C:\Users\ajou\Google 드라이브\0_취업관련\2016하반기\0_서류합격\LIG넥스원\2차면접\lig넥스원로고.PNG

**Architectural Drivers Specification**

<채팅 프로그램>



**유도무기1연구센터1팀**

**지원자 이병욱**

**Date of Submission: 2016.11.08**

Contents

[Figure contents 4](#_Toc466125055)

[Table contents 5](#_Toc466125056)

[1. Vision 6](#_Toc466125057)

[1.1 Revision History 6](#_Toc466125058)

[1.2 Glossary 6](#_Toc466125059)

[1.3 Introduction 7](#_Toc466125060)

[1.4 Positioning 7](#_Toc466125061)

[1.4.1 Business Opportunity 7](#_Toc466125062)

[1.4.2 Problem Statement 7](#_Toc466125063)

[1.4.3 Product Position Statement 7](#_Toc466125064)

[1.5 Stakeholder Descriptions 7](#_Toc466125065)

[1.5.1 Stakeholder Summary 7](#_Toc466125066)

[1.5.2 Key High-Level Goals and Problems of the Stakeholders 7](#_Toc466125067)

[1.5.3 User-Level Goals 8](#_Toc466125068)

[2. Requirements 9](#_Toc466125069)

[2.1 Functional Requirements 9](#_Toc466125070)

[2.1.1 Template 9](#_Toc466125071)

[2.1.2 Specifications 9](#_Toc466125072)

[2.1.3 Specifications Traceability Matrix 11](#_Toc466125073)

[2.1.4 Use Case Diagram 11](#_Toc466125074)

[2.1.5 Use Case List 12](#_Toc466125075)

[2.1.6 Use Cases 12](#_Toc466125076)

[2.2 Quality Attribute Requirements 17](#_Toc466125077)

[2.2.1 Quality Attributes List 17](#_Toc466125078)

[2.2.2 Quality attribute scenarios 18](#_Toc466125079)

[2.3 Constraints 20](#_Toc466125080)

[2.3.1 Business Constraints 20](#_Toc466125081)

[2.3.2 Technical Constraints 21](#_Toc466125082)

[3. Prioritization 22](#_Toc466125083)

[3.1 Priority scale 22](#_Toc466125084)

[3.2 Difficulty Ranking scale 22](#_Toc466125085)

[3.3 Use Cases 22](#_Toc466125086)

[3.4 Quality Attribute Scenarios 22](#_Toc466125087)

[3.5 Constraints 23](#_Toc466125088)

[3.5.1 Business Constraints 23](#_Toc466125089)

[3.5.2 Technical Constraints 23](#_Toc466125090)

[3.6 Significant Architectural Drivers 23](#_Toc466125091)

[4. Risk Management 24](#_Toc466125092)

[4.1 Risk Management Process 24](#_Toc466125093)

[4.2 Risk Identification & Analysis 24](#_Toc466125094)

[4.2.1 Qualitative Risk Analysis 24](#_Toc466125095)

[4.3 Risk Response Planning 24](#_Toc466125096)

[5. Architecture 25](#_Toc466125097)

[5.1 Architecture Design Decision 25](#_Toc466125098)

[5.1.1 Client Server Pattern 25](#_Toc466125099)

[5.1.2 Database Types 25](#_Toc466125100)

[5.1.3 Database Management System 25](#_Toc466125101)

[5.2 Context Diagram 26](#_Toc466125102)

[5.3 Physical View 26](#_Toc466125103)

[5.3.1 Deployment View 26](#_Toc466125104)

[5.4 Static View 27](#_Toc466125105)

[5.4.1 Module View 27](#_Toc466125106)

[5.5 Dynamic View 27](#_Toc466125107)

[5.5.1 Components and Connectors View 27](#_Toc466125108)

[6. Requirements Traceability Matrix 28](#_Toc466125109)

[7. Description of how the system will (or will not) satisfy the future needs listed in the project description 31](#_Toc466125110)

# Figure contents

[Figure 1. Use case diagram 11](#_Toc466125127)

[Figure 2. Context diagram of Chatting program 26](#_Toc466125128)

[Figure 3. Deployment view of Chatting program 26](#_Toc466125129)

[Figure 4. Module view of Chatting program 27](#_Toc466125130)

[Figure 5. High Level C&C view of Chatting program 27](#_Toc466125131)

# Table contents

[Table 1. Revision History 6](#_Toc466125116)

[Table 2. Glossary of Chatting program 6](#_Toc466125117)

[Table 3. Key High-Level Goals and problem of the stakeholders 8](#_Toc466125118)

[Table 4 Specifications Traceability Matrix 11](#_Toc466125119)

[Table 5. Use Case List of Chatting program 12](#_Toc466125120)

[Table 6. Quality attributes list of Chatting program 17](#_Toc466125121)

[Table 7. Business constraints of Chatting program 20](#_Toc466125122)

[Table 8. Technical constraints of Chatting program 21](#_Toc466125123)

[Table 9. Identified and analyzed risks 24](#_Toc466125124)

[Table 10. Response planning of analyzed risks 24](#_Toc466125125)

[Table 11 Requirements Traceability Matrix 28](#_Toc466125126)

1. **Vision**
   1. **Revision History**

Table 1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Vision** | **Date** | **Description** | **Author** |
| ADS | November 4, 2016 | Define Functional Requirements | Byungwook Lee |
| ADS | November 5, 2016 | Define Quality Attribute Requirements | Byungwook Lee |
| ADS | November 5, 2016 | Design Architecture | Byungwook Lee |

* 1. **Glossary**

Table 2. Glossary of Chatting program

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Term** | **Description** | **Author** | **Date** | **Version** |
| 사용자 | 채팅 프로그램 유저 = 사용자 = 고객 | ByungwookLee | 04/11/2016 | 0.01 |
| 채팅 프로그램 | 본 프로그램의 이름 | ByungwookLee | 04/11/2016 | 0.01 |
| 채팅방 | 사용자들이 채팅 할 수 있는 공간 | ByungwookLee | 04/11/2016 | 0.01 |
| 그림판 영역 | 사용자들이 공유하여 작업할 수 있는 그림판 공간 | ByungwookLee | 04/11/2016 | 0.01 |
| 채팅 영역 | 사용자들이 채팅내용을 공유하는 영역(채팅공유 영역과 입력 영역) | ByungwookLee | 04/11/2016 | 0.01 |
| 전송 버튼 | 채팅영역에 채팅내용을 입력하도록 하는 버튼 (=Enter) | ByungwookLee | 04/11/2016 | 0.01 |
| 품질 속성 | Quality attributes | ByungwookLee | 05/11/2016 | 0.02 |
| 품질 속성 시나리오 | Quality attribute scenario | ByungwookLee | 05/11/2016 | 0.02 |
| 아키텍처 | Architecture | ByungwookLee | 05/11/2016 | 0.02 |
| 세부적 설계 | Detailed design | ByungwookLee | 05/11/2016 | 0.02 |
| 정상 작업 | Normal Operation | ByungwookLee | 05/11/2016 | 0.02 |
| 비즈니스 제약 | Business constraints | ByungwookLee | 05/11/2016 | 0.02 |
| 기술적 제약 | Technical constraints | ByungwookLee | 05/11/2016 | 0.02 |
| 통합개발환경 | Integrated Development Environment (IDE) | ByungwookLee | 05/11/2016 | 0.02 |
| 채팅 프로그램 | Chatting program | ByungwookLee | 05/11/2016 | 0.02 |
| 기능적 요구사항 | Functional Requirements | ByungwookLee | 05/11/2016 | 0.02 |
| 컨텍스트 다이어그램 | Context Diagram | ByungwookLee | 05/11/2016 | 0.02 |

