목차

[1. Preface 3](#_Toc512810769)

[1.1. Objective 3](#_Toc512810770)

[1.2. Readership 3](#_Toc512810771)

[1.2.1. User Requirement Readership 4](#_Toc512810772)

[1.2.2. System Requirement Readership 4](#_Toc512810773)

[1.3. Document Structure 4](#_Toc512810774)

[1.3.1. Preface 4](#_Toc512810775)

[1.3.2. Introduction 4](#_Toc512810776)

[1.3.3. Glossary 4](#_Toc512810777)

[1.3.4. User Requirement Definition 4](#_Toc512810778)

[1.3.5. System Architecture 4](#_Toc512810779)

[1.3.6. System Requirement Specification 4](#_Toc512810780)

[1.3.7. System Models 5](#_Toc512810781)

[1.3.8. System Evolution 5](#_Toc512810782)

[1.3.9. Appendices 5](#_Toc512810783)

[1.3.10. Index 5](#_Toc512810784)

[1.4. Version of the Document 5](#_Toc512810785)

[1.4.1. Version Format 5](#_Toc512810786)

[1.4.2. Version Management Policy 5](#_Toc512810787)

[1.4.3. Version Update History 5](#_Toc512810788)

[2. Introduction 5](#_Toc512810789)

[2.1. Objective 5](#_Toc512810790)

[2.2. Needs 5](#_Toc512810791)

[2.3. Our Game 6](#_Toc512810792)

[2.4. Expected Effect of the Service 8](#_Toc512810793)

[3. Glossary 8](#_Toc512810794)

[3.1. Objective 8](#_Toc512810795)

[3.2. Term Definitions, Acronyms and Abbreviations 8](#_Toc512810796)

[3.2.1. Term Definition 8](#_Toc512810797)

[A. User & Service 관련 용어 8](#_Toc512810798)

[3.2.2. Acronym and Abbreviation 8](#_Toc512810799)

[4. User Requirement Definition 9](#_Toc512810800)

[4.1. Objective 9](#_Toc512810801)

[4.2. Functional Requirements 9](#_Toc512810802)

[4.2.1. 9](#_Toc512810803)

[4.3. Non-Functional Requirements 9](#_Toc512810804)

[4.3.1. 9](#_Toc512810805)

[5. System Architecture 9](#_Toc512810806)

[5.1. Objective 9](#_Toc512810807)

[5.2. Game Manager System 9](#_Toc512810808)

[5.3. User Profile Manager System 9](#_Toc512810809)

[5.4. 9](#_Toc512810810)

[5.5. 9](#_Toc512810811)

[6. System Requirement Specification 9](#_Toc512810812)

[6.1. Objective 9](#_Toc512810813)

[6.2. Functional Requirements 9](#_Toc512810814)

[6.2.1. 9](#_Toc512810815)

[6.3. Non-Functional Requirements 9](#_Toc512810816)

[6.3.1. 9](#_Toc512810817)

[6.4. Scenario 9](#_Toc512810818)

[6.4.1. Join Scenario 9](#_Toc512810819)

[6.4.2. Question & Answer Scenario 9](#_Toc512810820)

[6.4.3. Winning Scenario 10](#_Toc512810821)

[6.4.4. Managing User profile Scenario 10](#_Toc512810822)

[7. System Models 10](#_Toc512810823)

[7.1. Objective 10](#_Toc512810824)

[7.2. Context Models 10](#_Toc512810825)

[7.3. Interaction Models 10](#_Toc512810826)

[7.4. Structural Models 10](#_Toc512810827)

[7.5. Behavioral Models 10](#_Toc512810828)

[8. System Evolutions 10](#_Toc512810829)

[8.1. Objective 10](#_Toc512810830)

[8.2. Limitations and Assumption 10](#_Toc512810831)

[8.3. Evolutions of User Requirement 10](#_Toc512810832)

[8.3.1. Voice Recognition-Based System 10](#_Toc512810833)

