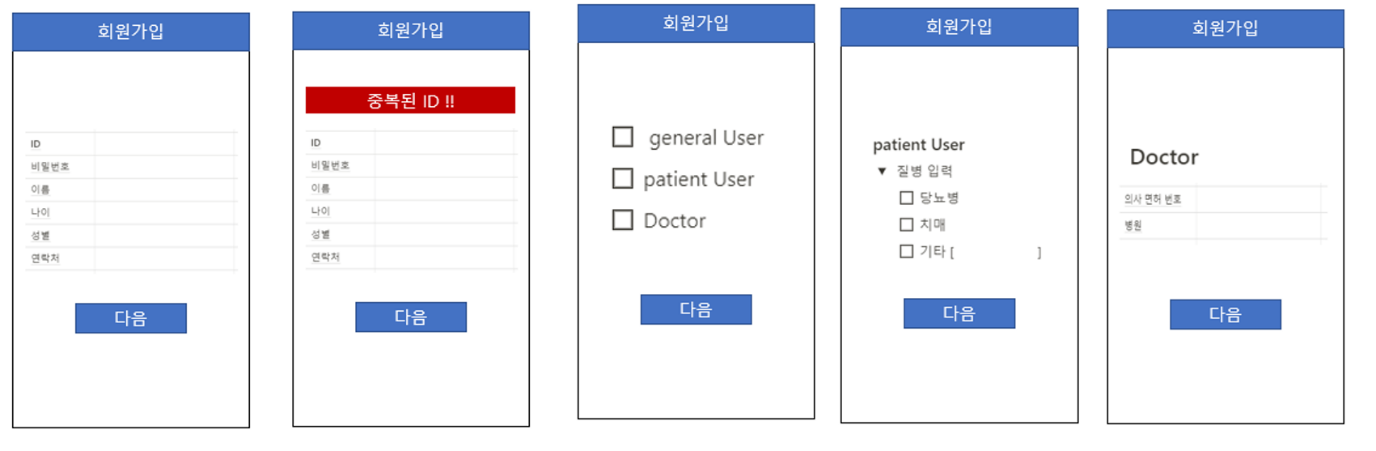
1. Extracting attributes

|  |  |  |
| --- | --- | --- |
| Concept | Attiributes | Attribute Description |
| Alarm | Provide notification | Send notification continuously at specific times depending on user's disease |



