

Requirement Specification

Team 14

2017310022 나경호

2017314698 김동락

2017314648 박세환

2016315178 이기요

2017314519 이은지

2016312961 조민재

Table of Contents

1. Preface	7
1.1. Readership	7
A. User Requirement Readership	7
B. System Requirement Readership	7
1.2. Document Structure	7
A. Introduction	7
B. Glossary.....	8
C. User Requirement Definition	8
D. System Architecture.....	8
E. System Requirements	8
F. System Models	8
G. System Evolution	8
H. Appendices	8
I. Index	8
J. Reference	8
2. Introduction.....	9
2.1. Needs.....	9
2.2. System Overview.....	10
2.3. Expected Effect	10
A. Replace face-to-face interview	10
B. Easy to use & More visual.....	10

C. Real-Time	10
3. Glossary	11
4. User Requirements Definition.....	12
4.1. Functional Requirements.....	12
A. White Board	12
B. YouTube Player.....	12
C. Office Information Showing.....	12
D. Office Password.....	13
E. Office Transferring	13
F. Office Locking	13
G. Ticketing System (Ticketing)	13
H. Ticketing System (Calling)	13
4.2. Non-functional Requirements.....	14
A. Product Requirements.....	14
1) Usability.....	14
2) Security.....	14
3) Reliability	14
4) Performance	14
B. Organizational Requirements	15
1) Development Requirement.....	15
2) Environment Requirement	15
C. External Requirements.....	15
1) Regulatory Requirement.....	15

2) Safety Requirement.....	16
5. System Architecture	17
5.1 System Overview	17
6. System Requirements specification.....	18
6.1. Functional Requirements.....	18
A. White Board	18
B. YouTube Player	19
C. Office Information Showing	19
D. Office Password	20
E. Office Transferring.....	21
F. Office Locking.....	21
G. Ticketing System (Ticketing).....	22
H. Ticketing System (Calling)	22
6.2. Non-functional Requirements	23
A. Product Requirements	23
1) Usability.....	23
2) Security.....	23
3) Reliability	23
4) Performance	23
B. Organizational Requirements.....	23
1) Environmental Requirement.....	23
2) Operational Requirement	24
C. External Requirements	24

1) Safety/Security Requirement.....	24
2) Regulatory Requirement.....	24
6.3. Scenario Examples	24
A. Connection Scenario	24
1) Initial assumption.....	24
2) Normal flow of events	25
3) What can go wrong.....	25
4) System state on completion	25
B. Ticketing Scenario.....	25
1) Initial assumption.....	25
2) Normal flow of events	25
3) What can go wrong.....	26
4) System state on completion	26
C. Meeting Scenario.....	26
1) Initial assumption.....	26
2) Normal flow of events	26
3) What can go wrong.....	27
4) System state on completion	27
D. Transfer Scenario.....	27
1) Initial assumption.....	27
2) Normal flow of events	27
3) What can go wrong.....	28
4) System state on completion	28

7. System Models.....	28
7.1. Context Models.....	29
A. Context Diagram.....	29
B. Process Diagram.....	29
7.2. Interaction Models	30
A. Use case Diagram.....	30
B. Tabular Description of Use case Diagram	30
1) Office Password	31
2) Office Locking	31
3) Calling.....	31
4) Office Information Showing.....	31
5) Calling.....	32
6) White board & Youtube Tool	32
7) Ticketing.....	32
7.3 Behavioral Models.....	33
8. System Evolution	34
8.1 school Authentication.....	34
8.2 Able to choose the environment of the meeting space	34
8.3 Focus on selecting professors.....	34
9. Appendices	
10. Index	

1. Preface

Preface 에서는 본 문서의 예상 독자를 정의하고 문서의 전체적인 구조에 대해 간략히 설명한다. 문서에 변경 사항이 생겼을 때 변경 내용과 이유를 포함한 version history를 각 version 마다 기록한다.

1.1 Readership

Readership에서는 본 문서를 읽을 독자층에 따라 user requirements와 system requirements로 구분한다.

A. User Requirements Readership

이 Readership 의 예상 독자는 system 사용 유저이다. user requirements 는 이용자가 쉽게 이해하기 위해 natural language 로 작성된 문서이다.

B. System Requirements Readership

System requirements는 개발자를 위해 만드는 문서이다. 또한 Client와 Contractor의 계약 문서로 사용될 수 있도록 자세하게 기술한다. System requirements에서 주로 시스템의 functional requirements와 non-functional requirements를 자세하게 설명한다.

1.2 Document Structure

A. Introduction

Introduction에서는 프로젝트의 필요성과 배경을 설명하고 프로젝트의 구조 및 기능을 간단하게 설명한다. 마지막으로 프로젝트의 기대효과를 전망한다.

B. Glossary

Glossary에서는 전문 용어 및 이해하기 어려운 용어를 설명한다.

C. User Requirements Definition

User requirements definition에서는 Functional requirement와 non-functional requirement로 분류하여 시스템이 제공하는 서비스에 대해 설명한다.

D. System Architecture

System architecture에서는 시스템의 전체적인 구조에 대해 기술한다.

E. System Requirements specification

System Requirements 에서는 개발자가 쉽게 이해하기 위해 functional requirements와 non-functional requirements를 기술한다.

F. System Models

System models에서는 시스템을 추상인 모델로 나타내는 system modeling에 대해 기술한다. 각 시스템의 component와 system의 관계, component간의 관계, system을 둘러싼 환경 사이의 관계를 modeling하여 UML diagram으로 나타낸다.

G. System Evolution

System evolution에서는 추측 중 사용자의 요구에 따라 새로운 기능이 추가하거나 기존 기능이 개선하거나 하는 추측을 할 것이다.

H. Appendices

I. Index

독자에게 전체 문헌의 구조와 대략적인 내용을 알려 준다.

J. Reference

본 문서에서 이용한 참고문서를 표시한다.

2. Introduction

이 챕터에서는 Smart space for the professor's office hour 시스템이 왜 필요한지에 대한 설명하고 시스템의 기능을 간략하게 서술하며, 시스템이 사람들에게 끼칠 영향을 예측하여 설명한다.