* 1. **Introduction**
* LIG넥스원 유도무기1연구센터1팀 2차 면접전형(현업면접)을 실시하기 위한 사전 과제인 채팅프로그램을 개발 목표로 한다.
  1. **Positioning**
     1. **Business Opportunity**
* 기존의 문서나 대화를 통한 채팅이 아닌 실시간으로 생각을 공유할 수 있는 채팅프로그램으로 공간적 시간적으로 같이 회의를 할 수 없는 경우 본 채팅 프로그램을 통해 실시간으로 회의를 진행 할 수 있다. 이는 채팅 프로그램의 새로운 사업기회라고 생각한다.
  + 1. **Problem Statement**
* 기존의 카카오톡, 라인, 네이트온, 스카이프, 구글의 행아웃 등 일반적인 채팅에는 실시간으로 대화나 문서를 공유할 수 있지만, 사용자들끼리 하나의 그림판에 아이디어를 생각나는 데로 그려가는 브레인스토밍을 공유하기에 어려운 점이 있다. 본 프로그램은 실시간으로 그림판을 공유함으로써 이 문제점을 해결하는 것을 목표로 한다.
  + 1. **Product Position Statement**
* 본 채팅 프로그램의 목표는 여러 사용자가 채팅 할 수 있는 환경을 제공하는 것이다. 이 시스템은 이 목적을 달성하기 위해 프로그램의 확장성을 고려하여 개발할 것이다.
  1. **Stakeholder Descriptions**
     1. **Stakeholder Summary**
* **채팅프로그램** **사용자**: 사용자들은 실시간으로 그림판을 공유하고 채팅하길 원한다.
* **LIG넥스원** **관계자**: 지원자가 평가기준을 만족하여 프로그램 소스 코드와 개발 문서를 제출했는지 확인한다.
* **지원자**: 제출 과제 세부 사항에 제시 되어있는 요구사항을 분석하고 이를 문서화하고 개발한다. (Architect, Project manager, Developer, Tester 역할을 수행)
  + 1. **Key High-Level Goals and Problems of the Stakeholders**

Table 3. Key High-Level Goals and problem of the stakeholders

|  |  |  |  |
| --- | --- | --- | --- |
| **High-Level Goals** | **Priority** | **Problems and Concerns** | **Current Solutions** |
| 채팅 프로그램을 통해 사용자끼리 그림판 영역 공유 | high | 시, 공간적 제약을 가지고 있는 회의 관계자들의 회의 시간을 잡기 어렵다.  관계자들끼리 회의 가능 시간을 공유하여 오프라인 회의를 해야 한다. | 기존 채팅프로그램을 사용하여 브레인스토밍한 내용을 파일로 공유하거나, 스카이프와 같은 프로그램을 이용하여 화면공유를 통해 브레인스토밍을 한다. |

* + 1. **User-Level Goals**
* **채팅프로그램 사용자**: 시간적 공간적 제약이 있는 사용자들끼리 그림판 공유를 통해 쉽게 회의를 하고 브레인스토밍을 할 수 있다.

1. **Requirements**
   1. **Functional Requirements**
      1. **Template**

|  |  |
| --- | --- |
| Title of functional requirement | **ID:** unique identifier. |
| **Priority:** priority integer based on the priority table. |
| **Version:** integer version number. |
| **Last Changed:** Date this specification was last changed. Both Version and this field will be changed at the same time. Only changes that affect the functionality should affect these two fields, and not minor grammatical changes. |
| Description of the requirement specification. | |
| **Open Issues:** These are any ambiguities or conflicts in requirements that need to be clarified with the client. Additionally, these can be issues that the team needs to solve on their own. | |

* + 1. **Specifications**
* Prioritization scale has been defined in section 6.1

|  |  |
| --- | --- |
| * + - 1. **여러 사용자가 채팅 할 수 있는 환경을 제공한다** | **ID:** FR1 |
| **Priority:** 1 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 사용자들은 채팅 프로그램을 통해 다른 사용자와 채팅 할 수 있는 환경을 제공받아야 한다. | |
| **Open Issues** | |

|  |  |
| --- | --- |
| * + - 1. **사용자가 채팅방을 만들 수 있도록 한다** | **ID:** FR2 |
| **Priority:** 2 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 사용자는 자신이 채팅 하고자 하는 방을 만들 수 있다. 사용자는 채팅방의 이름을 입력하고 비밀번호 설정이 필요한 경우 입력해야 한다. | |
| **Open Issues** | |

|  |  |
| --- | --- |
| * + - 1. **채팅방 사용자들에게 공유할 수 있는 그림판을 제공한다** | **ID:** FR3 |
| **Priority:** 1 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 사용자들은 공유 된 그림판에 내용을 입력하거나 지울 수 있다. | |
| **Open Issues** | |

|  |  |
| --- | --- |
| * + - 1. **채팅방 사용자들에게 공유할 수 있는 채팅 영역을 제공한다** | **ID:** FR4 |
| **Priority:** 1 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 사용자들은 공유 된 채팅영역을 통해 사용자들끼리 채팅내용을 공유할 수 있다. | |
| **Open Issues** | |

|  |  |
| --- | --- |
| * + - 1. **채팅방 사용자들에게 채팅영역에 입력할 수 있는 입력 영역을 제공한다** | **ID:** FR5 |
| **Priority:** 1 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 사용자들은 채팅영역에 입력란을 통해 채팅내용을 입력할 수 있다. 사용자는 채팅할 내용을 입력한 후 “Enter”또는 “전송”버튼을 누르는 시점에 타인의 채팅창에 내용이 공유된다. | |
| **Open Issues** | |

|  |  |
| --- | --- |
| * + - 1. **채팅 프로그램은 사용자를 구별하기 위해 가입과 로그인 기능을 제공한다** | **ID:** FR6 |
| **Priority:** 2 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 사용자는 가입을 하고 로그인 기능을 통해 채팅 프로그램을 실행 할 수 있다. | |
| **Open Issues** | |

|  |  |
| --- | --- |
| * + - 1. **사용자가 기존 채팅방에 참여할 수 있는 기능을 제공한다** | **ID:** FR7 |
| **Priority:** 2 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 사용자는 기존 채팅방이 존재한다면 채팅방에 참여할 수 있다. 만약 비공개 채팅방이라면 비밀번호를 입력해야 입장할 수 있다. | |
| **Open Issues** | |

|  |  |
| --- | --- |
| * + - 1. **사용자가 참여한 채팅방에서 나올 수 있는 기능을 제공한다** | **ID:** FR8 |
| **Priority:** 2 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 사용자가 참여한 채팅방에서 나가고 싶다면, 채팅방에 나올 수 있다. | |
| **Open Issues** | |

|  |  |
| --- | --- |
| * + - 1. **관리자는 채팅 프로그램을 관리한다** | **ID:** FR9 |
| **Priority:** 3 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| 관리자는 채팅방 사용자, 사용자의 정보, 채팅방 정보 등 시스템을 관리한다. | |
| **Open Issues** | |