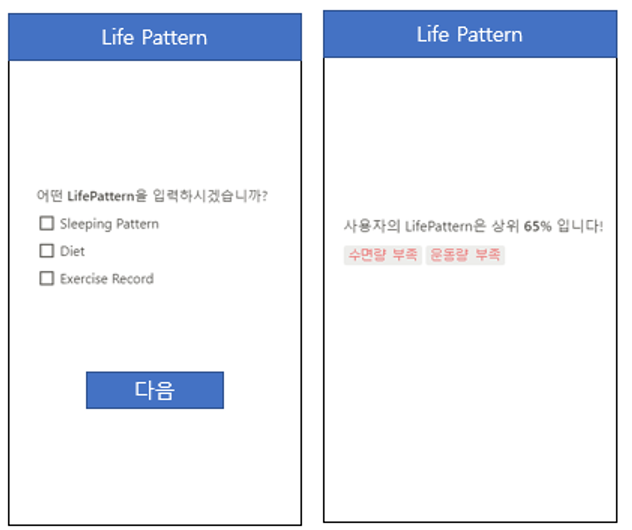
1. User Interface mockups

* 회원가입 페이지

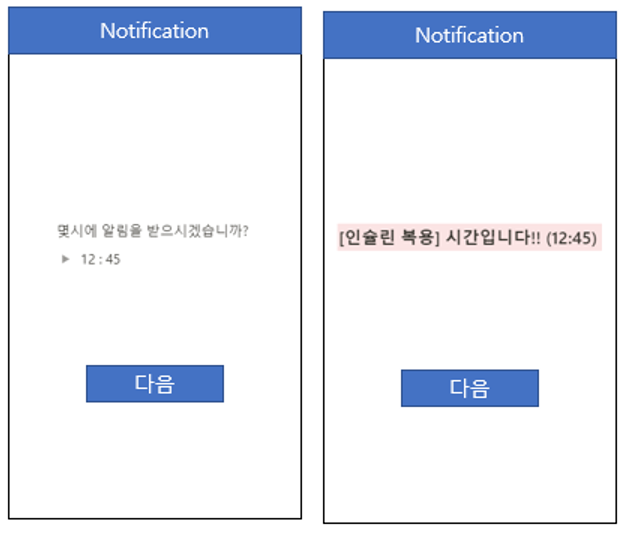


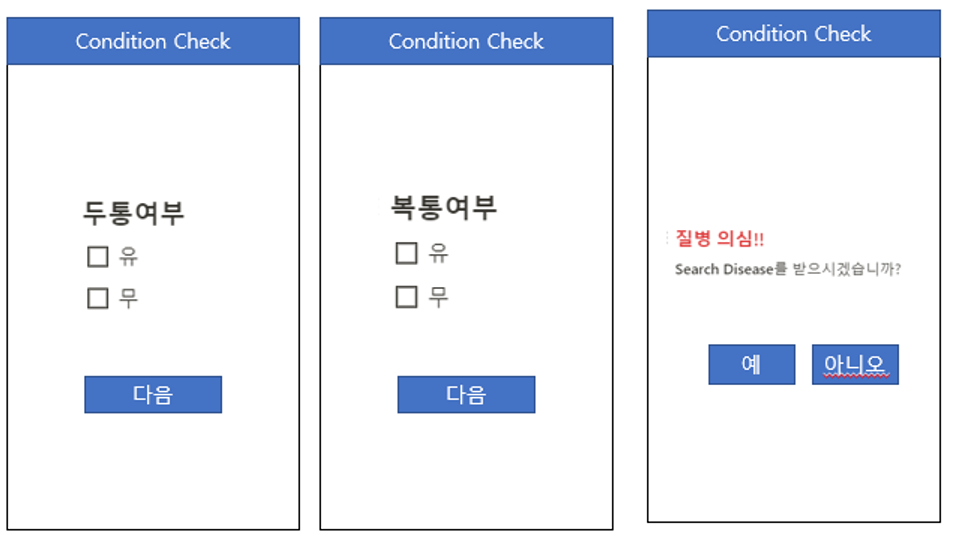
1. 회원가입시 general User 선택
2. 회원가입시 patient User 선택
3. 회원가입시 Doctor 선택

* General user Mypage



* Patient User Mypage

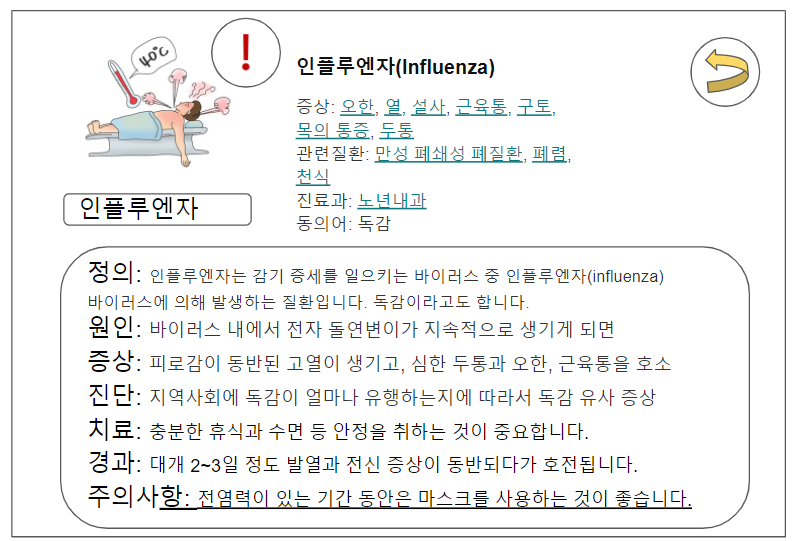
  
- 건강 체크리스트(General/Patient User)



