Requirements Analysis Document

Introduction

In large tertiary institutions, exam scheduling can be a cumbersome task. This is a difficult and tricky task, traditionally done manually by a team of tireless employees in the Exams and Graduations Office. The system that is being developed is aimed at solving this problem using a well-known computer science algorithm called Graph Colouring. The system will generate a more optimized exam timetable, that will be in the best interests of students by spreading out their exams over time.

Purpose

The purpose of this Requirements Analysis Document (RAD) is to document the results of the requirements elicitation and analysis . This document completely describes the system in terms of functional and non-functional requirements. It also serves as a contractual basis between the client and the developers.

Current system

Currently the Exams and Graduations Office staff have to manually do this cumbersome task of scheduling exam timetables. The large number of students, courses and the sheer diversity characteristic of cosmopolitan institutions can make the task of scheduling exams a bit more than trivial. The difficulty of this task is worsened by among other factors: cross faculty students and practical limitations on the number of exam sessions available, venues, the existence of religious and public holidays and the like.

Proposed system

This problem can be solved to a large extent by a well known computer science algorithm called Graph Colouring. With vertices of the graph represented by Courses and the weight of the edges between them being the number of students two courses share, the graph colouring algorithm aims to provide a colour to each vertex such that no two adjacent vertices have the same colour. A meeting was held with the client and the following system requirements were extracted. The requirements are split into two: functional requirements and non-functional requirements.

Functional Requirements:

These are the requirements that relate to the main purpose/functionality of the system. These are the requirements that determine what the system does and what main functionality should be added.

**The system should let a user upload two mandatory csv files**

On first use the system cannot function without both files.The first CSV file is a list of all courses in the University. The second CSV file is one that has a list of all the registered students and the courses that they are enrolled in.Upon uploading the CSV files the system does a check to see if there are courses that have no students enrolled or if a student is enrolled in a course that doesn’t exist. This data is then stored in a secure online database.

**The system should provide a way of selecting courses that the user needs to schedule**

The user will select the courses they want scheduled and leave out the necessary courses.

**The system should generate a timetable of exams**

The system should have a button that when clicked generates a timetable of exams. and displays it for the user to see.

**The system should be able to display the generated timetable showing exam sessions**

This timetable should show the session and all the exams in that session. This timetable has the minimum number of sessions possible without there being any clashes.

**The system should allow the user to specify a clash parameter**

The clash parameter is the number or proportion of students that two courses must share to constitute a clash therefore not allowing those courses to occupy to be written at the same time. This absolute parameter means that if there are 10 or more students shared between two courses then there is a clash. The relative clash parameter implies that if 10% or more of the students in both courses share the course then that constitutes a clash.

**The system should set an initial class parameter of 1 if the user does not specify a clash parameter**

If the clash parameter is not defined it is set 1.

**The system should allow the user to set a limit on the number of exam sessions**

An input box will be provided that accepts only integers greater than zero. The value of this input box is the maximum number of sessions that the timetable can have.

**The system should increase the clash parameter until the number of sessions in the generated timetable is less than or equal to the max sessions set**

**The system should find specified courses’ positions in the timetable**

The user will enter or select a subset of courses and find in which sessions those courses appear. This is equivalent to adding a hypothetical student and determining their timetable.

**The system must display students with the worst timetables**

Upon having generated a timetable the system will return a list of the top n students with the worst timetables(with the most amount of back to back exams)

**The system must generate the optimal timetable**

**The system must show all the other courses that interact with a specified course**

The system when prompted will be able to view all courses that interact with a particular courses(courses it shares students with) in calendar format.

Non-functional requirements

These are the requirements that do not relate to what the system does but rather the user experience. These requirements address the needs of the client that have nothing to do with the main functionality of the system.

**The system must generate a timetable within 5 minutes**

**The system must have a user interface that is user-friendly**

**The system must provide continuous maintenance**

**The system must provide help tips and support**

User Stories

Sprint 1

|  |  |  |
| --- | --- | --- |
| Identifier | User Story | Size |
| ET-1 | As an EGO member I must be able to upload two CSV files to view courses. | 20 |
| ET-2 | As an EGO member I must be able to select courses that I want to be scheduled | 10 |
| ET-3 | As an Ego member I must be able to generate and View a timetable for **Courses** in a way that minimises the number of clashes | 40 |
| ET-4 | As an EGO Member I must be able to set a limit on the number of sessions I want scheduled to restrict the length of the exam period | 10 |
| ET-6 | As an EGO Member I must be able to set a parameter that defines a clash in order to reduce the number of sessions | 10 |
| ET-7 | As an EGO Member I must be able to select a sorting scheme that allows me to prioritise different paramters of the timetable | 10 |

### Sprint 2

|  |  |  |
| --- | --- | --- |
| Identifier | User Story | Size |
| ET-1 | As an Ego member I must be able to generate and View a timetable for ***Exams*** in a way that minimises the number of clashes |  |
| ET-2 | As an EGO member I should be able to enter a list of courses and find those courses position in the timetable |  |
| ET-3 | As an EGO member I must be able to get the timetable for a particular student in Calendar Format to see should the student enquire |  |
| ET-4 | As an EGO member I must be able to see the students with the worst timetables so that I can make provisions for deferred or Supplementary Exams |  |
| ET-5 | As an EGO member I should be able to view courses that interact with a particular course so that I can interact with the relevant course coordinators to make arrangements |  |