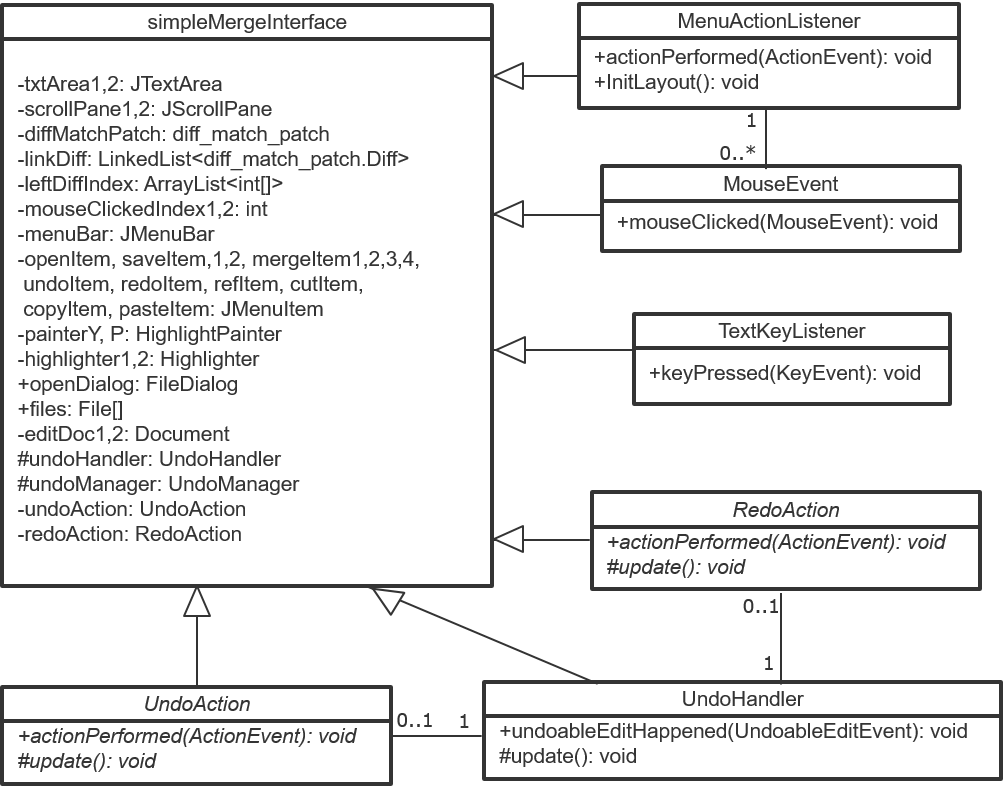
**Term Project Report**

**20155714 주은미 / 20151199 김나영**

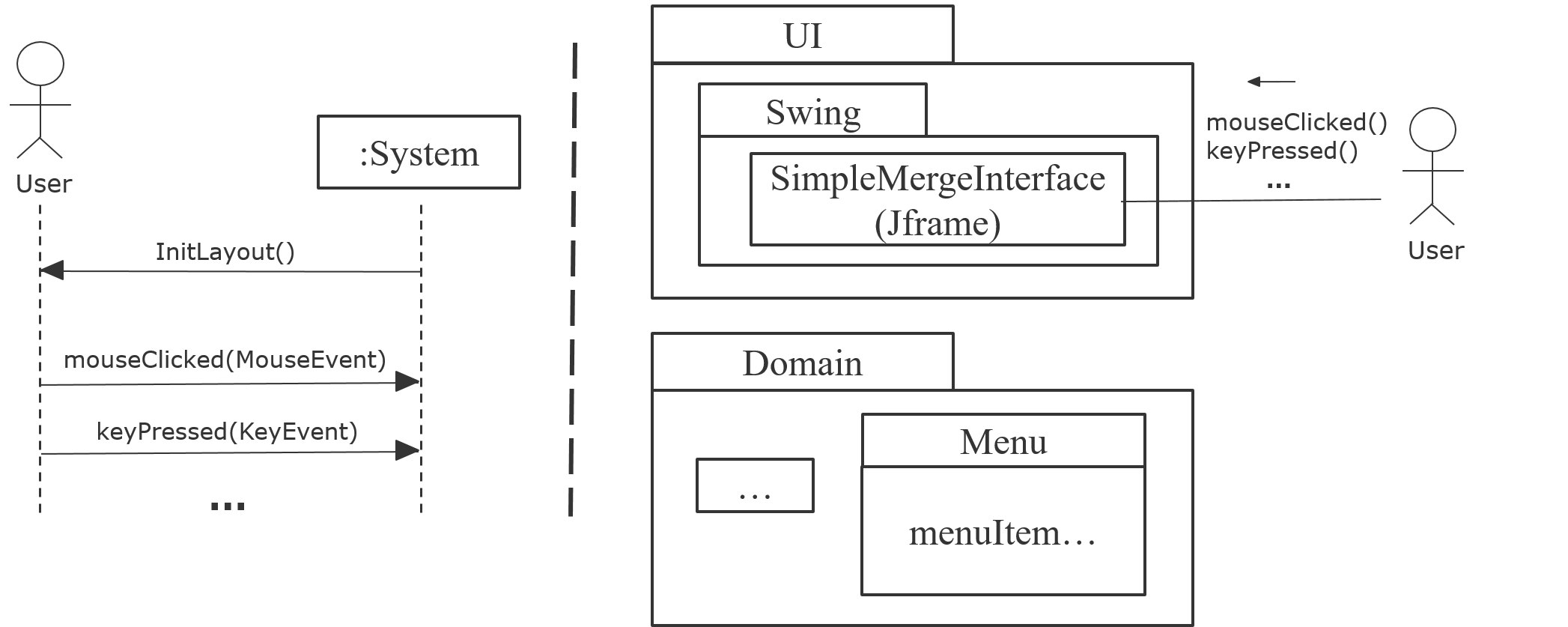
**Analysis & Design, Implementation, and Test Document:**

* Domain model (utilize class diagrams)



SRS에 기술한 요구 사항에 따라 수행할 기능을 보여주는 frame interface를 바탕으로 여러 class가 연결된 형태다. MenuActionListener는 각 메뉴가 실행해야 하는 액션에 대해 정의하고 있으며 어떤 메뉴가 선택 되었는지를 ActionEvent를 통해 받고 이에 따라 정해진 메뉴의 기능을 수행한다. 사용자는 UI의 메뉴와 단축키를 이용해 프로그램을 조작할 수 있고, 이 입력 값에 따라 시스템은 사용자가 원하는 기능을 제공한다.