2.1 Needs

몇 년 동안 코로나19의 영향 때문에 사람들의 생활 방식에 큰 변화가 생겼다. 원래 학교에서 오프라인 수업이나 오프라인 집회는 예사였다. 하지만 코로나19는 전염성이 강하기 때문에 이런 일들이 어려워졌다. 정부도 4단계 격리 시다인 오프라인 집회를 금지한다고 공지했다. 지금이라도 코로나19 발생이 시작된 지 많은 시간이 지났지만, 이러한 문제들은 여전히 우리의 생활을 괴롭히고 있다.

코로나19에 때문에 학생인 우리가 일상적으로 교수님과 커뮤니케이션하는데 큰 지장을 받았다. 예를 들면 예전에는 교수와 학생이 얼굴을 맞대고 소통할 수 있었지만 지금은 코로나19로 인해 다들 걱정할 것이다. 또한 지금은 수업이 온라인 수업이니까 학생들이 대부분 집에서 살고 학교에서 멀다. 사람들은 이런 상황을 해결하기 위해 메일로 소통하거나, 아이캠퍼스를 통해 대화를 나누거나, zoom, webex 등 온라인 미팅 앱을 이용하여 다양한 방법을 생각해냈다. 하지만 이 문제가 해결된 것은 아니다. 이메일의 커뮤니케이션은 즉각적인 것이 아니며 이러한 방법은 많은 시간이 걸릴 것이다. icampus는 이메일과 유사한 문제를 안고 있으며 많은 교수들이 icampus를 사용하지 않다. 그리고 zoom, webex 등 온라인 컨퍼런스 앱은 이러한 문제를 잘 해결하고 있지만, 이 앱들은 교수님과 일대일로 이야기를 나누기가 어렵다.

그래서 우리는 이런 문제를 해결하기 위해 VR chat기반으로 Smart space for the professor's office hour 시스템을 만들 것이다.

2.2 System Overview

Smart space for the professor's office hour은 vr chat를 기반으로 교수와 학생들에게 위안계의 회의 공간을 제공하는 소프트웨어이다. 학생들은 교수의 사무실을 선택해서 그곳으로 전송할 수 있다. 교수가 문을 잠그면 학생들은 교수와 일대일 상담을 하고 사무실에 들어갔을 때 다른 사람들은 사무실이 꽉 찼다는 것을 알 수 있다. 마지막으로 가상 사무실에서 교수와 학생이 YouTube 등 동영상 자료를 공유하고 공유할 수 있다. 그리고 화이트보드를 제공해서 그들이 원하는 것을 적을 수 있다.

2.3 Expected Effects

A. Replace face-to-face interview

단지 코로나 19로 인한 온라인 상황보다도 코로나 19 이후의 오프라인 미팅의 대체로써 사용할 수 있는 부분이 기대된다.

B. Easy to use & More visual

사용자는 간단한 페이지를 통해 원하는 기능을 찾을 수 있다. 기존 webex 등 소프트웨어보다 더 직관적이고 교수님과 학생들이 얼굴을 맞대고 교류하다.

C. Real-Time

학생과 교수는 지체 없이 대화할 수 있다.

3. Glossary

Glossary에서는 본 문서에 사용된 기술적 용어들을 정의하며 일반 사용자도 이해하기 쉽게 설명한다.

Term	Description
VR chat	《VRChat》는 Graham Gaylor 와 Jesse Joudrey 가 개발한 무료 대형 온라인 가상현실 게임이다
User	Smart space for the professor's office hour 서비스를 이용하는 사용자이다.
Sign up	사용자가 서비스를 이용하기 위하여 회원가입을 하는 것을 의미한다
Login	사용자가 서비스에 접근하는 방법이다.
Database	Database 는 데이터 구조에 따라 데이터를 조직, 저장, 관리하는 창고이다.
Mypage	사용자 개인의 공간으로 개인 정보를 수정하고 여러 설정 조건들을 조정할 수 있는 공간이다.

4. User Requirements Definition

아래 내용에서는 해당 시스템이 제공하는 서비스를 functional requirement 와 non-functional requirement, external requirement으로 나눠서 설명한다.

4.1 Functional Requirements

A. White Board

교수님이 설명하거나, 학생들이 질문할 것들을 표현할 수 있는 도구로 화이트 보드를 준비한다. 각 사용자는 화이트보드를 통해서 자신이 원하는 바를 시각적으로 보여줄 수 있다. 해당 내용은 학생이 방을 나가면 내용이 삭제됨으로 정보가 안전하게 보호된다.

B. YouTube Player

교수님이 설명하거나, 학생들이 질문할 것들을 표현할 수 있는 도구로 유튜브 시청 툴을 준비한다. 각 사용자는 플레이어를 통해서 자신이 원하는 바를 영상으로 보여줄 수 있다. 또한 본인이 영상을 조절할 때, 본인뿐만 상대방한테도 그 조절이 적용되어야 한다.

C. Office Information Showing

교수님이 본인의 방을 소개할 수 있도록 방 앞에 본인 정보를 표현할 수 있게 해야한다. 정보로는 교수님 성함, 해당 과목 등을 표현할 수 있다. 사용자가 수정을 원할 때 바로바로 수정이 되어야 하고, 그 수정사항이 방 밖에 사용자도 바로 알 수 있도록 업데이트를 해야 한다.

D. Office Password

교수님은 본인의 방에 password를 설정할 수 있고, password로 방을 열지 닫을지 결정할 수 있다. 교수님은 본인이 password로 방을 열었다면, 다음으로 들어오는 학생들은 password를 입력할 필요는 없다.

E. Office Transferring

사용자들은 본인이 원하는 공간으로 이동한다. 학생들이나 교수님들은 대기 장소에서 본인이 원하는 방으로 이동할 수 있다.

F. Office Locking

학생이 교수님 방에 있다면, 다른 학생들은 방에 출입을 못하게 한다. 그래서 학생들이 방에 들어오면 자동으로 문이 잠기게 된다. 이런 잠금 유무는 문 밖의 버튼에 의해서 구분할 수 있다.

G. Ticketing System (Ticketing)

학생들의 무분별한 방 진입을 막고 학생들에게 들어오는 순서를 정해주기 위해서 학생들은 티켓팅 시스템에서 번호를 뽑고 본인의 순서를 기다리는 시스템을 사용한다. 그래서 본인의 차례가 된다면 문밖에 표시되게 된다.

H. Ticketing System (Calling)

교수님은 밖의 대기 학생들을 위해서 방의 학생의 시간이 끝났다면, 버튼을 눌러서 다음 학생을 호출할 수 있다.

