1. Draft Plan

 Step 1 – Motivation and Objective of project

대다수의 대학생들은 수강신청 기간이 되어서야 계획을 세우고 준비를 한다.

성공적인 수강신청을 하기 위해 원하는 수업들의 목록작성에 많은 수고와 노력을 쏟아 붓고 있다.

컴퓨터가 발달 된 요즘에서야 종이와 연필로 일일이 수업 정보를 적는 학생은 없겠지만, 윈도우 노트패드와 엑셀을 오가며 상당히 귀찮은 작업을 하고 있는 것이 현실이다. 그리고 수강신청을 아직 경험 못한 갓 입학한 신입생들에게는 그것은 낯 설은 시스템일 것이다.

그리하여 재학생들에게는 성공적인 수강신청 플랜을 제공하고, 신입생들에게는 수강신청 시스템에 쉽게 적응하기 위해 현 시스템과 유사한 간편한 수강신청 계획 프로그램을 계획하게 되었다.

 Step 2 – Scope of project

 실재하는 현 시스템과 유사하지만, 최대한 간단한 프로세스로 플랜을 세울 수 있는 최소한의 범위 내에서 개발.

 Step 3 – Functional requirements

 \* 교수는 수업을 개설 및 삭제 할 수 있다.

\* 학생과 교수는 전체 수업리스트를 조회 할 수 있다.

\* 직관적으로 쉽게 계획을 세우기 위해 수업 리스트들은 각 항목별로 정렬할

수 있다

\* 교수는 개설한 수업에 소속된 학생들의 리스트를 열람 할 수 있다.

\* 수강신청 불가능한 수업은 별도의 표시를 하여 눈에 잘 띄게 한다.

\* 학생은 자신이 신청한 수업들만 따로 리스트로 조회 할 수 있다.

\* 학생은 신청한 수업의 정정 또는 삭제가 가능하다.

\* 계획한 수업리스트는 따로 텍스트나 엑셀 파일 형식으로 저장할 수 있다.

 Step 4 - Non-functional requirements

\* 1년에 많아야 2,3번 쓰이는 빈도이므로, 별도의 교육 없이 바로 사용할 정도로 간단하게 구축 되어야 한다.

\* 반응 시간은 5초 이내

 \* 어느 환경에서나 실행가능 하도록 JAVA 플랫폼으로 개발.

\* 쉽고 간단한 GUI개발을 위하여 JAVA SWING을 사용하여 구현

 Step 5 – Estimate resources

 \* Human Resources : 3명

 분석 및 설계, 구현, 검증 단계 중 형편 되는대로 같이하거나 나눠서 맡는다.

 \* Duration : 4주

 \* Budget : 개인당 식비(커피값 포함) - 4500원

1주에 2번 조 모임 -> 4 X 2 X 4500 = 36,000(원)

(각자 본인 부담, 먹는 식사 메뉴에 따라 상이할 수 있음)

2. Preliminary Investigation Report

 Step 1 – Alternative solutions

 \* 과거에 유사한 프로그램 작성 경험과 코드 재사용을 바탕으로 더 발전된 시스템 구축

 Step 2 – Project’s justification (Business needs)

 \* 현재 배우고 있는 소프트웨어 공학 분야의 한 부분인 OOAD를 직접 수행 함으로서 각 팀원 들의 실력 향상 기대

 \* 추후에 있을 실제 수강신청 기간 중 완성된 프로그램을 통한 비용 감소 기대

 Step 3 – Risk analysis, and risk reduction plans

3-1 Risk analysis

|  |  |  |
| --- | --- | --- |
| 위험 | 가능성 | 영향 |
| 팀원 잠적 | 보통 | 많이 심각함 |
| 개발 도구(노트북) 파손 | 낮음 | 중요하지 않음 |
| 요구사항 변경 | 높음 | 많이 심각함 |
| 개발시간 과소 추정 | 보통 | 심각함 |
| 소프트웨어 규모 과소 추정 | 보통 | 심각함 |
| 구현 가능한 코딩능력의 한계 | 높음 | 심각함 |
| 소스코드 및 관련 자료 손상 및 증발 | 낮음 | 재앙적 |