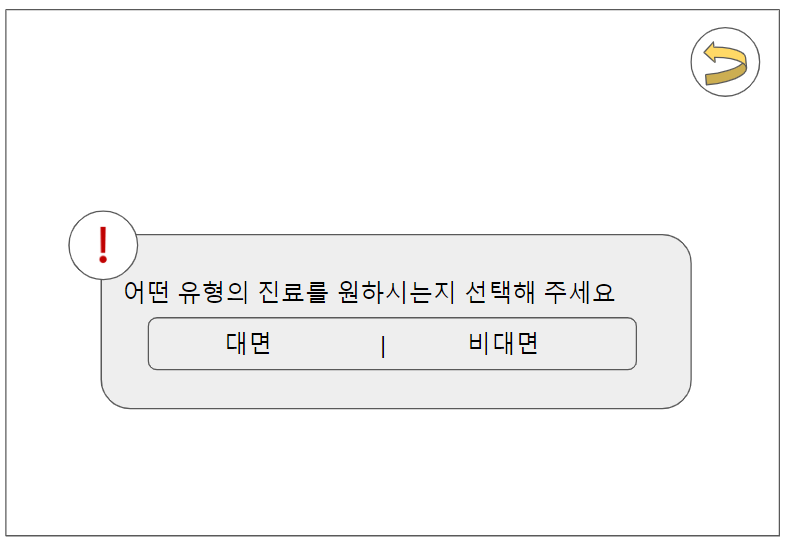
* 체크리스트 결과



* 질병 세부 페이지



* 진료유형 선택

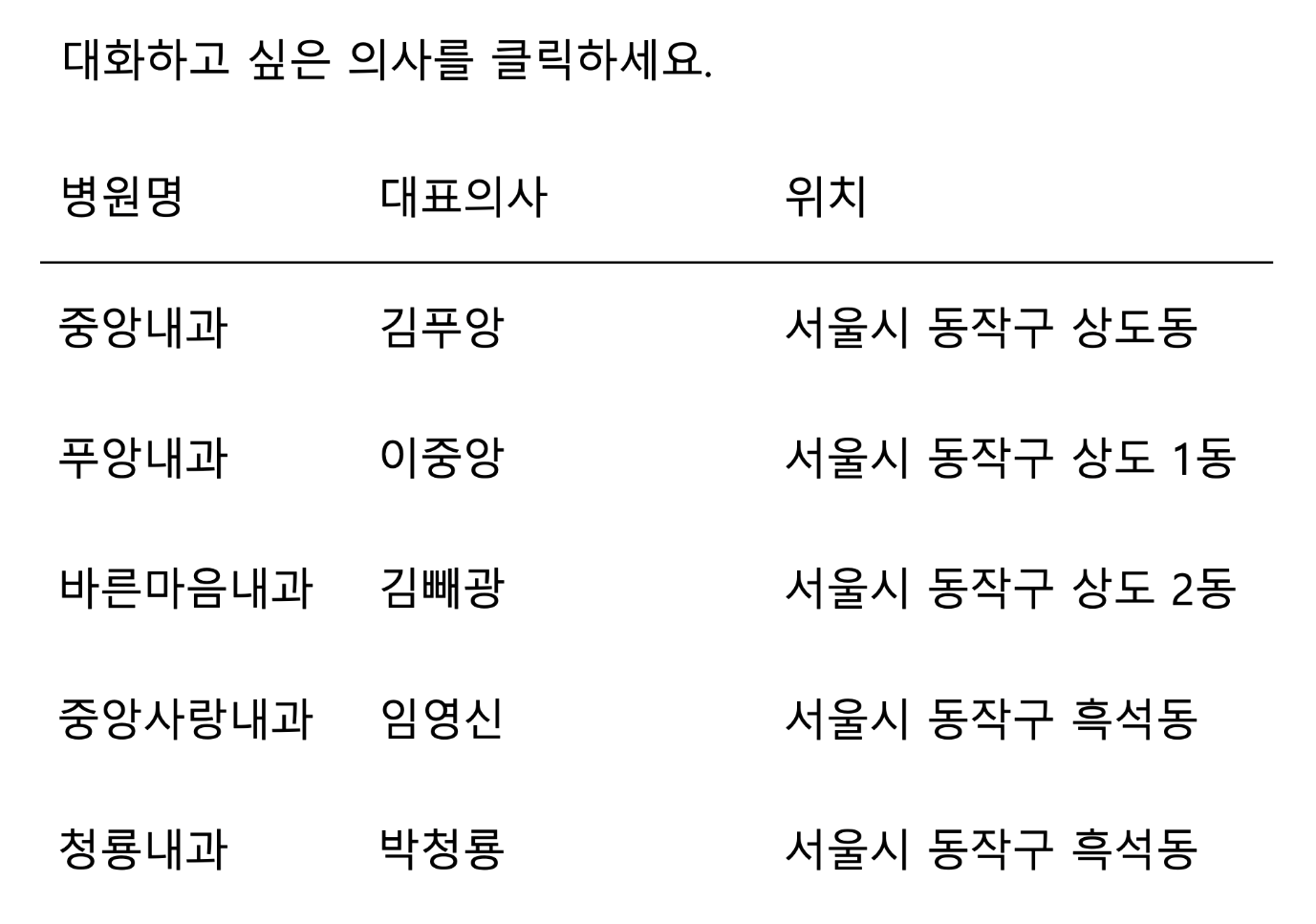


* 대면 선택



* 비대면 선택





* 블랙리스트에 추가