4.2 Non-functional Requirements

A. Product Requirements

1) Usability

이 시스템의 주 이용자는 대학생, 대학교수, 확장성을 생각하면 대학 조교까지 폭넓은 연령층을 가지고 있다. 따라서 이용하기 쉽고, 필요한 내용을 바로 알 수 있는 사용자 인터페이스를 제공해야한다. 이에 따라 부가적인 기능은 최대한 줄이고, 시스템이 제공하는 교수님과의 만남 조건 및 공간, 공간 내에서 화이트보드 및 유튜브 기능 등의 필요한 기능 위주의 인터페이스를 제공한다.

2) Security

이 시스템에서는 교수님과의 학생과의 대화 등의 정보가 외부로 유출되게 되면, 예민한 문제가 될 수 있기 때문에 관리가 철저해야 한다. 또한 교수님이 아닌 외부인이 방을 만들지 못하도록 관리가 되어야한다.

3) Reliability

교수님과 학생과의 시간 낭비를 줄이기 위해서, 서로의 상황을 빠르게 확인할 수 있도록 상황에 대한 처리가 빠르고 정확해야 한다. 또한 학생들의 대기 순서도 오류가 없도록 계속 업데이트를 해줘야 한다.

4) Performance

시스템에서 교수님과 학생의 만남 공간에서의 오류가 있거

나, 교수님 방에 아무도 없는데, 누군가 있다고 처리가 되어있으면, 이용률이 저조해질 수 있다. 그래서 최대한 사용자들 간의 오류가 없어야 하고, 방 내부도 기능 처리를 잘 해서 이용하는데 불편함이 없어야 한다.

B. Organizational Requirements

1) Development Requirement

시스템의 빠른 개발과 배포를 위해서 parallel development 으로 진행한다. 해당 월드의 디자인, 교수님 방 및 학생들 대기공간 구현, 교수님 방과 대기공간과의 상호작용 구현 등으로 나눠서 독립적으로 동시에 개발을 진행한다. 개발에 있어서 모든 과정은 Github를 통해서 관리된다.

2) Environment Requirement

사용자는 VRchat을 기반으로 하는 시스템을 사용하기 위해서 기본적으로 VRchat 환경을 가지고 있어야한다. 또한, 교수님과의 소통을 위해서는 네트워크에 연결되어 있어야한다.

C. External Requirements

1) Regulatory Requirement

위 시스템을 사용하기전에 VRchat 가입에 필요한 개인정보 동의를 구해야 하고, 취급 방침에 대한 알림을 제공해야한다. 또한, 시스템에 대한 설명을 충분히 해야 한다.

2) Safety Requirement

해당 과목 관련되지 않은 인원이 해당 world로 들어오거나, 교수님 외 권한이 없는 다른 사용자가 방을 만들면 시스템에 큰 문제를 일으킬 수 있다. 따라서, 각 사용자의 status를 정확히 확인한 후 권한을 정확히 나누어 줘야한다. 또한 문제를 일으킨 user들은 따로 별도 관리해서 해당 시스템에 제한을 두어야한다.

5. System Architecture

이 챕터에서는 시스템 구성 요소 간의 상호작용을 나타내고, 전체적인 구조를 highly abstract level로 설명한다. 보다 시각적으로 설명하기 위해 다이어그램으로 설명되어 있다.