[8.3.2. Rule Modification Sandbox 10](#_Toc512810834)

[8.3.3. New game mods 10](#_Toc512810835)

[9. Appendices 10](#_Toc512810836)

[9.1. Objective 10](#_Toc512810837)

[9.2. Database Requirements 10](#_Toc512810838)

[9.3. User-System Requirements 11](#_Toc512810839)

[9.4. Development Process 11](#_Toc512810840)

[10. Index 11](#_Toc512810841)

[10.1. Table Index 11](#_Toc512810842)

[10.2. Figure Index 11](#_Toc512810843)

[10.3. Diagram Index 11](#_Toc512810844)

[11. Reference 11](#_Toc512810845)

1. Preface
   1. Objective

Preface에서는 본 문서의 예상되는 독자들과 문서의 전반적인 구조, 그리고 각 부분의 역할에 대하여 제시한다. 또한 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경사항들과 그에 대한 근거들을 서술한다.

* 1. Readership

본 요구사항 명세서는 독자에 따라 크게 User Requirement와 System Requirement의 두 부분으로 구성되어 있다.

* + 1. User Requirement Readership

User Requirement는 사용자의 관점에서 요구사항을 간략히 명세한 것이다. 본 프로젝트에서는 Game Player의 관점에서 서술한다.

* + 1. System Requirement Readership

System Requirement는 본 프로젝트에서 개발해야 하는 시스템의 특정 기능이 어떻게 구현되어야 하는지를 자세하게 기술한 것이다. 본 프로젝트에서는 Software Developer와 Server 및 Database Manager 관점에서 서술한다.

* 1. Document Structure

이 문서는 총 10개의 부분으로 구성되어 있다. Preface, Introduction, Glossary, User Requirements Definition, System Architecture, System Requirements Specification, System Models, System Evolution, Appendices, Index로 구성된다. 각 장의 역할과 전반적인 서술은 다음과 같다.

* + 1. Preface

Preface에서는 본 문서의 예상되는 독자들과 문서의 전반적인 구조, 그리고 각 부분의 역할에 대하여 서술한다. 또한 버전 관리 정책, 버전 변경 기록, 그리고 문서들의 변경사항과 그에 대한 근거들을 서술한다.

* + 1. Introduction

Introduction에서는 시스템의 필요성과 해당 시스템이 어떠한 Needs를 반영하고 있는지 서술한다. 또한 시스템의 기능이 다른 시스템과 어떻게 상호작용하는지 간략하게 서술한다.

* + 1. Glossary

Glossary에서는 본 문서에 등장하는 기술적 용어들과 축약어들을 정의한다.

* + 1. User Requirement Definition

User Requirement Definition에서는 사용자의 요구사항을 User Level에서 서술한다. 크게 Functional Requirement와 Non-Functional Requirement로 나누어 서술한다.

* + 1. System Architecture

System Architecture에서는 각 Component들의 hierarchy에 대하여 Diagram등을 이용해 서술한다.

* + 1. System Requirement Specification

System Requirement Specification에서는 User Level에서 정의되었던 Functional/Non-Functional Requirement들을 Developer Level로 세분화하여 서술한다.

* + 1. System Models

System Models에서는 System Component, System, 그리고 System Environment 사이의 관계를 Figure 및 Diagram을 통해 서술한다.

* + 1. System Evolution

System Evolution에서는 시스템이 세우고 있는 주요한 가정들과 사용자의 Needs 변화, 혹은 추후 Update될 사항들에 대해서 서술한다.

* + 1. Appendices

Appendices에서는 본 프로젝트와 관련한 참고사항들을 서술한다.

* + 1. Index

Index에서는 본 문서에 사용된 Table, Diagram, Fiqure들의 Index를 서술한다.

* 1. Version of the Document
     1. Version Format

Version 번호는 major.minor[.maintenance]로 구성되며, 문서의 버전은 0.1부터 시작한다.