* + 1. **Specifications Traceability Matrix**
* Traceability matrix는 기능적 요구사항들 사이의 의존성들을 나타낸다. 의존성들은 기능적 요구사항에 언급된 명시적 의존성과 문서화된 기능들에 의해 암시된 암시적 의존성을 기반으로 나타낸다.

Table 4 Specifications Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **FR1** | **FR2** | **FR3** | **FR4** | **FR5** | **FR6** | **FR7** | **FR8** | **FR9** |
| **FR1** | **-** | **O** | **O** | **O** | **O** | **O** | **O** | **O** |  |
| **FR2** | **O** | **-** |  | **O** |  | **O** |  |  |  |
| **FR3** | **O** |  | **-** |  |  |  | **O** |  |  |
| **FR4** | **O** | **O** |  | **-** | **O** |  | **O** |  |  |
| **FR5** | **O** |  |  | **O** | **-** |  |  |  |  |
| **FR6** | **O** | **O** |  |  | **O** | **-** | **O** |  |  |
| **FR7** | **O** |  | **O** | **O** |  | **O** | **-** |  |  |
| **FR8** | **O** |  |  |  |  |  |  | **-** |  |
| **FR9** |  |  |  |  |  |  |  |  |  |

* + 1. **Use Case Diagram**

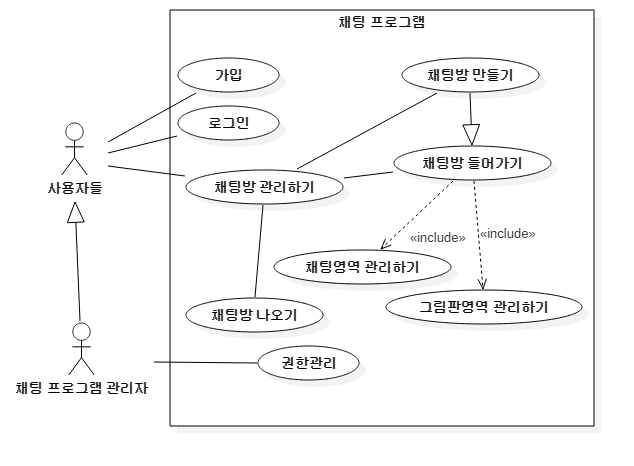


Figure 1. Use case diagram

* + 1. **Use Case List**

Table 5. Use Case List of Chatting program

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID** | | **UC Title** | **FR ID** |
| UC 1 | | 가입 | FR6 |
| UC 2 | | 로그인 | FR6 |
| UC 3 | | 채팅방 관리하기 |  |
|  | UC 3-1 | 채팅방 만들기 | FR1, FR2 |
|  | UC 3-2 | 채팅방 들어가기 | FR1, FR7 |
|  | UC 3-3 | 채팅방 나가기 | FR8 |
| UC 4 | | 채팅영역 관리하기 | FR1, FR4, FR5 |
| UC 5 | | 그림판영역 관리하기 | FR1, FR3 |
| UC 6 | | 권한관리 | FR9 |

* + 1. **Use Cases**
       1. **UC 1 가입**

|  |  |
| --- | --- |
| **Use case title:** 가입 | **Use case ID:** UC 1 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| **Scope** | |
| 채팅 프로그램 | |
| **Level** | |
| 사용자-goal | |
| **Primary Actor** | |
| 사용자 | |
| **Stakeholders and Interest** | |
| 사용자: 채팅 프로그램을 사용하기 위해 가입하길 원한다. | |
| **Preconditions** | |
| * 채팅 프로그램의 서버가 작동하고 있다. | |
| **Success Guarantee (Post conditions)** | |
| * 사용자가 가입한다. | |
| **Main Success Scenario** | |
| 1. 사용자는 채팅 프로그램을 실행한다. 2. 채팅 프로그램의 시작화면을 보여준다. 3. 사용자는 가입하기 버튼을 클릭한다. 4. 사용자는 가입을 위한 아이디, 비밀번호, 이메일을 입력한다. 5. 사용자는 가입 버튼을 클릭한다. 만약 가입하려는 아이디가 있다면 다른 아이디를 입력한 후 가입 버튼을 클릭한다. | |
| **Extension – Alternative Scenario** | |
| * N/A | |
| **Additional Requirements** | |
| * N/A | |
| **Technology and Data Variations List** | |
| * 사용자는 지정된 형식에 맞게 아이디와 비밀번호를 입력해야 한다. (특수문자 제한) | |

* + - 1. **UC 2 Manage the payment**

|  |  |
| --- | --- |
| **Use case title:** 로그인 | **Use case ID:** UC 2 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| **Scope** | |
| 채팅 프로그램 | |
| **Level** | |
| 사용자-goal | |
| **Primary Actor** | |
| 사용자 | |
| **Stakeholders and Interest** | |
| 사용자: 채팅 프로그램을 사용하기 위해 로그인하길 원한다. | |
| **Preconditions** | |
| * 사용자는 가입한 상태이다. * 채팅 프로그램의 서버가 작동하고 있다. | |
| **Success Guarantee (Post conditions)** | |
| * 사용자는 로그인한다. | |
| **Main Success Scenario** | |
| 1. 사용자는 채팅 프로그램을 실행한다. 2. 채팅 프로그램의 시작화면을 보여준다. 3. 사용자는 아이디와 비밀번호를 입력한다. 4. 사용자는 로그인하기 버튼을 클릭한다. | |
| **Extension – Alternative Scenario** | |
| 1. 아이디와 비밀번호가 일치하지 않는다. 2. 불일치 알림을 띄워준다. | |
| **Additional Requirements** | |
| * N/A | |
| **Technology and Data Variations List** | |
| * N/A | |