5.1 System Overview

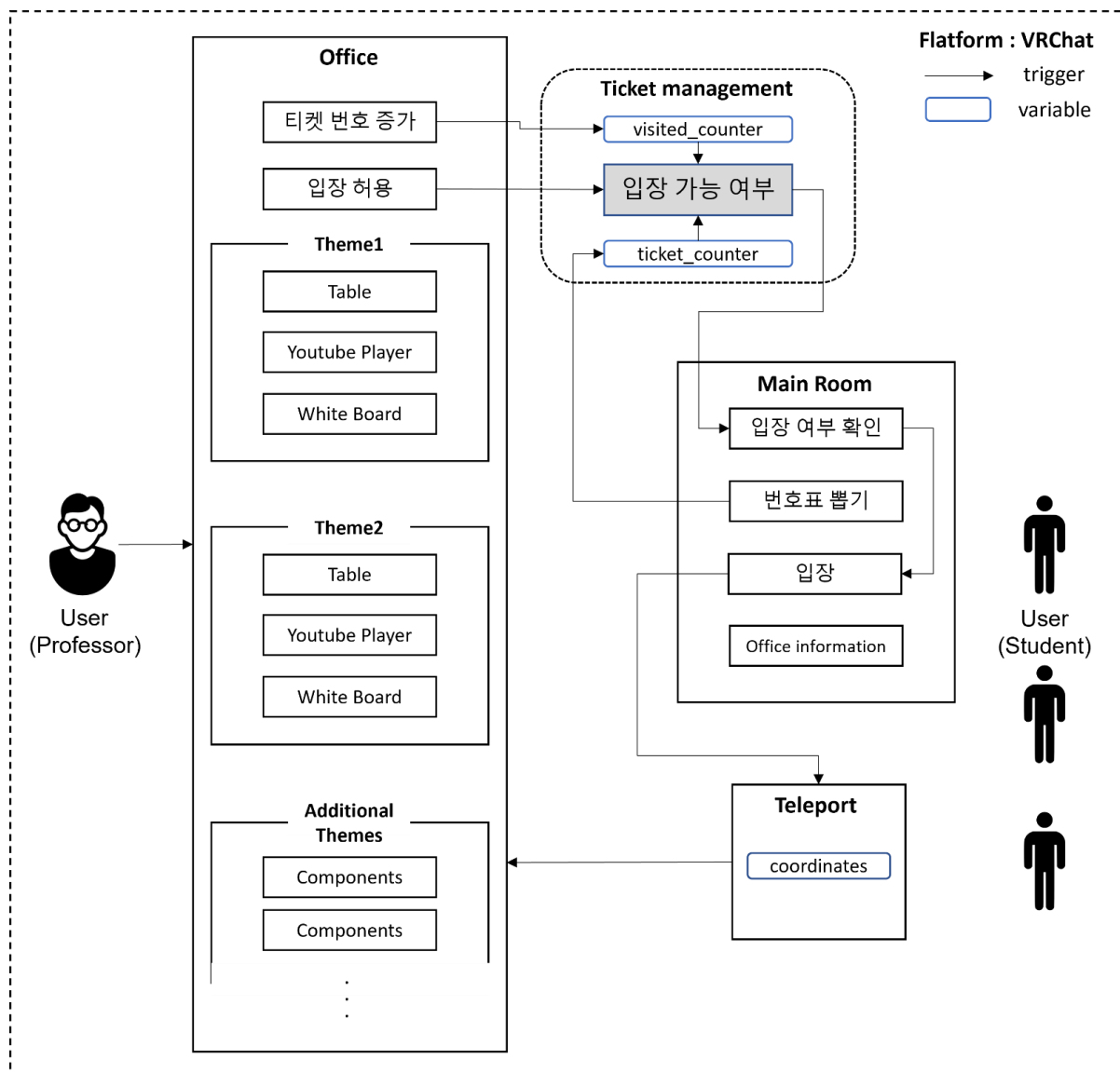


Diagram1: System Overview Diagram

교수님이 각 테마방에 들어가시고, 방을 개방하시면, 학생들이 티켓팅 시스템을 이용해서 출입하는 전반적인 시스템 구조를 나타냈다.

6. System Requirements specification

챕터 4에서 설명했던 User Requirement에 설명되어 있는 요구사항을 보다 상세히 설명하고 Functional Requirement와 Non-functional Requirements로 나눠서 진행한다. 또한 System의 Scenario 예시도 설명한다.

6.1 Functional Requirements

A. White Board

Name	White Board
Description	교수님 혹은 학생들이 화이트보드를 통해 글을 쓰거나 그림을 그려서 자신의 뜻을 시각적으로 보여줄 수 있다.
Normal Course	<ol style="list-style-type: none"> 1. 놓아져 있는 펜을 상호작용하여 집는다. 2. 마우스를 클릭하는 동안 원하는 글씨나 그림이 그려지도록 한다. 3. 특정 키를 통해 펜을 손에서 놓을 수 있다. 4. 놓아져 있는 지우개 또한 상호작용하여 집는다. 5. 마우스를 클릭하는 동안 특정 글씨나 그림을 지운다 6. 특정 키를 통해 지우개를 손에서 놓을 수 있다.

Precondition	N/A
Post Condition	상호작용한 사람뿐만 아니라 다른 사람도 글씨나 그림을 볼 수 있어야 한다.
Assumptions	N/A

B. YouTube Player

Name	Youtube Player
Description	교수님 혹은 학생들이 영상 자료를 통해 의논할 수 있도록 한다. Youtube 뿐만 아니라 Google drive, Vimeo 등의 url 도 사용할 수 있다.
Normal Course	<ol style="list-style-type: none"> 1. 비디오 플레이어를 사용하고자 하는 자가 텍스트 입력란에 영상의 url 을 입력한다. 2. 영상이 재생되는 것을 본다. 3. 컨트롤러를 통해 일시정지 및 재생을 조작하거나 볼륨을 조절할 수 있다.
Precondition	비디오가 유튜브나 구글 드라이브 등에 업로드 되어있어야 하고, 링크를 알고 있어야 한다.
Post Condition	상호작용한 사람 뿐만 아니라 다른 사람에게도 비디오의 변경 사항(재생, 일시정지, 볼륨)이 적용되어야 한다.
Assumptions	N/A

C. Office Information Showing

Name	Office Information Showing
------	----------------------------

Description	교수님 방문 앞에 방의 정보(교수님 성함, Office hour 등)를 써 두고 사용자가 볼 수 있게 한다.
Normal Course	<ol style="list-style-type: none"> 1. 교수님은 방 안에서 방의 정보를 입력한다. 2. 입력과 동시에 방 밖에서 보이는 문 앞에 방의 정보가 나타난다. 3. 학생들이 방의 정보를 읽고 출입할 수 있다.
Precondition	방 안에서 방의 정보를 수정한다.
Post Condition	방 밖에서 모든 유저가 수정된 방의 정보를 읽을 수 있어야 한다.
Assumptions	교수님만 방의 정보를 수정할 수 있다.

D. Office Password

Name	Office Password
Description	교수님은 문 앞의 비밀번호를 입력하여 자신의 Office 에 자유롭게 출입이 가능하다.
Normal Course	<ol style="list-style-type: none"> 1. 문 앞에 0 부터 9 까지의 10 개의 숫자를 가진 키패드가 존재한다. 2. 교수님은 각 숫자가 쓰여 있는 버튼을 3 번 반복하여 눌러서 길이가 3 인 비밀번호 입력한다. 3. 비밀번호가 맞을 경우 Office 안으로 이동하며, 틀릴 경우 틀렸다는 안내문을 띄워준다.
Precondition	방마다 세 자리의 비밀번호가 미리 지정되어 있다.
Post Condition	비밀번호 정답 여부에 따라 특정 트리거가 발동한다.
Assumptions	교수님은 비밀번호를 이미 알고 있다고 가정한다.

E. Office Transferring

Name	Office Transferring
Description	사용자가 방 문에 접촉하면 월드 내 별도로 마련된 방으로 사용자를 이동시킨다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자는 방 문에 다가선다. 2. 해당 방 문과 연결된 방으로 자동으로 이동하게 된다. 3. 방에 잠금이 자동으로 걸린다.
Precondition	방에 잠금이 걸려있지 않다.
Post Condition	N/A
Assumptions	N/A

F. Office Locking

Name	Office Locking
Description	미팅이 예정된 학생 외의 사람들의 접근을 막기 위해, 문 밖에서의 입장은 기본적으로 막혀 있다. 교수님에 의해 방의 잠금이 풀릴 때만 방에 들어갈 수 있다. 잠금 여부는 문 앞에 표시된다. 방 안에서 밖으로 나갈 땐 자유롭게 나갈 수 있다.
Normal Course	<ol style="list-style-type: none"> 1. 교수님은 방 안에 위치한 버튼을 누른다. 2. 해당 버튼에 의해 방의 잠금이 해제되고 문 위의 전구의 불빛이 변한다 (빨간색 -> 초록색). 3. 학생은 문에 접근해 방으로 이동이 가능해진다.
Precondition	교수님은 방 안에 계신다.