* + 1. Version Management Policy

본 문서를 수정할 때마다 버전을 업데이트 한다. 이미 완성한 파트를 변경할 때는 Minor number를 변경하며, 새로운 부분을 추가하거나 괄목한 만한 변화가 있을 경우 Major number를 변경한다.

* + 1. Version Update History

|  |  |  |
| --- | --- | --- |
| Version | Committer | Explanation |
| 0.1 | 김동현 | 문서 목차 작성 및 Preface 서술. |
| 1.0 | 이용철 | Introduction 부분 서술 및 세부항목으로 Our game 파트 추가. |
| 2.0 | 김장훈 | User Requirement 일부 서술 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Introduction
   1. Objective

Introduction에서는 시스템의 필요성과 해당 시스템이 어떠한 Needs를 반영하고 있는지 서술한다. 또한 시스템의 기능이 다른 시스템과 어떻게 상호작용하는지 서술한다.

* 1. Needs

사람들이 모여 다같이 즐길 수 있는 놀이로 보드 게임들을 많이 한다. 또한, 최근 몇 년 사이에 보드 보드게임 카페의 수가 증가하고 있고, 카페를 가면 여러 사람들이 모여 다양한 게임을 하는 모습, 심지어 손님이 너무 많아 자리가 없어 다른 카페를 찾아 나가는 사람들을 볼 수 있다. 보드 게임 중에서 마피아류 보드 게임을 하는 것을 심심치 않게 볼 수 있는데, 예를 들어 '뱅!', '스파이폴', '한밤의 늑대인간' 등을 하는 것을 볼 수 있다. 앞의 설명에서 보듯이, 우리는 보드 게임 중에서 마피아류 게임에 집중을 하고 있는데, 그 이유는 이 게임은 특별한 도구 없이 각자의 역할에 충실하여 속임을 통해 몇 명이든 다 같이 참여할 수 있는 게임으로 보드게임 카페뿐만 아니라, MT, 혹은 온라인에서 모여서 하기도 한다.

우리가 이 게임을 하면서 여러 문제를 겪을 수 있는데, 그 중 역할만 주어질 뿐 정보의 획득은 추리를 통해 서로의 심리를 읽어 얻는 것으로 정보의 불균형으로 인해 자칫하면 일방적으로 끝날 수 있다. 또한, 따로 사회자가 없어 게임의 진행을 참여자들이 진행을 간섭함으로써, 방향을 유리하게 만들어 가는 경우도 생긴다.

우리는 이에 따른 대안으로 게임의 사회자로 chatBot을 참여시키는 방법이다. chatBot이 사회자 역할을 함으로써, 게임을 흥미진진하게 진행이 될 수 있도록 상황에 맞추어 한쪽에 유리한 정보의 쏠림 방지하기 위해 약간의 정보의 제공을 해주는 기능, 참여자의 질문에 대해 다른 참여자의 질문에 따른 참 / 거짓의 판별, 투표 등의 행위를 처리 등의 기능을 제공할 계획이다.

* 1. Our Game

여기서 우리는 기존의 마피아 게임을 수정하여 제안한다. 우리의 게임은 다음의 규칙이 있다.

* 구성 : 사회자(chatBot), TEAM-A(king & members), TEAM-B(king & members)
* 각 참여자에 대해 갖는 제약조건들 :
  + .....
  + .....
  + ....
* 규칙 :
  + 각 참여자는 역할에 충실하면서 자유롭게 추리를 한다. 참여자는 자신이 속한 팀의 구성원이 누구인지 알 수 없다. 그 중에 사회자에게 질의를 할 수 있는 기회를 얻은 참여자가 모두에게 혹은 지목 & 선언을 통해 질의를 할 수 있다.
  + 사회자는 받은 질의를 모두에 공개하고 이에 대상자(들)은 답변을 해야 한다.
  + 사회자는 참여자의 제약조건과 규칙에 비교 후 결과를 공개한다.
  + 일정 시간 후에 사회자가 혹은 참여자 중 누군가 '투표!' 등을 발언 시 투표 시스템을 시작한다.
  + 팀의 왕이 죽으면 게임 종료 & 살아남은 왕의 팀이 승리.