* + - 1. **UC 3 채팅방 관리하기**

|  |  |
| --- | --- |
| **Use case title:** 채팅방 만들기 | **Use case ID:** UC 3-1 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| **Scope** | |
| 채팅 프로그램 | |
| **Level** | |
| 사용자-goal | |
| **Primary Actor** | |
| 사용자 | |
| **Stakeholders and Interest** | |
| 사용자: 채팅방 만들기를 원한다. | |
| **Preconditions** | |
| * 채팅 프로그램의 서버가 작동하고 있다. * 사용자는 로그인한 상태이다. | |
| **Success Guarantee (Post conditions)** | |
| * 사용자가 채팅방을 만들고 채팅방에 입장한다. | |
| **Main Success Scenario** | |
| 1. 사용자가 채팅 관리 화면에서 채팅방 만들기 버튼을 클릭한다. 2. 채팅 프로그램은 채팅방 만들기 화면을 보여준다. 3. 사용자가 채팅방 이름, 공개여부, 비공개라면 비밀번호를 입력한다. 4. 사용자가 완료 버튼을 클릭한다. | |
| **Extension – Alternative Scenario** | |
| * N/A | |
| **Additional Requirements** | |
| * N/A | |
| **Technology and Data Variations List** | |
| * 사용자는 지정된 형식에 맞게 채팅방 이름, 공개여부, 비밀번호를 입력해야 한다. (특수문자 제한) | |

|  |  |
| --- | --- |
| **Use case title:** 채팅방 들어가기 | **Use case ID:** UC 3-2 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| **Scope** | |
| 채팅 프로그램 | |
| **Level** | |
| 사용자-goal | |
| **Primary Actor** | |
| 사용자 | |
| **Stakeholders and Interest** | |
| 사용자: 기존 채팅방에 들어가길 원한다. | |
| **Preconditions** | |
| * 채팅 프로그램의 서버가 작동하고 있다. * 사용자는 로그인한 상태이다. * 참여할 채팅방이 존재한다. | |
| **Success Guarantee (Post conditions)** | |
| * 사용자가 선택한 채팅방에 들어간다. | |
| **Main Success Scenario** | |
| 1. 사용자는 채팅 관리 화면에서 원하는 채팅방을 클릭한다. 2. 채팅 관리 화면은 사용자가 클릭한 방의 정보를 보여준다. 3. 비공개 채팅방인 경우 사용자가 비밀번호를 입력 후 채팅방 들어가기 버튼을 클릭한다. | |
| **Extension – Alternative Scenario** | |
| 1. 채팅방의 비밀번호가 불일치 할 경우, 불일치 알람을 보여준다. | |
| **Additional Requirements** | |
| * N/A | |
| **Technology and Data Variations List** | |
| * N/A | |

|  |  |
| --- | --- |
| **Use case title:** 채팅방 나가기 | **Use case ID:** UC 3-3 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| **Scope** | |
| 채팅 프로그램 | |
| **Level** | |
| 사용자-goal | |
| **Primary Actor** | |
| 사용자 | |
| **Stakeholders and Interest** | |
| 사용자: 참여한 채팅방에서 나오길 원한다. | |
| **Preconditions** | |
| * 채팅 프로그램의 서버가 작동하고 있다. * 사용자는 로그인한 상태이다. * 사용자는 참여한 채팅방이 있다. | |
| **Success Guarantee (Post conditions)** | |
| * 사용자가 참여한 채팅방에서 나온다. | |
| **Main Success Scenario** | |
| 1. 사용자는 참여한 채팅방에서 채팅방 나가기 버튼을 클릭한다. | |
| **Extension – Alternative Scenario** | |
| * N/A | |
| **Additional Requirements** | |
| * N/A | |
| **Technology and Data Variations List** | |
| * N/A | |

* + - 1. **UC 4 채팅영역 관리하기**

|  |  |
| --- | --- |
| **Use case title:** 채팅영역 관리하기 | **Use case ID:** UC 4 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| **Scope** | |
| 채팅 프로그램 | |
| **Level** | |
| 사용자-goal | |
| **Primary Actor** | |
| 사용자 | |
| **Stakeholders and Interest** | |
| 사용자: 채팅영역에 내용을 입력하길 원한다. | |
| **Preconditions** | |
| * 채팅 프로그램의 서버가 작동하고 있다. * 사용자는 로그인한 상태이다. * 사용자는 참여한 채팅방이 있다. | |
| **Success Guarantee (Post conditions)** | |
| * 채팅영역에 내용이 입력된다. | |
| **Main Success Scenario** | |
| 1. 사용자는 채팅영역의 입력란에 입력할 내용을 입력한다. 2. 사용자는 입력 버튼을 클릭한다. | |
| **Extension – Alternative Scenario** | |
| * N/A | |
| **Additional Requirements** | |
| * N/A | |
| **Technology and Data Variations List** | |
| * N/A | |

* + - 1. **UC 5 그림판영역 관리하기**

|  |  |
| --- | --- |
| **Use case title:** 그림판영역 관리하기 | **Use case ID:** UC 5 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| **Scope** | |
| 채팅 프로그램 | |
| **Level** | |
| 사용자-goal | |
| **Primary Actor** | |
| 사용자 | |
| **Stakeholders and Interest** | |
| 사용자: 그림판영역에 내용을 입력하길 원한다. | |
| **Preconditions** | |
| * 채팅 프로그램의 서버가 작동하고 있다. * 사용자는 로그인한 상태이다. * 사용자는 참여한 채팅방이 있다. | |
| **Success Guarantee (Post conditions)** | |
| * 그림판영역에 내용이 입력된다. | |
| **Main Success Scenario** | |
| 1. 사용자는 그림판영역에 입력할 도구를 선택한다. 2. 사용자는 그림판영역에 마우스를 이용하여 내용을 입력한다. | |
| **Extension – Alternative Scenario** | |
| * N/A | |
| **Additional Requirements** | |
| * 그림판영역의 기본 도구는 검은펜으로 되어 있다. | |
| **Technology and Data Variations List** | |
| * N/A | |

* + - 1. **UC 6 권한관리**

|  |  |
| --- | --- |
| **Use case title:** 권한관리 | **Use case ID:** UC 6 |
| **Version:** 1 |
| **Last Changed:** 04/11/2016 |
| **Scope** | |
| 채팅프로그램 | |
| **Level** | |
| 관리자-goal | |
| **Primary Actor** | |
| 관리자 | |
| **Stakeholders and Interest** | |
| 관리자: 채팅 프로그램을 관리하길 원한다. | |
| **Preconditions** | |
| * 채팅 프로그램의 서버가 작동되고 있다. | |
| **Success Guarantee (Post conditions)** | |
| * 관리자가 채팅프로그램을 관리한다. | |
| **Main Success Scenario** | |
| 1. 관리자는 관리자 페이지로 들어간다. 2. 관리자는 사용자관리 채팅방관리를 한다. | |
| **Extension – Alternative Scenario** | |
| * N/A | |
| **Additional Requirements** | |
| * N/A | |
| **Technology and Data Variations List** | |
| * N/A | |

* 1. **Quality Attribute Requirements**
* 품질 속성은 시스템의 환경에서 시스템의 의도된 행동을 묘사하는 기준점이다. 이는 제품의 적합성을 측정하기 위한 수단을 제공한다. 이 요구사항은 제품의 아키텍처 설계와 개발에 대해 영향을 미칠 것이다.
  + 1. **Quality Attributes List**