	잠금 상태의 방 문의 전구는 빨간색이다.
Post Condition	방의 잠금이 풀린다.
Assumptions	N/A

G. Ticketing System (Ticketing)

Name	Ticketing System (Ticketing)
Description	현재 Office 에서 회의가 진행중이라면 학생은 일정 시간 뒤에 다시 방문해야 한다. 이 때 번호표를 뽑아서 가면 차례가 됐을 때 알림이 오게 된다.
Normal Course	<ol style="list-style-type: none"> 1. 학생은 원하는 교수님의 방 문 앞에 도달한다. 2. 문 옆에 있는 버튼을 눌러 번호표를 뽑고 기다린다. 3. 차례가 되면 눈 앞에 안내문이 띄워지면서 알림이 온다. 4. 교수님의 Office 에 출입할 수 있다.
Precondition	교수님은 방 안에 계시고 다른 사람과 회의 중이다.
Post Condition	차례가 되면 알림이 온다.
Assumptions	학생이 교수님과 회의를 하기를 원한다.

H. Ticketing System (Calling)

Name	Ticketing System (Ticketing)
Description	Office 에서 회의가 끝난 후 교수님은 대기중인 다음 학생을 부를 수 있다.
Normal Course	<ol style="list-style-type: none"> 1. 교수님은 방 안에 위치한 버튼을 누른다.
Precondition	진행중인 회의가 끝난 상태다.

Post Condition	다음 학생에게 알림이 간다.
Assumptions	대기중인 학생이 있다.

6.2 Non-functional Requirements

A. Product Requirements

1) Usability

본 시스템의 사용 대상은 교강사와 학생들이다. Office hour 시간에 교수자와의 미팅을 용이하게 만들기 위해서 본 시스템을 구현한 것이기 때문에, 복잡한 디자인 보다는 간결한 인터페이스를 지향하여 가상 공간에서의 만남 기회를 보다 쉽고 빠르게 제공한다.

2) Security

별도로 구현된 방이 메인 홀과 가까워 소리가 새어나가는 것을 방지하기 위해, 메인 홀과의 거리를 충분히 멀게 한다.

3) Reliability

미팅이 진행중인 방에 타인이 침입하여 미팅이 방해받는 일이 발생하지 않도록 해야 하기 때문에, 메인 홀에 대기중인 학생들이 방 입장에 대한 사항을 안내 받도록 한다.

4) Performance

다수의 사용자가 접속해도 월드 운영이 원활해야 한다. 따라서 메인 홀을 대학별로 나누어 둠으로써 인원을 분산시킬 수 있다.

B. Organizational Requirements

1) Environmental Requirements

시스템의 대부분 요소는 합법적으로 만들어진 Prefab 을 사용한다. 이미 정교하게 구현된 모델들로서 개발 비용을 상당히 줄일 수 있다.

2) Operational Requirements

시스템은 각각 VR Chat 내 아이디로 사용자를 구분한다. 한 월드 내에 여러 명의 학생과 교수의 동시 접속을 허용하며, 구현된 물체들과 상호 작용이 가능하다. 장소나 시간대에 상관없이 VR Chat 에 접속한다면 시스템을 자유롭게 이용 가능하다.

C. External Requirement

1) Safety/Security Requirements

학교 관계자가 아닌 사람이 시스템에 접속할 경우 중요한 정보 유출이 발생할 수 있다. 따라서 VR Chat 내부에서 해당 world 는 private 으로 만든 후, 학교 관계자로 확인되고 초대받은 사람만 이용 가능하도록 한다.

2) Regulatory Requirements

시스템을 이용하기 앞서 VR Chat 에 가입할 때 개인정보 수집에 대한 동의를 미리 구하고 community 가이드라인을 제시하여 이용에 불편함이 없도록 한다.

6.3 Scenario examples

A. Connection Scenario

1. Initial assumption

교수는 학생들에게 room code 와 office hour 를 미리 공지한다. 해당 시간에 교수는 이미 NEMO(Neo Metaverse Office)의 본인 office 에 접속해 있는 상태이며 학생은 미팅을 위해 시스템에 접속하고자 한다.

2. Normal flow of events

교수는 본인 office 에 비밀번호를 입력하여 들어간다. 학생은 교수가 공지한 office hour 에 교수의 메타버스 오피스 공간에 입장한다.

3. What can go wrong

교수가 본인 office 에 입장할 때 비밀번호를 잘못 입력하면, 비밀번호가 틀렸다는 메시지가 뜨도록 한다. 교수가 학생과 개별 미팅 시간을 정한 것이 아니라면, 한 번에 여러 학생들이 교수의 office 공간에 접속하여 혼란이 빚어질 수 있다. 따라서 학생들이 처음에 입장을 할 때, 교수의 office 공간에 바로 들어오는 것이 아니고 main hall 에 접속을 하도록 한다. 이 main hall 은 학생들의 대기 공간과 같은 역할을 한다.

4. System state on completion

교수는 office hour 에 본인의 office 에 접속해 있는 상태이며, 학생들은 main hall 로 접속하여 대기하고 있는 상황이다.

B. Ticketing Scenario

1. Initial assumption

여러 학생이 main hall 에 접속해 있으며, 교수의 office 에 개별로 이동하여 미팅을 하고자 한다.

2. Normal flow of events

한 학생이 이미 office 에서 교수와 미팅을 하고 있다면, 다른 학생들은 티켓 버튼을 눌러 번호표를 뽑고 대기할 수 있다. 학생과의 미팅이 끝나면 교수는 특정 버튼을 눌러 입장 번호를 1 씩 증가시킨다. 해당 입장 번호에 해당하는 학생에게는 "Your turn has come!" 안내문이 뜬다. 이 안내문을 확인한 학생은 교수의 office 로 이동한다.

3. What can go wrong

대기 번호가 있음을 확인하지 못해 본인의 차례가 아닐 때에 교수님의 office 에 입장하려는 학생이 있을 수 있다. 또한 교수가 office hour 에 잠시 자리를 비우고자 하는 경우도 있다. 이러한 경우, 다른 학생들이 교수의 office 로 들어가는 것을 막기 위해 office 의 문을 잠그는(locking) 기능을 활용한다. 문이 잠기면 문 밖에서 안으로의 입장은 불가능하며 문의 잠금 여부는 문 위 전구의 색으로 학생들에게 알려줄 수 있다. 문이 잠긴 경우에는 빨간색 불빛, 문이 열린 경우에는 초록색 불빛이 뜨도록 한다.

