안녕하십니까. 이번 발표를 맡게 된 6조의 PM인 컴퓨터과학과 임현 입니다.

그럼 2017년 소프트웨어공학 발표를 시작하도록 하겠습니다.

‘해야 할 일, 빠짐 없이’ 저희 팀 TodoList 프로젝트의 슬로건입니다.

지금부터 저희가 이 슬로건을 지키기 위해 어떻게 하였는지 시작하겠습니다.

자, 목차 부분을 보시면 발표는 크게 6가지 주제로 분류되어 있으며, (소개, 요구사항 분석, 프로젝트 계획, 설계, 구현 그리고 테스팅)으로 구성되어져 있습니다. 피피티는 총 19페이지로, 발표 중간 질문사항이 있으실 경우 중간에 질문 하셔도 무방합니다.

조직도 및 역할입니다.

PM 및 메인 개발자인 임현, 데이터베이스 담당 김태정, UI 담당 박다빈, TEST 담당 신준섭, 프로그램의 전체적인 부분을 도와준 서브 개발자 황은선이 있습니다.

프로젝트의 시작은 요구사항 분석으로 시작됐습니다. 기본 요구사항으로는 왼쪽과 같이 5가지가 있었고, 개발 도중에 2번 요구사항에 대하여 완료 항목에 대해서는 숨기기가 가능해야 하며, 다시 보이기도 가능해야 한다. 가 추가됐다는 특이점이 있습니다.

★ 저희는 이에 그치지 않고, 팀 요구사항으로 로그인 기능을 구현했고, 로그인 기능 안에는 기본적으로 로그인, 회원가입 그리고 로그인 문제 해결, 즉 비밀번호 찾기 기능이 구현되어 있습니다. 저희가 로그인 기능을 추가한 구체적인 이유는 뒤에 프로젝트 계획에서 설명해도록 하겠습니다.

다음은 저희 팀 USE CASE 부분의 일부를 가져온 것 입니다. 저희 팀은 USE 케이스를 이용하여 기능 하나하나를 빠뜨리지 않고, 구현에 옮기기 위해 노력했습니다.

이제부턴 프로젝트 계획에 대해서 설명해드리고자 합니다.

저희는 요구사항이 분석이 끝나자마자 프로젝트 계획에 들어갔습니다. 개발 대상 및 목적은 모든 팀과 똑같이 to do List 관리 프로그램을 개발하였고, 구현 환경은 Windows 10에서, 요구하신 JDK1.8로 JAVA를 통하여 개발하였습니다.

주요 일정입니다. 전형적인 Water Fall, 즉 폭포수 모델이며, 이 모델의 단점인 요구사항이 명세서가 제대로 되지 않을 경우, 프로젝트가 실패할 가능성이 높기 때문에 처음 요구사항 명세서부터 중점을 두고 프로젝트를 하였습니다.

저희는 우선 처음 주신 프로세스 상세 스케줄 pdf 파일을 보고 전체적인 흐름을 구성했고, 그 외 자세한 부분은 Git에 올려주신 템플릿을 보며 정확한 일정을 WBS를 통해 나누었습니다.

WBS를 통해 나눠진 일정은 왼쪽의 Gantt차트와 오른쪽의 Pert 차트로 나누어서 Critical Path를 정확히 알고, 어떤 일정을 하지 않으면 프로젝트가 늦어지는 지 확인하고, 주요 경로는 꼭 맞추어 프로젝트를 하고자 다짐했습니다. 그 외 Critical Path가 아닌 부분은 여유를 두어 수행했습니다.

저희 팀은 크게 4가지 부류의 도구를 사용했습니다. 먼저 ★ 산출물을 관리한 깃과 드롭박스, ★프로젝트의 이슈를 관리한 레드마인, ★ 프로젝트 개발 및 구현에 쓰인 이클립스, 마이에스큐엘, 마븐과 젠킨스, ★ 문서 관리에 쓰인 워드, 피피티, 엑셀 그리고 PDF 입니다.

저희는 이 중에서도 남들과 다르게 쓴 도구들을 중점적으로 설명해드리고자 합니다. 우선 산출물 관리에서는 저희 팀원이 ★ 클라우드 서비스인 Dropbox에 가공되지 않은 데이터를 올리고, 이에 대한 검증이 끝나면 ★ 형상관리 도구인 깃에 올리는 시스템을 구축했습니다. 저희가 굳이 깃에 바로 올리지 않고, Dropbox를 거친 이유는 프로젝트 자체가 일반적인 프로젝트가 아닌 깃도 평가 요소에 들어가기 때문에 가공되지 않은 데이터는 거르고 싶었기 때문입니다. 또한 클라우드 서비스의 대표주자인 Dropbox를 프로젝트에 활용해보고 싶었던 마음도 컸습니다.

혹시 자바의 모토를 알고 계신 분 있으십니까?

자바의 모토는 W O R A , 즉 ★★ Write Once, Run Anywhere 으로 자바를 만든 제임스 고슬링이 한 개발 코드를 한 번만 작성하면 어떠한 플랫폼에서 쓸 수 있길 원했습니다.

저희 또한 이 점을 지키고 싶어서 ★ 데이터베이스 도구 중 하나를 사용하여 어느 플랫폼에서든지 저희 프로그램과 동기화가 됐으면 하는 바람에 MYSQL을 사용하였습니다.

다음은 설계 부분입니다.

저희 팀의 클래스 설계서는 크게 클래스 명, 개요, 스테레오타입, 퍼블릭 인터페이스가 있습니다. 그 중에서도 스테레오타입으로 바운더리, 컨트롤, 엔티티 부분으로 나누어 구현에 있어서 인터페이스와 기능 그리고 그 둘을 이어줄 부분을 확실히 구분 해 두었습니다.

다음은 UI 설계서입니다. UI 설계서에는 화면 순서와 항목 속성을 미리 알아두어, 프로그램의 구조가 어떻게 돌아가는지, 어떠한 순서로 실행되는지를 파악했습니다.

다음은 구현 부분입니다.

저희 팀은 Javadoc라는 주석 표준을 사용하여, 코드 맨 앞 부분에 왼쪽과 같이 기술하였습니다. 이를 통해 작성자와 코드 설명 그리고 변경 이력을 쉽게 확인함으로써 Git의 특성인 프로그램이 갑자기 오류가 생겨서 돌아가지 않을 시, 어느 버전으로 돌아가야 할 지 쉽게 볼 수 있게 하였습니다.

주석 표준의 실제 예시입니다.

마지막으로 실제 테스트 영상 보시도록 하겠습니다.

영상처럼 우선 회원가입을 합니다.

비밀번호를 까먹었을 시, 로그인 문제 해결을 눌러서 비밀번호를 확인할 수 있습니다.

자신의 아이디와 비밀번호로 들어갔을 시 TodoList 메인 창이 뜨게 됩니다.

우선 과목 정보에 아무 정보도 없으니 과목을 추가해주는 모습입니다.

과목이 추가 된 후 실제 TodoList를 넣고 있습니다.

추가, 변경, 삭제 해주는 모습입니다.

또한 완료로 변경한 항목에 대해서 숨김 및 보이기가 가능하다는 것을 보여주는 모습입니다.

이상으로 6조의 발표 마치도록 하겠습니다. 감사합니다.

질문 사항 있으신 분 있으십니까?

그럼 이상으로 6조의 Blossom 발표 마치도록 하겠습니다. 감사합니다.