Table 6. Quality attributes list of Chatting program

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **UC #** | **Ability** | **Description** | **Priority  H/M/L** |
| QA-S-01 | UC 3-2 | Scalability | N명의 사용자는 채팅방에 들어갈 수 있다. | H |
| QA-S-02 | UC 1 | Scalability | N명의 사용자는 채팅 프로그램에 가입할 수 있다. | M |
| QA-S-03 | UC 3-1 | Scalability | 최대 사용자만큼 채팅방을 만들 수 있다. | L |
| QA-M-01 | All UC | Maintainability | 추가적인 기능이 제시될 때 채팅 프로그램을 수정하기 편하다. (낮은 결합도, 높은 응집도) | H |
| QA-P-01 | UC 4  UC 5 | Performance | 사용자는 실시간으로 그림판영역의 화면과 채팅영역을 공유한다. | H |
| QA-U-01 | All UC | Usability | 사용자가 채팅 프로그램을 사용하기위한 기능을 읽히는 시간은 한 번 사용으로 가능하다. | M |

* + 1. **Quality attribute scenarios**
       1. **Template**
* 아래 품질 속성 시나리오는 설계작업을 하는 동안 아키텍처, 세부적 설계 그리고 구현 과정에서 더 정확한 아이디어가 나왔을 때 수정이 가능하다. 따라서 설계작업 동안 새로운 품질 속성 시나리오가 추가될 수 있다.

|  |  |  |
| --- | --- | --- |
| **Title of scenario:** Descriptive title if desired | | **ID:** Mnemonic reference if desired |
| **Quality attribute:** The one-word quality attribute characterization | **Characterization ID:** The ID of the stage 1 characterization used as a basis for this scenario | |
| **Describe stakeholder role proposing the description:** The stakeholder(s) or stakeholder communities interested in this quality attribute scenario | | |
| **Source(s) of the stimulus** | Description of the originating source(s) or potential sources of the stimulus or stimuli that will result in a system response. | |
| **Stimulus** | Phenomenon, event, situation, etc., that prompts the system or stakeholders to react in some way or take some action. It might be a user request, an event or interrupt, an error, a request for change, and so forth. | |
| **Environment** | Description of any relevant environmental conditions that could affect the response and the measures of the response. It might include such conditions as normal operation, peak load, degraded operation, at development time, during operation, without interrupting operation, and so forth. | |
| **Artifact** | System elements affected by the stimulus. Early in the development life cycle (pre-architectural design) the affected elements might not be known or may be a very high-level description. As design commences, scenarios should be refined and elements identified as they are known. | |
| **System response** | Desired response of the system. This should be as specific as possible because this will be how the system will be designed to respond to the stimulus. | |
| **Response measure(s)** | Measure by which the quality of the response will be measured. Response measures depend on the quality attribute. Response measures vary and may be in terms of person-hours, error detection, response time, recovery time, and so forth. | |

* + - 1. **Scalability**

|  |  |  |
| --- | --- | --- |
| **Title of scenario:** N명의 사용자는 채팅방에 들어갈 수 있다. | | **ID:** QA-S-01 |
| **Quality attribute** | Scalability | |
| **Describe stakeholder role proposing the description :** 사용자, LIG넥스원 관계자 | | |
| **Source(s) of the stimulus** | 사용자 | |
| **Stimulus** | 채팅방에 들어가려 한다 | |
| **Environment** | 정상 작업 | |
| **Artifact** | 채팅방에 최대 N명의 사용자가 들어온다 | |
| **System response** | 채팅 프로그램은 최대 N명의 사용자들이 채팅방에 들어오도록 한다 | |
| **Response measure(s)** | 채팅방에 최대 N명이 들어올 수 있다 | |

|  |  |  |
| --- | --- | --- |
| **Title of scenario:** N명의 사용자는 채팅 프로그램에 가입할 수 있다 | | **ID:** QA-S-02 |
| **Quality attribute** | Scalability | |
| **Describe stakeholder role proposing the description :** 사용자, 관리자, LIG넥스원 관계자 | | |
| **Source(s) of the stimulus** | 사용자 | |
| **Stimulus** | 채팅 프로그램에 가입하려 한다 | |
| **Environment** | 정상 작업 | |
| **Artifact** | 채팅 프로그램에 최대 N명의 사용자가 가입한다 | |
| **System response** | 채팅 프로그램은 최대 N명의 사용자들이 가입할 수 있도록 한다 | |
| **Response measure(s)** | 채팅 프로그램에 최대 N명이 가입할 수 있다 | |

|  |  |  |
| --- | --- | --- |
| **Title of scenario:** N명의 사용자만큼 채팅방을 만들 수 있다 | | **ID:** QA-S-03 |
| **Quality attribute** | Scalability | |
| **Describe stakeholder role proposing the description :** 사용자, 관리자, LIG넥스원 관계자 | | |
| **Source(s) of the stimulus** | 사용자 | |
| **Stimulus** | 채팅방을 만들려 한다 | |
| **Environment** | 정상 작업 | |
| **Artifact** | 채팅 프로그램에 최대 N개의 채팅방을 만든다 | |
| **System response** | 채팅 프로그램은 최대 N개의 채팅방을 만들 수 있도록 한다 | |
| **Response measure(s)** | 채팅 프로그램에 최대 N개의 채팅방을 만들 수 있다 | |

* + - 1. **Maintainability**

|  |  |  |
| --- | --- | --- |
| **Title of scenario:** 추가적인 기능이 제시될 때 채팅 프로그램을 수정하기 편하다 (낮은 결합도, 높은 응집도) | | **ID:** QA-M-01 |
| **Quality attribute** | Maintainability | |
| **Describe stakeholder role proposing the description :** 개발자, LIG넥스원 관계자 | | |
| **Source(s) of the stimulus** | 개발자 | |
| **Stimulus** | 추가적인 기능을 개발하려 한다 | |
| **Environment** | 정상 작업 | |
| **Artifact** | 낮은 결합도와 높은 응집도로 프로그램을 개발한다 | |
| **System response** | 개발자가 프로그램 수정을 편하게 한다 | |
| **Response measure(s)** | 불필요한 수정 또는 디버그 필요없이 개발자가 요구사항 문서 또는 설계 문서를 보고 수정할 수 있다 | |

* + - 1. **Performance**

|  |  |  |
| --- | --- | --- |
| **Title of scenario:** 사용자는 실시간으로 그림판영역의 화면과 채팅영역을 공유한다 | | **ID:** QA-P-01 |
| **Quality attribute** | Performance | |
| **Describe stakeholder role proposing the description :** 사용자, LIG넥스원 관계자 | | |
| **Source(s) of the stimulus** | 사용자 | |
| **Stimulus** | 그림판영역 또는 채팅영역에 데이터를 입력한다 | |
| **Environment** | 정상 작업 | |
| **Artifact** | 그림판영역 또는 채팅영역에 사용자가 입력한 데이터가 보여진다 | |
| **System response** | 채팅 프로그램은 사용자가 입력한 데이터를 받아서 그림판영역 또는 채팅영역에 보여준다 | |
| **Response measure(s)** | 채팅 프로그램은 실시간으로 입력내용을 공유한다 | |