4. System state on completion

학생들은 main hall 에서 대기하고 있다가 본인의 차례가 되어 알람이 뜨면 교수의 office 로 개별 입장한다.

C. Meeting Scenario

1. Initial assumption

교수와 학생이 교수의 office 에서 office hour 를 갖고자 한다.

2. Normal flow of events

본인의 차례가 된 학생은 교수의 office 공간으로 이동하여 교수와 자유롭게 미팅을 갖는다. 이 때, office 안의 여러 소품들을

활용해 메타버스 공간에서 미팅을 하는 장점을 극대화할 수 있다. 예를 들어, office 안의 화이트보드는 중요한 내용을 설명하거나 메모를 하는 용도로 교수와 학생 모두가 활용할 수 있다. 또한 시각 자료가 필요할 때에는 YouTube player 를 통해 영상을 공유하여 함께 시청할 수 있다.

3. What can go wrong

미팅을 원하는 학생들이 많아 주어진 office hour 내에 모든 학생들과 만남을 가질 수 없는 경우가 있을 수 있다. 따라서 office hour 시간 외에 추가적인 미팅 또는 질문을 하고 싶어할 학생들을 위해 교수의 office 앞에 교수의 이메일, 연락처, 그리고 정해진 office hour 시간 등의 정보를 표시한다. 이 정보는 해당 교수만이 작성 및 수정 가능하다.

4. System state on completion

학생은 한 명씩 교수의 office 로 이동하여 교수와의 미팅을 갖는다. 미팅이 끝난 학생은 자유롭게 교수의 office 를 나갈 수 있다. 다른 학생들은 본인의 차례를 main hall 에서 기다린다.

D. Transfer Scenario

1. Initial assumption

교수와 학생이 교수의 office 에서 미팅을 갖던 중 다른 종류의 방으로 함께 이동하고자 한다.

2. Normal flow of events

교수와 학생은 미팅의 특성에 따라 여러 다른 테마의 방들로 함께 이동할 수 있다. 예를 들어, 교수의 office 는 간단한 상담 및

질의응답에 적합한 공간으로 설계되어 있다. 그러나 만약 실습 및 시연이 필요한 경우라면, 더 넓은 공간에 여러 소품들이 있는 방으로 문을 통해 이동할 수 있다.

3. What can go wrong

문을 클릭해야 이동할 수 있다는 추가적인 설정 등을 하면, 어떻게 문을 넘어가야 할지 모르는 학생들이 생겨날 수 있다. 따라서 특정한 행동을 해야 이동 가능한 방식이 아니라, 문 가까이 다가가서면 자동적으로 그 방에 입장할 수 있도록 구현한다.

4. System state on completion

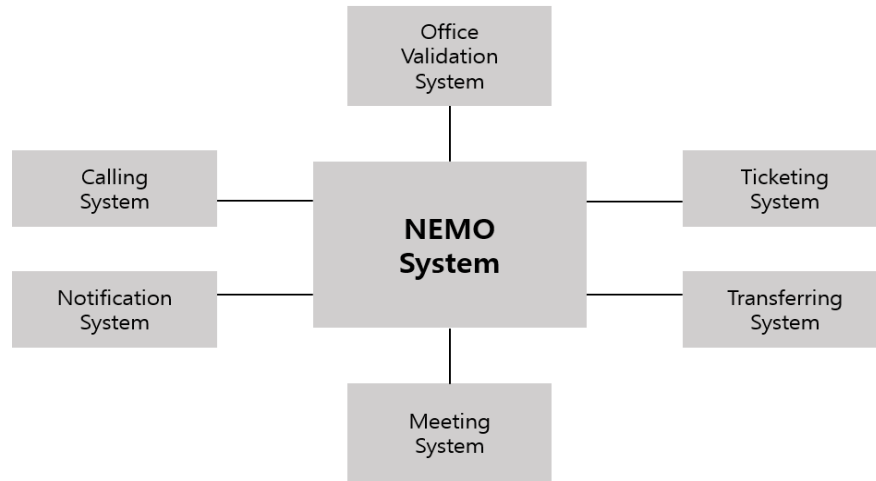
교수와 학생은 미팅의 목적에 최적화된 공간에서 미팅을 가질 수 있다.

7. System Models

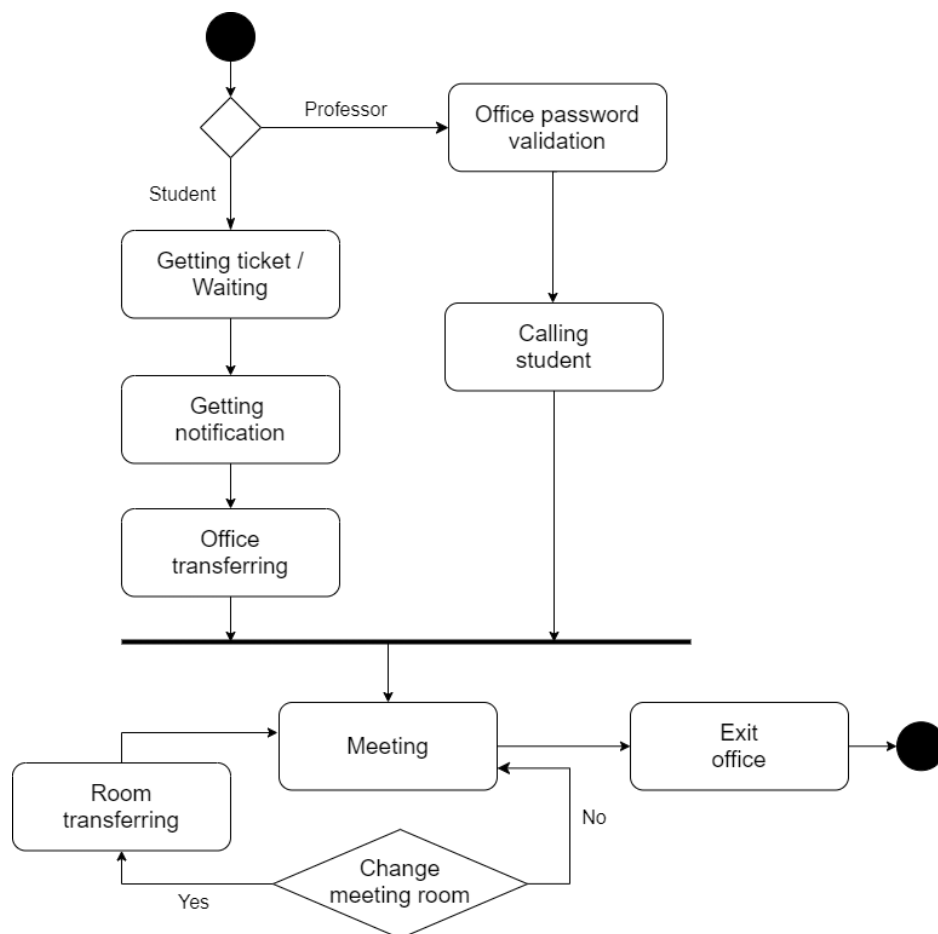
이번 챕터에서는 각 시스템 컴포넌트와 전체 시스템 그리고 시스템을 둘러싼 환경 사이의 관계를 여러 가지 Diagram을 통해 기술한다