우리가 제안한 게임 시스템은 Discord 기반에서 실행 될 것이며, 다음의 기능은 3가지의 기능 제공할 계획이다.

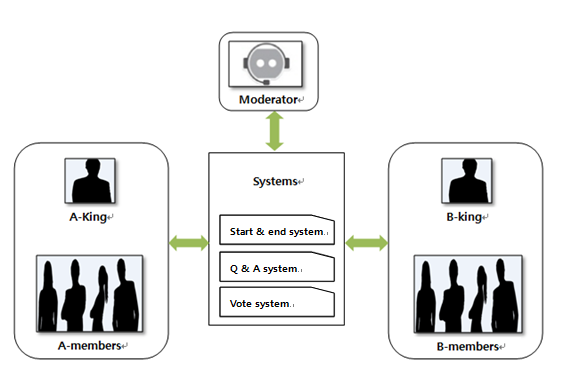
* 참여자의 역할 및 각 참여자에 대한 제약조건 랜덤 지정 기능 (Start System)
* 참여자의 질문 해석 및 다른 참여자들의 답변에 대한 참 & 거짓 판별 기능  
  (Q & A System)
* 투표 시스템 기능. (Vote System)

Figure1 Our game architecture

Start & End system : 게임의 시작과 엔드 조건을 충족하는지 관리하는 시스템이다. Start system은 참여자들에게 역할 & 제약 조건 & 시작은 위한 기본 초기화를 제공하기 위한 시스템이다. End System은 게임의 끝을 체크하기 위한 것으로, 왕이 죽었는지 체크하기 위한 것이다.

Q & A system : 참여자의 질의 및 답변을 관리하기 위한 시스템. 참여자의 질의를 해석 및 관리하고, 이를 다른 참여자들에게 통보. 이에 답변들을 관리하고, 해석을 통해 결과를 통보한다.

Vote system : 참여자들의 투표를 행하고 투표 결과를 통보하고, 많은 표를 받은 참여자를 게임에서 제외시키는 시스템이다.

* 1. Expected Effect of the Service
* Player
  + 정보의 불균형으로 인한 게임의 편향성을 최소화함으로써, 진행함에 있어서 흥미진진하게 게임을 즐길 수 있을 것으로 예상된다.

1. Glossary
   1. Objective

Glossary에서는 본 문서에 등장하는 기술적 용어들과 축약어들을 정의한다. 해당 문서를 어느 누가 읽어도 이해할 수 있도록 모든 용어에 대해 설명한다.

* 1. Term Definitions, Acronyms and Abbreviations
     1. Term Definition
        1. User & Service 관련 용어

|  |  |
| --- | --- |
| Terms | Definition |
| 참여자(Player) | 게임에 참여하는 사람들을 말한다. |
| 사회자(Moderator) | 게임의 진행을 제어하고 원활하게 진행을 이끄는 자를 말한다. |
| 왕(King) | 게임 내에서 한 팀의 왕으로, 생존을 함으로써 팀을 승리로 이끄는 자를 말한다. |
| 멤버(member) | 게임 내에서 한 팀의 왕을 제외한 구성원으로, 추리를 통해서 왕을 보호하고, 상대 구성원 및 왕를 죽이는 자를 말한다. |

* + 1. Acronym and Abbreviation

1. User Requirement Definition
   1. Objective

User Requirement Definition에서는 사용자의 요구사항을 User Level에서 서술한다. 크게 Functional Requirement와 Non-Functional Requirement로 나누어 서술한다.

* 1. Functional Requirements
     1. 게임의 사회자

[봇]은 사회자로서 다음의 기능들을 관리한다.

* + - 1. Start/End Game

Discord를 통해 게임을 시작하는 명령어가 입력될 경우, [봇]은 게임을 시작하기 위한 준비 과정에 들어간다. 이 과정에서 게임에 참가하는 참여자, 진행할 게임의 종류, 세부 규칙 등의 설정을 할 수 있다. 모든 준비가 끝났다면 준비 완료 명령어를 입력하여 게임 시작단계로 들어간다.