* Software Architecture + Design Model

설계함에 있어서 가장 중요하게 생각한 부분은 일반적인 윈도우 프로그램에 위배되는 사항이 없도록 하는 것이었다. 윈도우에서 제공하는 기본적 기능을 충분히 활용하고, 통용되는 단축키를 그대로 사용함으로써 사용자로 하여금 여타 프로그램을 사용할 때와 같이 불편함이 없도록 하였다.

GUI는 스윙(Swing)을 활용해 전체 프레임을 구성했다. 프레임의 상단에는 메뉴바가 있고, 그 아래에는 비교하고자 할 두 파일을 보여주는 텍스트 뷰 두 창을 같은 비율로 그리드 레이아웃 시켰다. 메뉴바는 ‘File’, ‘Merge’, ‘Edit’의 세 항목으로 각 기능 별로 분류된 메뉴들로 구성했다. 우리의 주된 설계 방향에 맞게 통용되는 형식을 차용해 직관적인 인터페이스와 단축키를 사용했다. (이를 테면, 저장에는 ‘ctrl+S’, 실행 취소에는 ‘ctrl+Z’ 등이 있다.) 그러면서도 ‘Refresh’ 기능과 같이 우리가 임의로 이름 지은 기능은 그의 약자를 이용해 단축키에 활용했다.