7.1 Context Models

A. Context Diagram

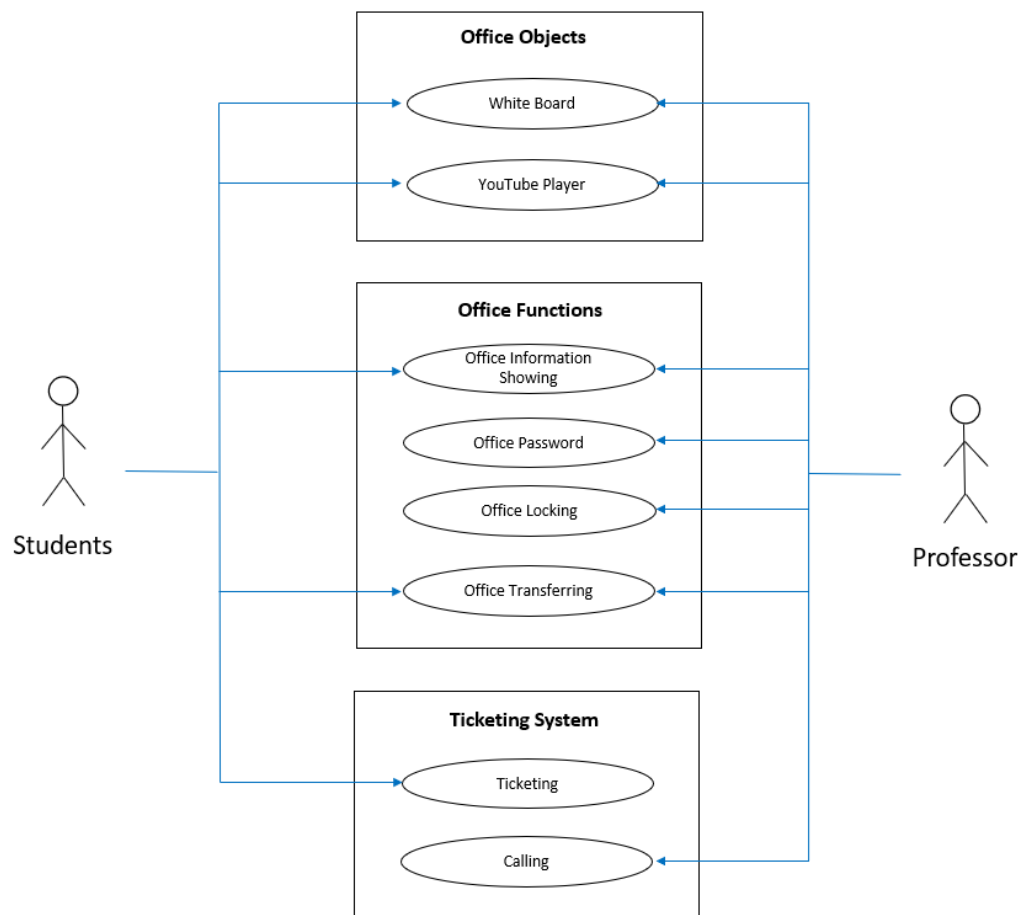


B. Process Diagram



7.2 Interaction Models

A. Use case Diagram



B. Tabular Description of Use case Diagram

1) Office Password

Use Case	Office Password
Actor	Professor
Description	사용자가 방 비밀번호를 알고 있는 교수님인지 확인한다.
Trigger	사용자가 비밀번호를 입력한다.
Success Response	사용자가 본인의 공간으로 갈 수 있는 권한이 생긴다. 또한, 방에 다른 사용자들이 출입할 수 있는 권한도 부여된

	다.
Failure Response	비밀번호가 일치하지 않는 경우, 오류 메시지를 띄운다.

2) Office Locking

Use Case	Office Locking
Actor	Professor
Description	교수님이 방을 나올 때 방에 출입 제한을 시킨다.
Trigger	교수님이 방을 나간다.
Success Response	교수님이 방을 나가면, 다시 출입 제한이 되었다는 알림이 뜨고, 교수님을 제외한 다른 사용자는 방에 출입하려고 할 시 제한을 받는다.
Failure Response	-

3) Calling

Use Case	Calling
Actor	Professor
Description	교수님이 방 미팅이 끝났음을 알리고 다음 학생이 들어올 수 있도록 알림을 보낸다.
Trigger	교수님이 방 내의 버튼을 누른다.
Success Response	버튼을 누를 경우, 방 밖의 출입 가능 알림과 번호 알림에 신호를 전달한다.
Failure Response	-

4) Office Information Showing

Use Case	Office Information Showing
Actor	Professor, Student
Description	교수님 방문 앞에 교수님 성함, Office hour와 같은 방의 정보를 써 두고 사용자가 볼 수 있게 한다.

Trigger	방 내의 교수님이 정보를 바꾼다.
Success Response	교수님이 바꾼 정보를 그대로 방 밖의 알림판에 전달하여, 방 밖에 사용자들이도 내용을 확인 할 수 있게한다.
Failure Response	-

5) Office Transferring

Use Case	Office Transferring
Actor	Professor, Student
Description	사용자가 정해진 장소로 이동한다.
Trigger	사용자가 포탈에 진입한다.
Success Response	각 포탈에 권한을 확인한다. 권한에서 문제가 없을시, 정해진 좌표로 사용자를 이동시킨다.
Failure Response	-

6) White board & YouTube Tool

Use Case	White board & YouTube Tool
Actor	Professor, Student
Description	서로의 상호작용을 효율적으로 하기 위해서 화이트 보드 및 동영상 플레이어를 사용할 수 있다.
Trigger	화이트 보드 내의 사용자가 표시하고 싶은 정보를 나타내거나, 동영상 플레이어로 사용자가 정보를 전달한다.
Success Response	화이트 보드내의 내용 및 플레이어 내의 영상이 사용자에게 정확하게 전달된다.
Failure Response	동영상이 정확한 경로가 아닐 경우, 없는 영상이라고 알림이 뜬다.