3-2 Risk reduction Plans

|  |  |
| --- | --- |
| 위험 | 계획 또는 전략 |
| 팀원 잠적 | 팀원 서로에게 따뜻한 관심 또는 감시  |
| 개발 도구(노트북) 파손 | 학교 컴퓨터로 대체 |
| 요구사항 변경 | 개발 기한 내에 반영 가능할 경우 변경 |
| 개발시간 과소 추정 | 철야 작업 강행 |
| 소프트웨어 규모 과소추정 | 철야 작업 강행 |
| 구현 가능한 코딩능력의 한계 | 주변 능력자들에게 도움 요청 |
| 소스코드 및 관련 자료 손상 및 증발 | 저장공간에 주기적인 백업 수행 |

 Step 4 – Business market analysis

\* 쉽고 간단한 사용을 강점으로 사용자에게 어필 가능.

 Step 5 – Managerial issues

\* 개발 중 의견충돌 발생시 결정은 PM이 내린다.

\* 최종 발표일 (2010-12-08) 까지는 시스템 구축이 완료되어야 한다.

3. Requirement specification

1. Introduction

 1.1 Purpose

이 문서는 수강신청 계획 프로그램 개발을 위한 소프트웨어 요구사항 명세서이다.

시스템 개발에 있어서, User 및 Stakeholder, 시스템 Requirement들을 도출하여 체계적으로 진행함으로써 개발 비용을 최소화 하는 것이 이 문서의 목적이다.

 1.2 Product scope

기존까지는 학생들이 수강신청 계획을 세우기 위해, 종이와 펜을 이용하여 일일이 수업목록을 작성하거나 컴퓨터를 이용하여도 학교 웹 페이지와 기타 다른 보조 응용프로그램(노트패드, 그림판, 엑셀 등)들을 여러 개 띄워서 작업해야 했기에 비교적 비용이 많이 들어갔다. 그리하여 우리는 그러한 수고를 덜어주는 하나의 어플리케이션으로 모든 것을 가능케 하는 프로그램 개발을 목표로 한다.

 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| GUI | Graphic User Interface |
| OOAD | Object Oriented Analysis Designed |
| UML | Unified Modeling Language |

2. Overall Description

2.1 Product Perspective

 사용빈도가 연 1,2 회 정도로 낮지만, 많은 사용자들이 사용 할 것으로 예상 됨

2.2 Product function

 사용자가 모든 수업 리스트를 열람하고, 원하는 항목만을 선택하여 목록 작성이 가능하며 목록은 파일로 저장 및 불러올 수 있다.

2.3 User Characteristics

이 시스템은 현 대학생들을 대상으로 하며, 수강신청을 한번도 경험해보지 않은 신입생들도 쉽게 사용할 수 있어야 하므로, 기초적 컴퓨터지식(키보드 타이핑과 마우스 클릭)을 갖춘 사용자를 대상으로 가정한다.

2.4 Constraints

 \* 사용되는 기술은 이미 충분히 성능이 입증 되었거나 보장되는 것을 사용한다.

 \* 개발에 필요한 소프트웨어는 정품 또는 프리웨어만 사용한다.

2.5 Assumptions and Dependencies

\* Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 요구사항 기능내역 | 설명 | 중요도 |
| R-1 | 전체 수업 보기 | 개설된 전체 수업 리스트를 볼 수 있어야 한다. | S |
| R-2 | 수강한 수업 보기 | 사용자가 수강한 수업리스트를 볼 수 있어야 한다. | A |
| R-3 | 수강한 수업 삭제 | 사용자가 수강한 수업 항목을 삭제 할 수 있다. | A |
| R-4 | 수업 리스트 저장 | 사용자가 수강한 수업리스트를 텍스트파일로 저장 할 수 있다.  | B |
| R-5 | 수업 리스트 불러오기 | 양식에 맞게 저장된 수업리스트 텍스트 파일을 불러 올 수 있다. | B |
| R-6 | 시간표 저장 | 작성된 수업리스트를 바탕으로 엑셀 파일로 저장 할 수 있다. | B |

 \* Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 요구사항 기능내역 | 설명 | 중요도 |
| R-7 | 사용자 편리성 | 마우스 클릭만 할 줄 알아도 사용 가능해야 한다. | S |
| R-8 | 성능 | 모든 작업은 5초 이내에 이루어 져야 한다. | A |

4. Define Use Case Diagram



5. Define Class Diagram