게임이 시작되면 [봇]은 모든 참여자에게 팀과 역할 그리고 제약 조건을 부여한다. 부여된 정보들은 참여자 본인만 확인할 수 있도록 개인 메시지를 통해 전달한다.

게임 도중 제외된 참여자의 역할이 왕이었다면, 게임의 승패를 가릴 수 있게 된다. [봇]은 게임 종료를 선언하고, 결과를 발표한 뒤, 참여자들의 전적을 갱신한다.

* + - 1. Q & A Management

통상적인 게임 단계에서 [봇]은 참여자들의 문답을 관리해주는 상태가 된다. 한 참여자가 Discord에 질문으로 여겨지는 text를 입력할 경우, [봇]은 질문을 해석한 뒤 질문 대상자에게 대답을 할 것을 요구한다. 대답은 긍정 또는 부정 두 가지 형태가 존재할 수 있으며, 그 외의 대답과 대상자가 아닌 참여자가 입력한 text들은 혼선을 피하기 위해 무시된다. 질문 대상자가 대답 text를 입력하면, [봇]은 질문과 대답 그리고 대상자의 제약 조건을 비교하여 대상자가 올바른 대답을 하였는지 검증해준다. 만약 질문 대상자가 부적절한 대답을 했을 경우 [봇]은 대상자에게 패널티를 부과한다. 이 과정이 정상적으로 이루어진 뒤 [봇]은 다시 질문을 받아들이는 상태가 된다. 질문과 대답 text는 별도의 명령어 없이 자연어 형태로 입력할 수 있다.

* + - 1. Vote

*투표 시스템에 대해서는 개선할 부분들이 필요함*

* + - 1. Help/Log

Discord에 도움말 명령어를 입력하는 것으로 [봇]을 사용하기 위해 필요한 명령어나 게임의 룰을 볼 수 있다. 또한, 게임 도중에 로그 명령어를 입력하는 것으로 지금까지 이루어진 문답 내역을 볼 수 있다.

* + 1. 전적 기록

참여자들의 승패 기록은 매 게임마다 자동적으로 데이터 베이스에 저장된다. 프로필 명령어를 입력하는 것으로 자신의 게임 기록, 승률, 순위 등을 확인할 수 있으며, 원한다면 기록을 초기화할 수 있다.

* 1. Non-Functional Requirements
     1. Product Requirement
        1. Performance Requirement

사람이 사회자를 맡는 경우와 다르게 bot은 게임 도중 발생한 문제를 적절히 해결하고 조율하는 능력이 없다. 따라서, [봇]이 조금이라도 잘못 작동하면 게임의 전체적인 양상이 이상해질 수 있다. 전체적인 게임의 진행을 bot에게 맡기는 이상 [봇]의 performance가 확실해야 한다.

Discord에 입력되는 text들을 빠르고 정확하게 처리하여 게임에 혼선을 빚는 일이 없어야 한다. 자연어 text를 해석해야 하는 경우, [봇]이 오역을 하는 일이 없도록 확실한 학습이 필요하다.

* + - 1. Efficiency Requirement

[봇]은 서버에 의해서 항시 실행되는 상태를 유지하고 있다. 따라서, 서버에 부하가 걸려 [봇]의 작동 효율이 나빠지는 것을 방지하기 위해 이동하는 데이터의 양을 최소화해야 한다. 복잡한 연산이 필요한 부분은 [봇] 내부에서 처리하는 것 보다 외부 API를 적극적으로 활용하여 서버 유지에 필요한 자원을 줄인다.