* + - 1. **Usability**

|  |  |  |
| --- | --- | --- |
| **Title of scenario:** 사용자가 채팅 프로그램을 사용하기 위한 기능을 읽히는 시간은 한 번 사용으로 가능하다 | | **ID:** QA-U-01 |
| **Quality attribute** | Usability | |
| **Describe stakeholder role proposing the description :** 사용자, 개발자 | | |
| **Source(s) of the stimulus** | 사용자 | |
| **Stimulus** | 채팅 프로그램을 사용하려 한다 | |
| **Environment** | 정상 작업 | |
| **Artifact** | 도움말없이 사용법을 읽힌다 | |
| **System response** | 사용자가 채팅 프로그램을 한 번 사용하고 사용법을 읽힌다 | |
| **Response measure(s)** | 사용자가 채팅 프로그램 한 번 사용으로 모든 기능을 읽힌다 | |

* 1. **Constraints**
* LIG넥스원 과제인 채팅 프로그램에는 비즈니스 제약과 기술적 제약 두가지 제약이 있다. 비즈니스 제약은 예산, 인력 및 장비 등의 비즈니스 목표를 달성하기 위해 재정, 시간 그리고 물리적 제한이 있다. 기술적 제약은 표준, 프로토콜, 개발 언어 등 프로젝트를 수행하면서 이미 결정된 설계들이 있다.
  + 1. **Business Constraints**

Table 7. Business constraints of Chatting program

|  |  |
| --- | --- |
| Consideration | Business Constraint |
| Schedule Mandates | * 채팅 프로그램은 2016년 11월 4일부터 2016년 11월 8일 화요일 오후 6시까지 요구사항에 맞게 개발하고 개발 문서와 함께 제출해야 한다. |
| Budget Limitations | * 요구사항 도출과 문서작성 그리고 개발까지 지원자 이병욱 혼자 한다. * 하드웨어 제약사항으로 채팅 프로그램을 테스팅할 때 3~5명의 사용자로 테스트한다. |

* + 1. **Technical Constraints**

Table 8. Technical constraints of Chatting program

|  |  |
| --- | --- |
| Consideration | Technical Constraint |
| Computer language | * 채팅 프로그램은 Visual C++ (MFC)로 개발해야 한다. * 통합개발환경은 Visual Studio를 사용해야 한다. |
| Protocols, interfaces, standards | * 채팅 프로그램의 기본 아키텍처는 클라이언트/서버 아키텍처이다. |

1. **Prioritization**
   1. **Priority scale**

|  |  |  |
| --- | --- | --- |
| **Priority**  **(numeric)** | **Priority**  **(name)** | **Description** |
| **1** | Must Have | Must be present in the end-product at all costs. |
| **2** | Nice to Have | Customer would greatly appreciate implementation of these features. |
| **3** | If There’s Time | Consider in Fall 09 semester if customer deems them important enough. |

* 1. **Difficulty Ranking scale**
* The difficulty scale has been defined on the basis of complexity and effort. Complexity is defined as how difficult the design of a solution is and whether the team has previous experience in designing or implementing such a design. Both measures, for complexity and effort, are relative to each other.

|  |  |
| --- | --- |
| **Difficulty**  **(numeric)** | **Description** |
| **1** | High complexity and large amount of effort required |
| **2** | High complexity or large amount of effort required |
| **3** | Moderate complexity and medium amount of effort required |

* 1. **Use Cases**

|  |  |  |
| --- | --- | --- |
| **Use Case prioritizations** | | |
| **ID & Title** | **Stakeholder priority** | **Difficulty ranking** |
| **UC 1 가입** | 3 | 3 |
| **UC 2 로그인** | 3 | 3 |
| **UC 3 채팅방 관리하기** | 2 | 2 |
| **UC 3-1 채팅방 만들기** | 2 | 2 |
| **UC 3-2 채팅방 들어가기** | 2 | 2 |
| **UC 3-1 채팅방 나가기** | 2 | 2 |
| **UC 4 채팅영역 관리하기** | 1 | 1 |
| **UC 5 그림판영역 관리하기** | 1 | 1 |
| **UC 6 권한관리** | 3 | 1 |

* 1. **Quality Attribute Scenarios**

|  |  |  |
| --- | --- | --- |
| **Quality Attributes Scenario prioritizations** | | |
| **ID** | **Stakeholder priority** | **Difficulty ranking** |
| **QA-S-1** | 1 | 3 |
| **QA-S-02** | 3 | 3 |
| **QA-S-03** | 3 | 3 |
| **QA-M-01** | 1 | 1 |
| **QA-P-01** | 1 | 1 |
| **QA-U-01** | 3 | 1 |

* 1. **Constraints**
* Constraints have not been prioritized in terms of importance because by definition they are of the highest priority. We have however prioritized them by difficulty, as judged by the development team.
  + 1. **Business Constraints**

|  |  |
| --- | --- |
| **Business constraint prioritizations** | |
| **Description** | **Difficulty ranking** |
| Schedule Mandates | 1 |
| Budget Limitations | 1 |

* + 1. **Technical Constraints**

|  |  |
| --- | --- |
| **Technical constraint prioritizations** | |
| **Description** | **Difficulty ranking** |
| Computer language | 1 |
| Protocols, interfaces, standards | 2 |

* 1. **Significant Architectural Drivers**
* The following architectural drivers have been given a high priority by the customer and have been ranked by the engineering team as being of high difficulty. Thus these should be given preference when making any decisions that affect architectural drivers.

|  |  |
| --- | --- |
| **Architectural Drivers** | **Comments** |
| **UC3– 채팅방 관리하기** | 채팅방 관리는 채팅 프로그램의 기본적인 기능적 요구사항이다. 그렇기 때문에 전체적인 아키텍처는 채팅방 관리를 고려하여야 한다. |
| **UC4– 채팅영역 관리하기** | 채팅영역 관리는 고객의 요구사항으로 채팅 프로그램에서 충족시켜야 하는 아주 중요한 기능적 요구사항이다. 실시간으로 채팅영역을 공유해야 하기 때문에 이를 고려하여 개발해야 한다. |
| **UC5– 그림판영역 관리하기** | 그림판영역 관리는 고객의 요구사항으로 채팅 프로그램에서 충족시켜야 하는 아주 중요한 기능적 요구사항이다. 실시간으로 그림판영역을 공유해야 하기 때문에 이를 고려하여 개발해야 한다. |

1. **Risk Management**
   1. **Risk Management Process**

* The project manager working with the project team and project sponsors will ensure that risks are actively identified, analyzed, and managed throughout the life of the project. Risks will be identified as early as possible in the project so as to minimize their impact. The steps for accomplishing this are outlined in the following sections. The <project manager or other designee> will serve as the Risk Manager for this project.
  1. **Risk Identification & Analysis**
     1. **Qualitative Risk Analysis**

