Software Requirements Specification (SRS)
Course Selection Database

Author: Yuju Weng, Zi Hu, Houmin Sun

Instructor: Mustafa Misir

1 1 Introduction

The following Software Requirements Specification (SRS) document provides a concise overview of the requirements and specifications for the Course Selection Database software. This document covers various critical aspects of the software, including functionality, design framework, and implementation details. The content of the SRS is presented in an organized structure to support its development.

1.1 1.1 Purpose

The primary purpose of this SRS document is to establish a clear and thorough understanding of the requirements for the Course Selection Database software. It defines the functional and non-functional specifications to ensure all stakeholders, including the development team, quality assurance staff, project sponsors, and other participants in the software lifecycle, have a consistent and comprehensive point of reference. This document serves as a foundation for aligning objectives and requirements across all stages of the Course Selection Database software's development.

1.2 1.2 Scope

The Course Selection Database is designed to assist students at DKU in managing and planning their academic schedules efficiently. This software will support students in selecting courses that align with their academic goals while addressing various constraints, such as prerequisites, anti-requisites, and scheduling conflicts. By leveraging this database, students will receive a customized course plan that accounts for both the institution's academic requirements and individual preferences.

Application Domain: This software falls within the academic advising and course scheduling domain, integrating data and rules from DKUHUB and providing a user-centered interface. Its primary role is to facilitate complex scheduling, ensuring that students meet their program requirements and optimize their semester schedules without conflicts.

Objectives and Goals: The system will assist students in arranging courses in the following areas:

1. Avoid time conflicts.
2. Managing prerequisite and anti-requisite requirements.
3. Planning a clear path to meet all credit and graduation requirements.
4. Meet other personalized needs of the users such as time periods and instructors of the courses.

Functional Boundaries: The Course Selection Database will generate course plans based on data from DKUHUB, including student input on course interests, previously completed courses, and personal availability. This tool will not act as a definitive advisor but rather as a scheduling assistant. While it will provide recommendations, students may need to consult academic advisors for final confirmation on critical academic decisions.

1.3 1.3 Organization

This Software Requirements Specification document is structured into several subsections. The first section covers the explanation of the document's Purpose, Scope, and Organization. The second section is Overall Description, which is divided into five distinct parts, each elaborating on specific aspects of system usage and the associated actions. These sections include Product Perspective, Product Functions, User Characteristics, Assumptions and dependencies, and Apportioning of Requirements. In the third section, you will find Use-case, Sequence, State, and Class diagrams that model the system.

2 2 Overall Description

2.1 2.1 Product Perspective

- Users can enter a list of requirements and get a specific plan for their course selection
- User Interface
  - Basic Requirements blocks
    - Entering the user's target course ID and end time
    - Entering the courses taken by the user before
      - If the user has been logged into DKUHUB, this can be left blank
  - Optional Advanced requirements
    - For users to enter their other specific requirements
      - Their intended majors
      - Their wanted/unwanted professors

- Their intended time periods
- Their time going to Duke

- Confirming button
  - Button to generate a course selection plan
- Clear button
  - Button to clear all the requirements
- Course selection plan
  - Left View (Occupying a small part of the whole interface)
    - A table showing that all the courses taken grouped by terms and sections
  - Right View
    - A calendar presenting the schedule
    - A button to switch presenting by weeks/months

2.2 2.2 Product Functions

- User Interface
  - A list of blocks for users to enter their requirements
  - A few buttons for users to do operations to the information they enter
  - A result page to display the course selection plan
- Possible requirements for course selection
  - Target courses
  - Courses taken before
  - Intended time periods
  - Intended professors
  - Target majors(all the compulsory courses)
- The result page for the course selection plan
- Error Message
  - Message for Simple errors
    - Straightly inform the user about requirements that are impossible to satisfy
      - conflicts between two anti-requisite course

            - conflicts between time slots and required courses
            - Wrong course ID
        - Inform there are complex conflicts between multiple requirements

- Implementing algorithms for constraints
    - All the course data will be filtered for constraints, and be scheduled afterwards
    - Conflicts between two courses
        - Reasons
            - Anti-requisite
            - time conflicts
        - Translate into 2-sat problems and use Tarjan Scc to solve
    - Pre-requisite requirements
        - Use top-sort to find possible orders for each course
    - Time constraints(May not be needed because the amount of data is small)
        - Use algorithms (KD-tree, Merge sort) concerning partial order models

2.3 2.3 User Characteristics

Students having trouble making plans for complex course selection or any student needing a reference for their course selection plan

2.4 2.4 Assumptions and dependencies

We assume that users need to have a general plan for their course selection. They must have target courses and need to know some basic information for their future calendars so that they won't easily find any conflicts after they use our program. For users to run our program, they must be able to connect to the Internet and have access to DKUHUB so that our program can access the data from the website. Besides, their computers need to install MySQL for a properly functioning database system.

2.5 2.5 Apportioning of Requirements

Managing and organizing the data mentioned above will require extensive operations. By leveraging database management languages, the project can be implemented efficiently, enabling faster data processing. To address the various constraints, algorithms such as topological sorting and Tarjan's algorithm for strongly connected components (SCC) will be used to resolve logical contradictions. The goal is to implement these algorithms within the database system to optimize performance.

3 3 Modeling Requirements

3.1 3.1 Use Cases

3.1.1 3.1.1 Entering Requirements

User enter their requirements using the blocks in the interfaces and click confirm to load data

3.1.2 3.1.2

Getting Data From The Web According to the user's requirements, the program grabs data from DKUHUB and load them into Mysql

3.1.3 3.1.3

Filtering Courses The database filters duplicate courses and courses beyond constraints using algorithms

3.1.4 3.1.4

Sorting Courses The database decides the correct order and schedule for all the courses to satisfy pre-requisite requirements

3.1.5 3.1.5

Presenting Results The result is shown on the interface with two forms, a list and a calendar.

3.2 3.2 Diagram For Use Cases

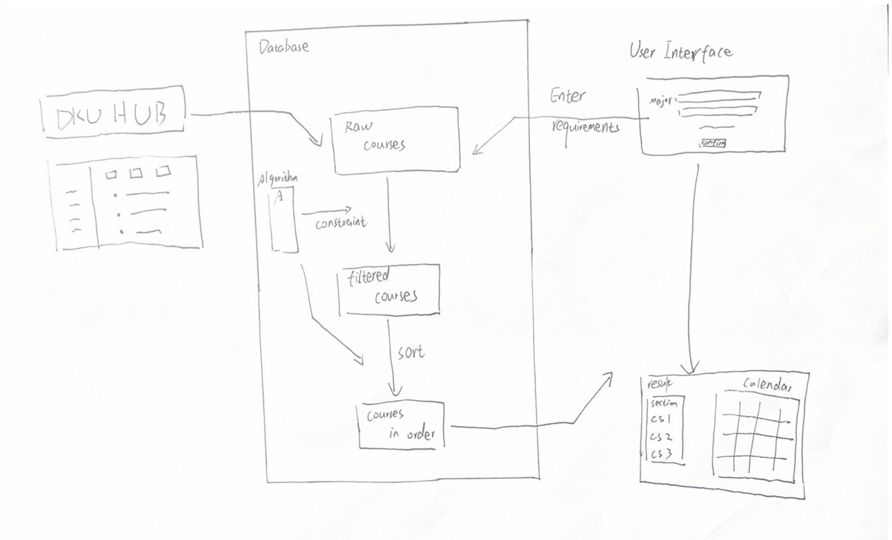

Figure 1: Diagram for Use Cases