

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б11-мм

Автоматизация процесса обработки расписания СПбГУ

Кузьмина Ксения Антоновна

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
старший преподаватель Я.А. Кириленко

Санкт-Петербург
2024

Оглавление

1. Введение	3
2. Постановка задачи	4
3. Обзор	5
3.1. Обзор существующих решений	5
4. Ход работы	7
4.1. Выбор инструментов	7
4.2. Пользовательский интерфейс	7
4.3. Предоставляемые возможности	9
Заключение	12
Список литературы	13

1. Введение

В современном информационном обществе, где технологии становятся неотъемлемой частью повседневной жизни, автоматизация процессов играет важную роль в улучшении эффективности работы различных организаций. В контексте образовательных учреждений автоматизация процессов становится важным фактором, способствующим оптимизации деятельности учебных заведений. Она позволяет уменьшить затрачиваемые ресурсы на налаживание учебного процесса.

Важной частью процесса обучения является составление расписания занятий. На математико-механическом факультете СПбГУ данной работой занимается диспетчер. Несмотря на предпринятые попытки создания инструментов, способствующих оптимизации времени работы над расписанием [1] [2], на текущий момент в рамках математико-механического факультета все этапы данной деятельности, за исключением публикации готового расписания на информационном портале timetable [3], не являются автоматизированными.

В рамках учебной практики будет продолжена работа осеннего семестра, в которой были осуществлены моделирование предметной области составления расписания математико-механического факультета и её анализ. Данная учебная практика будет направлена на реализацию кода, необходимого для автоматизации процессов.

2. Постановка задачи

Цель данной работы — реализовать код для работы с пожеланиями преподавателей в рамках составления расписания занятий на основе материалов, полученных в осеннем семестре. Для достижения данной цели были поставлены следующие задачи.

1. Реализовать синтаксический анализатор педагогической нагрузки.
2. Реализовать синтаксический анализатор расписания занятий на семестр.
3. Реализовать валидацию расписания с учётом имеющихся пожеланий преподавателей.
4. Разработать пользовательский интерфейс для взаимодействия диспетчера с инструментом.

3. Обзор

3.1. Обзор существующих решений

Прежде всего были рассмотрены несколько самых популярных по упоминаниям в поисковых системах Google и Яндекс инструментов. Для поиска использовался запрос ”инструмент для автоматического составления расписания вуза” и аналогичные.

1. Экспресс-расписание колледж. [4]
2. 1С:Автоматизированное составление расписания. [5]
3. Система «АВТОРасписание» [6]
4. Расписание занятий: «Ректор-Колледж» [7]
5. Хронограф [8]

Многие из программ являются лишь вспомогательными для ручного составления и ведения расписания, что уже умеет инструмент ”ЕС Расписание”, установленный на удаленном столе всех диспетчеров СПбГУ. А если инструменты и умеют составлять расписание сами (например, «Ректор-Колледж»), то делают они это долго и не учитывают всей специфики расписания СПбГУ. Кроме того, четыре из пяти из вышеперечисленных программ платные.

Кроме того, под рассмотрение попала работа выпускника Санкт-Петербургского политехнического университета им. Петра Великого Виктора Хомякова. [9] Его математическая модель процесса довольно близка к той, что получилось составить в ходе обсуждения с диспетчером, однако, программный код готового продукта заточен под специфику его вуза, в частности, есть некоторые различия в организации курсов. В ходе обсуждения кода с автором, выяснилось, что его программа не очень подходит для расширения и добавления деталей, так как основной упор был сделан на оптимизацию времени работы алгоритма, а не на работу с данными. Однако идеи, примененные в его продукте могут оказаться полезными в дальнейшей разработке.

Кроме того, на кафедре системного программирования была предпринята попытка написания подобного алгоритма [1], однако работа направлена на оптимизацию скорости работы и не совсем учитывает специфику входных данных, например, алгоритм не может учитывать аудитории, в то время как пожелания на место проведения занятия являются довольно часто встречающимися. Кроме того, модель пожеланий преподавателей, составленная для данного инструмента, дает только возможность избегания окон, в то время как спектр возможных пожеланий в несколько раз шире [10].

4. Ход работы

Изначально было решено реализовать консольное приложение, которое может принимать на вход различные файлы с расширением .xlsx, анализировать их в зависимости от содержания таблиц и выводить результат.

4.1. Выбор инструментов

Для реализации проекта была выбрана платформа .NET, так как у кафедры системного программирования уже есть реализованный инструмент разбора учебных планов [2], который был написан с её использованием. Поэтому для облегчения интеграции инструментов было решено продолжить использование данной платформы, так как информация из учебных планов может быть полезна в будущем при расширении функциональности инструмента работы с расписанием [11].

Для реализации синтаксических анализаторов расписания и академической нагрузки, а также алгоритма валидации преподавателей был выбран язык F#, так как он является функциональным языком, подходящим для обработки и анализа данных. Важным аспектом выбора F# в данном проекте является возможность работы с коллекциями данных через встроенные функции обработки последовательностей, такие как `map`, `filter` или `fold`. Это упрощает написание кода для разбора сложных структур в расписаниях и нагрузках. Кроме того, F# предоставляет лаконичный синтаксис для функций высшего порядка и возможности сопоставления с образцом, что сокращает объем вспомогательного кода и позволяет сосредоточиться на основной логике при разработке алгоритма валидации пожеланий преподавателей.