Table 9. Identified and analyzed risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Risk | Probability[[1]](#footnote-1) | Impact[[2]](#footnote-2) | Risk rating[[3]](#footnote-3) |
| Technology | Manufacturing failure | 2 | 5 | Moderate |
| People | Communication failure | 1 | 1 | Low |
| Requirements | Requirements change | 1 | 4 | Low |
| Estimation | Product competition | 5 | 1 | Moderate |
| Estimation | Specification delay | 3 | 5 | High |
| Estimation | Size under estimate | 3 | 4 | Moderate |

* 1. **Risk Response Planning**

Table 10. Response planning of analyzed risks

|  |  |  |  |
| --- | --- | --- | --- |
| Risk rating | Risk | Response Type | Response Plan |
| High | Specification delay | Avoid | Adding more resources for specification |
| Moderate | Manufacturing failure | Mitigate | Save the all intermediate results and encourage pair-programming |
| Moderate | Size under estimate | Avoid | Review the architecture design iteratively |
| Moderate | Size under estimate | Avoid | Review the architecture design iteratively |
| Low | Communication failure | Avoid | Make a meeting to decrease the difference of opinion |
| Low | Requirements change | Mitigate | Make a prototype to make client understand about system and make communications with client iteratively |

1. **Architecture**
   1. **Architecture Design Decision**
      1. **Client Server Pattern**
         1. **Main Architecture Pattern Decision**
            1. **Reasoning**

* 채팅 프로그램은 사용자에게 제공하는 어플리케이션과 채팅방을 관리하는 관리자 어플리케이션으로 구분하여 구현한다. 서버에서는 여러 사용자, 그리고 채팅방의 데이터를 관리하고 클라이언트에서는 서버에서 받아온 데이터를 이용하여 사용자에게 보여주고 또한 사용자가 입력한 데이터를 서버에 전달한다.
  + 1. **Database Types**
       1. **Pros and Cons**

1. SQL Type
   * Pros : Easy to define schema. Provides centralized control
   * Cons : Hard to manage very large scale data.
2. No SQL
   * Pros : Provides distributed control
   * Cons : Hard to manage.
     + 1. **Decision**
3. SQL Type
   * 채팅 프로그램 서버에서 관리할 데이터는 만들어진 채팅방, 들어온 사용자와 같이 실시간으로 제공되어야 한다. 그러므로 우리는 대용량의 데이터 관리가 필요하지 않기 때문에 SQL type을 사용한다.
     1. **Database Management System**
        1. **Pros and Cons**
4. Mysql
   * Pros : Open source, developers are familiar with.
5. MongoDB
   * Pros : Open source, developers are familiar with.
6. Others
   * Cons : Developers are not experienced with those environments.
     + - 1. **Decision**
7. Mysql

* 채팅 프로그램의 사용자 관리와 채팅방 관리를 위해 사용한다.
  1. **Context Diagram**

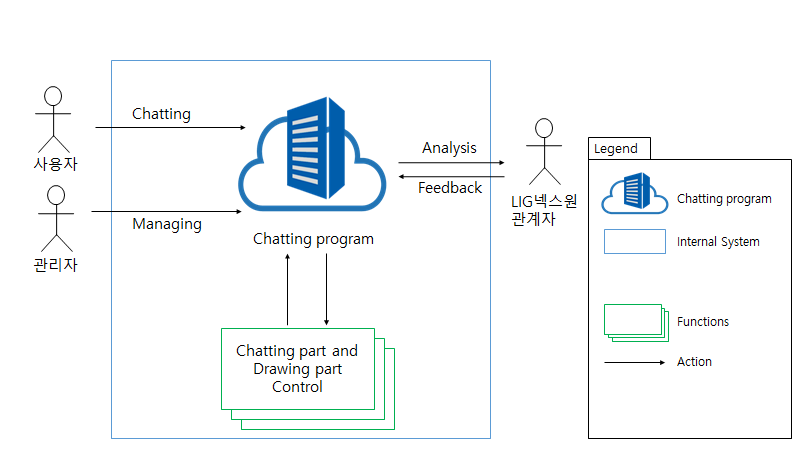


Figure 2. Context diagram of Chatting program

* Figure 2는 채팅 프로그램의 컨텍스트 다이어그램을 보여준다. 사용자는 채방 프로그램을 통해 채팅을, 관리자는 관리를 LIG넥스원 관계자는 전체 프로그램을 평가한다.
  1. **Physical View**
     1. **Deployment View**