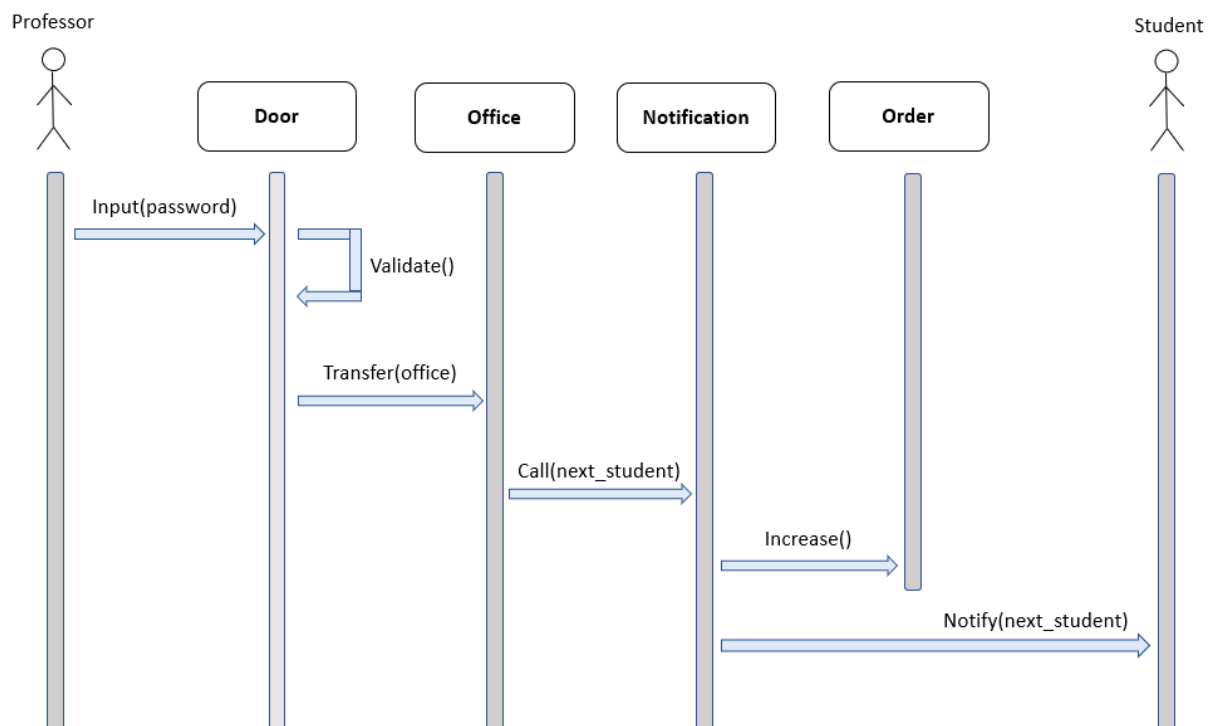
7) Ticketing

Use Case	Ticketing
----------	-----------

Actor	Student
Description	앞의 학생이 방에 있을 경우, 중복 출입을 방지하기 위해서 번호 받고 본인의 순서를 대기한다.
Trigger	번호표에서 버튼을 누르면 본인의 번호가 나온다.
Success Response	저장되어 있는 번호가 사용자에게 정확히 출력되고, 그 변수는 다음 사용자를 대기하면서 값에 1이 추가된다.
Failure Response	학생이 방에 출입할 수 있으면서 번호표의 버튼을 누른 경우, 방에 출입할 수 있다는 알림이 뜨면서, 번호가 출력되지 않는다.

7.3 Behavioral Models

시스템의 전반적인 흐름에 대하여 Sequence Diagram 을 통해 System 의 Behavioral Model 을 도식화 하였다.



8. System Evolution

이 챕터에서는 시스템이 배포된 후 진행될 수 있는 시스템의 진화를 예측하고, 시스템의 발전 방향을 제시한다.

8.1 school Authentication

교수와 학생에게 회의공간을 제공하는 시스템이기 때문에 회원가입을 할 때 학교인증을 하면 구분이 잘 된다.

8.2 Able to choose the environment of the meeting space

장시간 반복되는 환경 속에서 회의가 진행되면 피로감과 집중력이 저하될 수 있기 때문에 후속으로 여러 회의환경이 제공되거나 사용자 스스로 회의환경을 정의할 수 있다.

8.3 Focus on selecting professors

많은 교수님들을 위한 서비스이기 때문에 학생들이 대화를 많이 하는 교수를 찾을 때 불편할 수 있으며, 따라서 자신이 자주 만나는 교수를 빨리 찾을 수 있도록 교수를 관심하는 기능을 제공한다.

9. Appendices

이번 챕터에서는 보다 시스템 개발에 관한 세부 정보들을 기술한다.

9.1 Hardware requirement

본 시스템은 VRchat 을 기반으로 Unity 개발하여서 모든 맥 OS, 윈도우 등 거의 모든 운영체제를 지원한다. 윈도우는 최소 윈도우 7 을 지원하며, 권장 사양으로는 윈도우 10 이다. 또한 VRchat 을 통해서 소통하기 때문에, 네트워크와 연결된 상황에서만 사용할 수 있다.

10. Index

10.1 Table

Table 1: Glossary	11
Table 2: White Board.....	18
Table 3: YouTube Player	19
Table 4: Office Information Showing	19
Table 5: Office Password.....	20
Table 6: Office Transferring.....	21
Table 7: Office Locking.....	21
Table 8: Ticketing System (Ticketing)	22
Table 9: Ticketing System (Calling).....	22

10.2 Diagrams

Diagram 1: System Overview Diagram.....	17
Diagram 2: Context Diagram	29
Diagram 3: Process Diagram.....	29
Diagram 4: Use case Diagram.....	30
Diagram 5: Behavioral Model.....	33