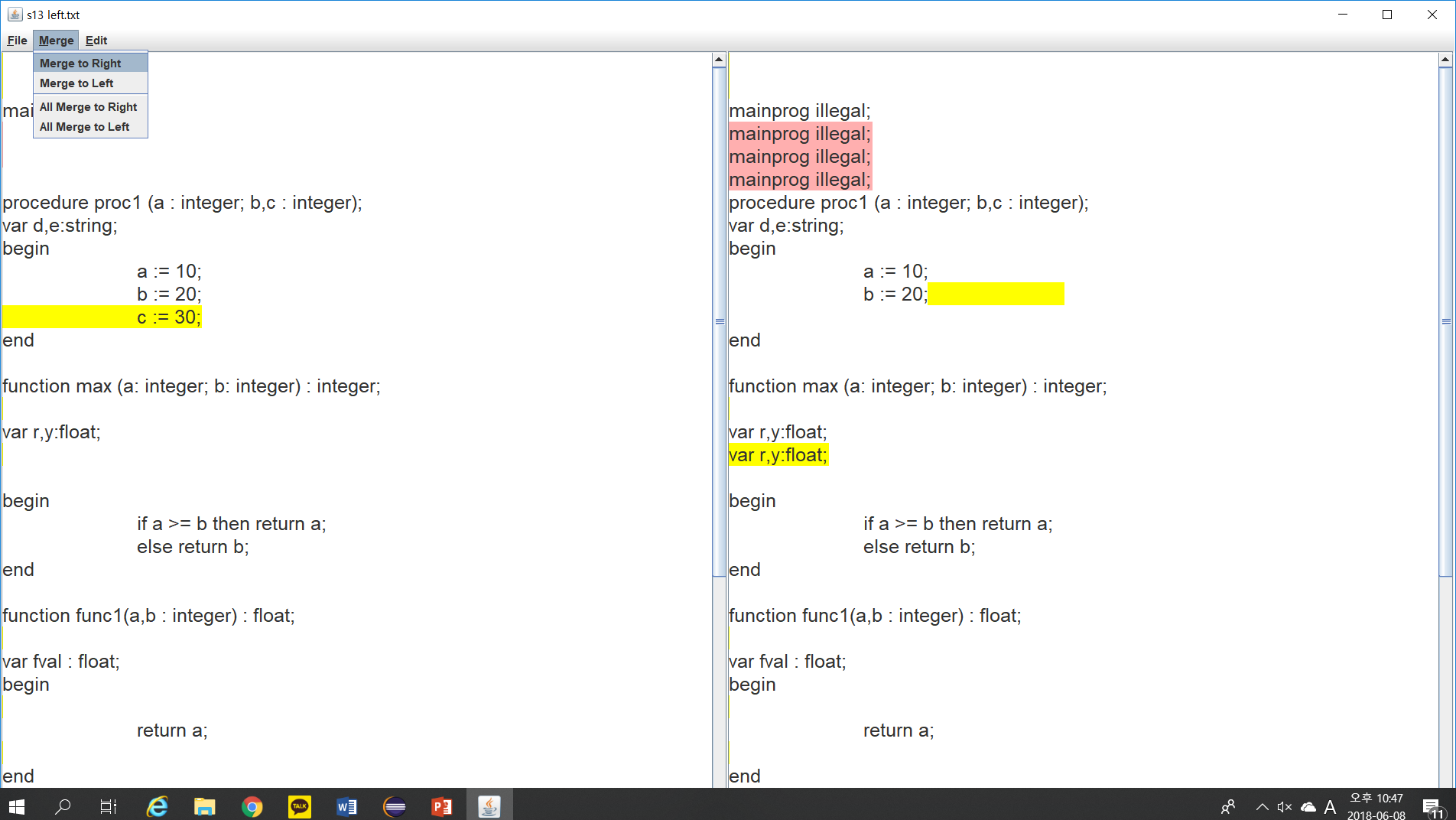
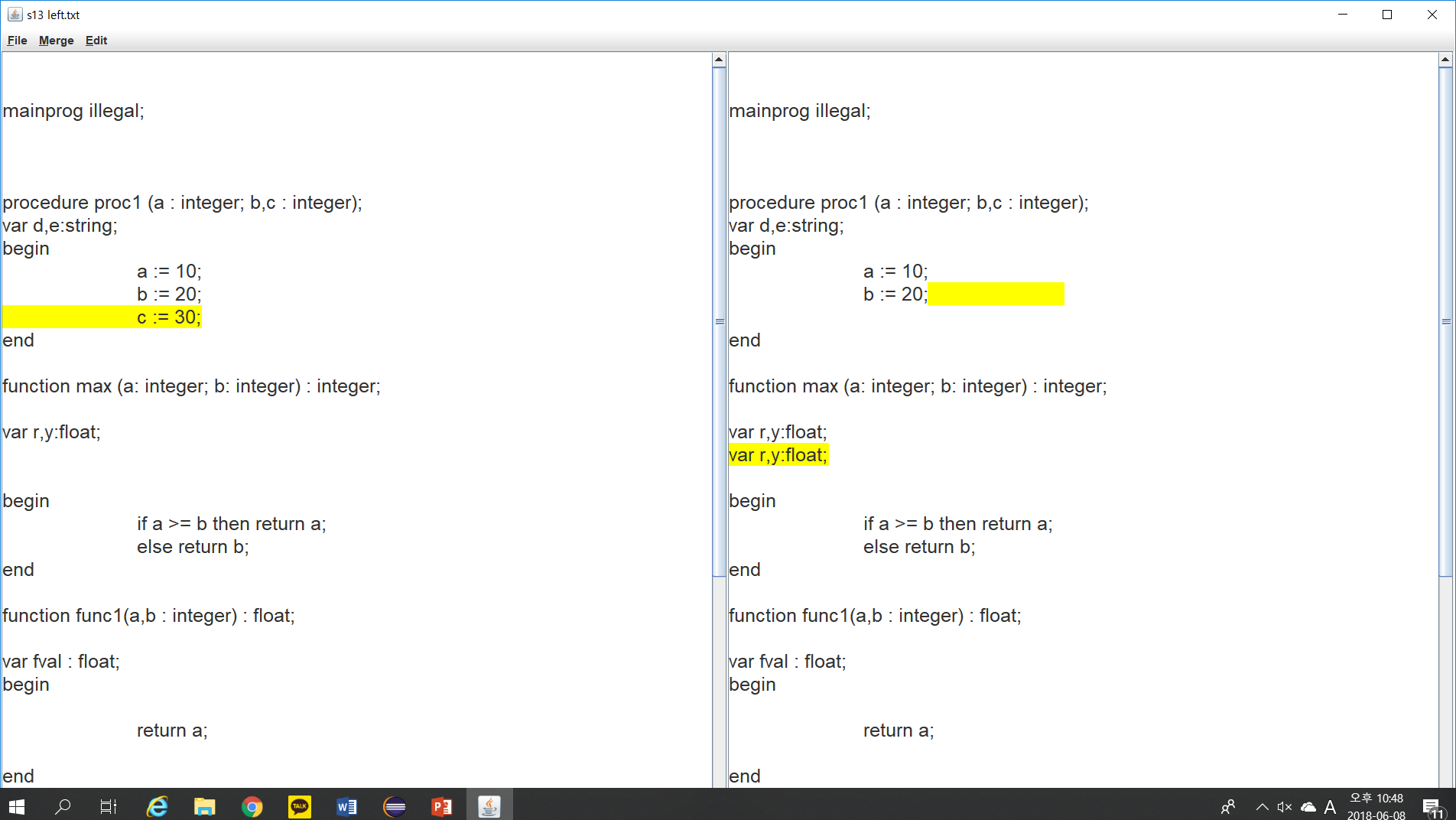
텍스트 창은 JTextArea 이용해서 양쪽에 구별되게 표시해 스크롤될 수 있는 영역이게 했다. 또한, 두 텍스트의 비슷한 구문에 따라 한 쪽 텍스트 뷰에 커서를 두고 스크롤할 때, 동시에 스크롤될 수 있도록 했다. 텍스트창 내에서 텍스트를 다루는 방법은 일반적인 노트패드 프로그램과 동일하다. cut, copy, paste 기능은 윈도우에서 기본적으로 제공하는 단축키로도 가능하지만 단축키를 사용하지 않는 경우를 위해 Edit 메뉴에 추가했다.