* + - 1. Dependability Requirement

[봇]에게 충분한 자연어 해석 학습을 시켜도 오역을 하는 경우가 발생할 수 있다. 이런 경우 봇이 해석한 바를 참여자에게 전달하지 않으면, 사용자의 해석과 시스템의 해석 사이에서 발생한 차이가 메워지지 않은 채로 게임을 혼란스럽게 만들 수 있다. 이를 방지하기 위해서 [봇]은 해석한 바를 참여자가 볼 수 있게 출력한다. 오역이 발생하지 않은 경우 [봇]이 올바르게 작동하고 있음을 시각적으로 확인할 수 있고, 오역이 발생하더라도 참여자가 정정할 기회를 줘서 [봇]에 대한 신뢰도가 높아진다.

* + - 1. Usability Requirement

별도의 가이드라인을 제공하지 않아도, [봇]을 사용하는 데 어려움이 없어야 한다. [봇]이 Discord 상에 메시지를 출력할 때마다, 도움말 명령어를 입력하여 자세한 정보를 볼 수 있다는 문구를 첨가하여, 사용자가 도움말을 읽고 [봇]을 사용하는 방법을 익힐 것을 권고한다. 게임 도중 진행 상황을 참여자들이 혼동하지 않도록, 현황 정보를 알려주는 메시지를 눈에 띄게 출력한다.

* + - 1. Security Requirement

[봇]을 사용할 때 발생할 수 있는 개인정보와 보안 문제점들은 대부분 Discord의 시스템에 기인한다. [봇]과 사용자 사이에는 항상 Discord가 존재하기 때문에, Discord 상에 보안 문제가 없다면 [봇]을 사용할 때도 문제가 발생하지 않는다. [봇]이 자연어를 처리하는 과정에서 개인 정보가 포함된 text를 데이터 베이스에 저장하는 일을 방지하기 위해서, 모든 text는 게임에 연관된 정보만을 논리식으로 번역하여 다루도록 한다.

* + 1. Organizational Requirement
       1. Environmental Requirement

Discord bot으로서 [봇]의 대부분 기능은 Discord API에 묶여있다. 따라서, [봇]은 Discord에서 사용하기에 최적화되어 있어야 한다. Discord는 PC 프로그램과 모바일 어플리케이션 두 종류의 플랫폼을 지원한다. [봇]은 두 플랫폼 모두에서 사용될 것을 고려하여 만들어져야 한다.

* + - 1. Operational Requirement

[봇]은 자연어의 해석을 위해 Dialogflow API와

* + - 1. Development Requirement
    1. External Requirement
       1. Regulatory Requirement
       2. Accounting Requirement
       3. Safety/security Requirement

1. System Architecture
   1. Objective

System Architecture에서는 각 Component들의 hierarchy에 대하여 Diagram등을 이용해 서술한다.

* 1. Game Manager System
  2. User Profile Manager System

1. System Requirement Specification
   1. Objective

System Requirement Specification에서는 User Level에서 정의되었던 Functional/Non-Functional Requirement들을 Developer Level로 세분화하여 서술한다.

* 1. Functional Requirements
  2. Non-Functional Requirements
  3. Scenario
     1. Join Scenario
     2. Question & Answer Scenario
     3. Winning Scenario
     4. Managing User profile Scenario

1. System Models
   1. Objective

System Models에서는 System Component, System, 그리고 System Environment 사이의 관계를 Figure 및 Diagram을 통해 서술한다.

* 1. Context Models
  2. Interaction Models
  3. Structural Models
  4. Behavioral Models

1. System Evolutions
   1. Objective

System Evolution에서는 시스템이 세우고 있는 주요한 가정들과 사용자의 Needs 변화, 혹은 추후 Update될 사항들에 대해서 서술한다.

* 1. Limitations and Assumption
  2. Evolutions of User Requirement
     1. Voice Recognition-Based System
     2. Rule Modification Sandbox
     3. New game mods

1. Appendices
   1. Objective

Appendices에서는 본 프로젝트와 관련한 참고사항들을 서술한다.

* 1. Database Requirements
  2. User-System Requirements
  3. Development Process

1. Index
   1. Table Index
   2. Figure Index
   3. Diagram Index
2. Reference