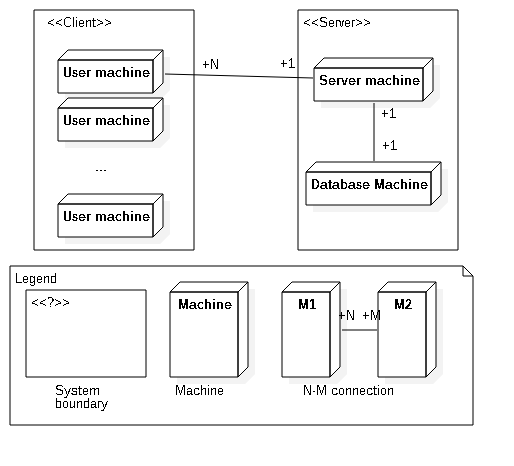


Figure 3. Deployment view of Chatting program

* 1. **Static View**
     1. **Module View**

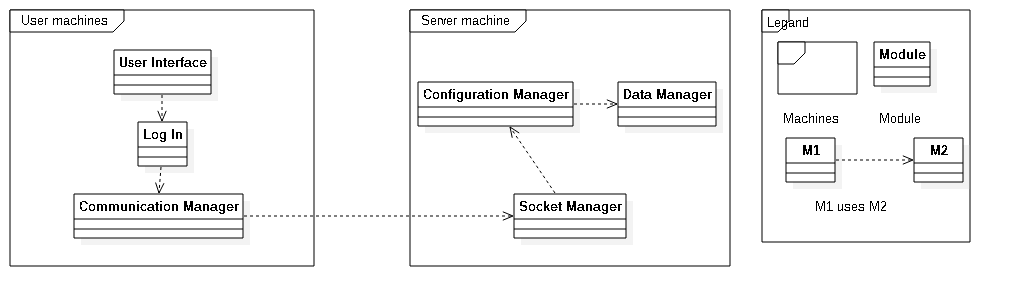


Figure 4. Module view of Chatting program

* 1. **Dynamic View**
     1. **Components and Connectors View**

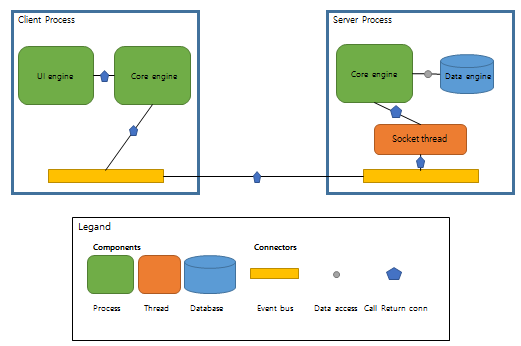


Figure 5. High Level C&C view of Chatting program

1. **Requirements Traceability Matrix**

Table 11 Requirements Traceability Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement satisfied** | **Design decision** | **Element** | **Relationship** | **Comments and descriptions** |
| QA-A-01 | Pipe and Filter Pattern | Module independency | Modules are independent with each other | Because of independency between each module, developer can add new statistics algorithms. |
| QA-A-02 | This quality attribute is related to whole of system availability, and as such does not have an influence on the software architecture. | | | |
| QA-SC-01 | Publish-Subscribe Pattern, Server-Client Pattern | Parking controller and local server | Parking controller and local server consist of Publish-Subscribe pattern | Register several parking controllers. Local server subscribes the event of the registration and parking controller publishes the data to local server. |
| QA-P-01 | Caching | Web server and local server, parking controller | Web server shared parking lot data | Web server and local server shared data. So, parking controller be able to response quickly |
| QA-P-02 | Caching | Web server, database | Data caching in the database | The data can be served in smaller chunks or as a whole, depending on user requirements. |
| QA-P-03 | State Management | Parking controller | Local server has stateless | Depend on state, server can response immediately |
| QA-P-04 | Caching | Administration application, database | Data caching in the database | The data can be served in smaller chunks or as a whole, depending on owner requirements. |
| QA-P-05 | Caching | Web server, database | Data caching in the database | The data can be served in smaller chunks or as a whole, depending on owner requirements. |
| QA-SE-01 | Two-factor authentication | Owner’s account | Owner’s account depends on two-factor authentication | By using two-factor authentication, system can detect |
| QA-SE-02 | Authentication | Attendant’s account, owner’s account | Accounts of attendant and owner should be accessed through authentications | To access accounts of attendant and owner, user should get through the authentication. Attendants should know their id and password, owner should pass through two-factor authentication. |
| QA-U-01 | This quality attribute is related the user interface, and as such does not have an influence on the software architecture | | | |
| QA-U-02 | OAuth authorization token. | Validation process | Validation process depends on OAuth authorization token. | Drivers are able to enter parking lot with easy validation process by using OAuth authorization token. |
| QA-U-03 | This quality attribute is related to the layout of the user interface, and as such does not have an influence on the software architecture | | | |
| QA-U-04 | Lighting up LED | LED sensor, gate | LED sensor depends on gate sensor | When the gate is opened, to see where the driver should park intuitively, LED above the right parking spot is lighted up. |
| QA-I-01 | Database exchange | database of parking spot | In the system, The database of correct parking spot is exchanged with database of wrong parking spot. | When a driver parks his car at wrong parking spot, system just exchange the database of correct spot with database of wrong spot. It does not yield waste of manpower. |
| QA-SA-01 | Socket connection, | Session | Session depends on socket connection | To distinguish whether connection between local server and Arduino, socket connection is used to keep session. When the socket connection is broken, Arduino realizes itself that the connection is broken. It can reduce overhead than method of heartbeat |
| UC 1-1 | Server-Client Pattern, Tomcat, SQL type database, Android Application | Web server, database | Reservation data is based on SQL type | Reservation data will be removed from table as soon as reservation invalidated. Therefore, we do not need large scale data management for dealing with reservation data |
| UC 1-2 | Android Application, SQL type database | User application,  database | User can use android application for checking reservation | It is easy to compose user interface |
| UC 2 | HTTPS, TCP, OAuth2 | Parking controller, Web server, | Protocol has better security than other protocol | It is used to provide better security. |
| UC 3-1 | HTTPS, TCP, OAuth2, SQL type database | Parking controller, Web server, database | Reservation data is based on SQL type | It is used to provide better security. Reservation data will be removed from table as soon as reservation invalidated and change the reservation information easily |
| UC 3-2 | SQL type database | Local server, database | Reservation data is based on SQL type | The system can manage the database without disrupting |
| UC 4-1 | SQL type database | database | Reservation data is based on SQL type | Reservation data will be removed from table as soon as reservation invalidated. Therefore, we do not need large scale data management for dealing with reservation data |
| UC 4-2 | No SQL Type database | statistics | Statistics are made according to No SQL database(Mongo DB) | No SQL Type database is used to analyze accumulated log data of customers for parking lot |
| UC 5 | .NET C# WPF application | Monitoring application | Monitoring application is developed using .NET C# WPF application | It is easy to use to develop application. |
| UC 6-1 | Two-factor authentication | Owner’s account | Owner’s account depends on two-factor authentication | By using two-factor authentication, system can detect |
| UC 6-2 | No SQL Type database | statistics | Statistics are made according to No SQL database(Mongo DB) | No SQL Type database is used to manage accumulated log data of customers for parking lot |
| UC 6-3 | Publish-Subscribe pattern | parking controller, local server | Parking controller is registered through publish-subscribe way | We select Publish-Subscribe patterns in order to register several parking controller. Local server subscribes the event of the registration and parking controller publishes the data to local server. |

1. **Description of how the system will (or will not) satisfy the future needs listed in the project description**

* The Sure-Park system we developed satisfies the initial key motivations described in project descriptions. The system reduces the driver frustration by providing intuitive user interface. Simple design makes driver to make reservation easily. Moreover, when driver enters the entrance gate, LED which is above the proper parking spot is lighted up to help driver to know where they should park their vehicle. It may reduce traffic congestion and the chance for accidents inside the parking facilities. And none-overlapped reservations which is done by more than two drivers are blocked. The system does not allow overlapped reservations doing real time synchronization.

The system automation is worthy to get attention. We have pursued a future-oriented automation system to reduce redundant people required to operate any given garage reducing operating costs. All the operation of facility, gate open, checking vacancy, receiving fee and so forth are done automatically by the system. It needs just minimum people to cope with unexpected situations.

And data analysis on average occupancy, peak usage ours, revenue, car size and so forth can utilize the parking space. Then any configuration for grace-period, available time to make reservation, lane mark and so forth can be changed according to the analysis.

Also it is scalable due to our architectural decision. Our system comprises web server and local server. This server-client pattern make it easier to add additional servers. It makes owner to decide their business structure with flexibility.

1. The probability of the risk: Almost certain(>90% chance; 5), Likely(between 50% and 90% chance; 4), Moderate(between 10% and 50% chance; 3), Unlikely(between 3% and 10% chance; 2), or Rare(<3% chance; 1) [↑](#footnote-ref-1)
2. The impacts of the risk: Catastrophic(Business survival is impossible; 5), Major(Operations severely damaged; 4), Moderate(Significant time/resources required; 3), Minor(Some disruption possible; 2) or Insignificant(Minor problem easily handled by normal day to day processes; 1) [↑](#footnote-ref-2)
3. The rating of the risk: probability \* impact; low(<=4, tolerable level), moderate(between 5 and 12, meddle level), or High(>=15, intolerable level) [↑](#footnote-ref-3)