SRS를 작성할 때부터 쓸모 없는 항목을 최소화하고 프로그램에 반드시 필요한 기능에 대해서만 기술했고, 초기에 생각한 기능을 모두 구현했다. Original SRS에서 놓친 부분인 ‘Edit - Cut’과 각각을 따로 저장하는 ‘File - Save Left As’, ‘File – Save Right As’ 기능을 추가해 1.1 버전의 SRS를 작성하였다.

개선해야 할 점으로는 각 텍스트 뷰가 인식하는 실행 취소가 가능한 수정 사항에는 우리가 ‘Merge’에 사용하는 replace하는 함수를 통한 변화가 포함되지 않는다는 것이다. 여러 변수를 통해 ‘Merge’한 내용도 ‘Undo’, ‘Redo’할 수 있게끔 보완했지만 1번까지만 가능하다는 한계가 있다.

다만, Java를 사용함에 있어서 어려움이 있어 개발 초기에 간단한 기능은 UI를 구성함과 동시에 구현했는데, 결론적으로 우리의 목표와 멀어지 게 한 오류를 범했다고 할 수 있다. 프로그램을 발전시키면서 그 이후 구현하는 것들은 주로 Oracle에서 제공하는 Java Documentation[[1]](#footnote-1) 의 기본 함수들 예제를 참고해 별개의 클래스로 작성함으로써 각 유닛이 서로 독자적으로 연결될 수 있도록 했다.

하지만, 주가 되는 UI를 표현하는 클래스가 단위가 너무 커져 유닛 테스트를 하기에 적절하지 않았고 testable한 코드와 testable하지 않은 코드 간에 연결점을 제대로 파악하지 못해 코드를 고치기 어려웠다. 프로그램 구현 중에 유닛 테스트를 위한 환경을 만들었지만, 안타깝게도 요구 사항을 최대한 반영시키는 것에 더 큰 비중을 두다 보니 끝내 테스트 케이스를 작성하지 못하고 아직 도전 과제로 남았다.

* usage of program & screen shots of examples

**Project Management Report**

* **GitHub URL :** [**https://github.com/ny8585/studious-guacamole**](https://github.com/ny8585/studious-guacamole)
* Briefly explain our project progress history

GitHub 프로젝트 개설 이후 5월 중순까지 SRS 1.0을 작성했다. 이후 본격적으로 개발에 착수한 후 매주 업데이트 했으며, 팀원 주은미가 Merge와 View 등의 핵심적인 기능을 개발했고 팀원 김나영은 주로 UI와 보조적 기능을 개발했다. 6월 초 기존 SRS 1.0의 대부분의 기능을 구현한 후, 빠뜨린 요구 사항을 개선해 SRS 1.1을 작성했다.

* Briefly explain our experience

두 팀원 모두 java로 프로그램을 개발한 적이 없어 오라클 사에서 제공하는 다수 예제를 참고했으며, 그럼에도 이상적인 객체 지향적인 설계에는 가깝지 않지만 java 프로그래밍을 실무적으로 익힐 수 있었다.

초기에는 GitHub를 사용하기 보다는 각자 부분을 나누고 해당 코드를 완성시키면 GitHub에 게시하는 식으로 진행했는데, 사용에 익숙해 지고 나니 이클립스로 바로 불러오는 기능이나 간단한 코드 수정을 GitHub에서 간편하게 함으로써 능률을 향상시킬 수 있었다.

무엇보다 이제껏 프로젝트를 진행할 때 코드 작성부터 시작했다면, SRS를 자세히 기술함으로써 Use case를 기능이 서로 어떻게 연관되는 지, 어떤 흐름에 따라 분기되는 지 등을 팀원 모두가 미리 파악하고 코드를 작성할 수 있다는 것이 크게 다른 점이었다. 또한, 개발 중에 기능을 개발함에 있어서 요구사항을 확인해 빠뜨리지 않을 수 있었다.

1. URL : https://docs.oracle.com/javase/tutorial/uiswing/components/ [↑](#footnote-ref-1)