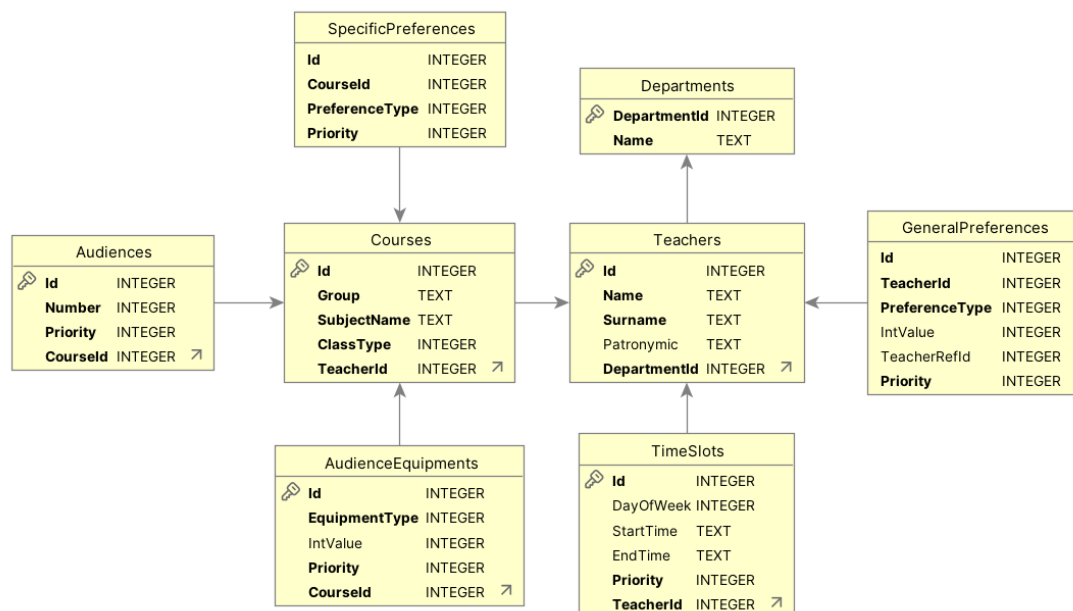
4.2. Пользовательский интерфейс

Работа с консольным приложением показалась диспетчеру неудобной. Поэтому было решено доработать инструмент, добавив графический пользовательский интерфейс.

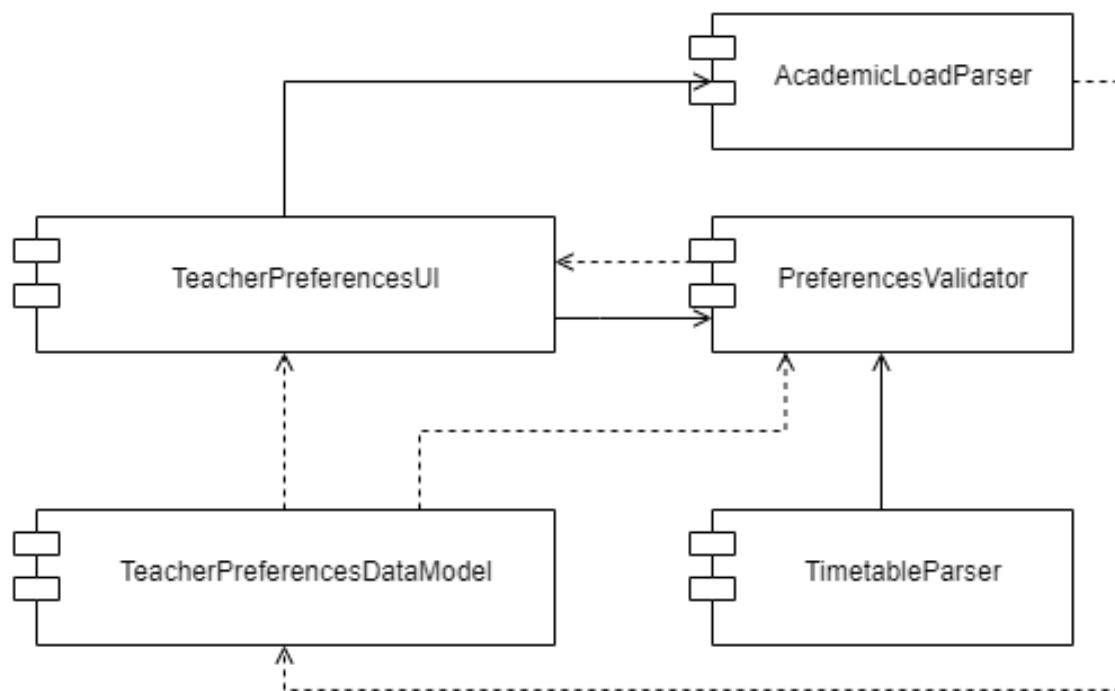
С учётом локального хранения данных, необходимых для работы инструмента, а также особенностей рабочего компьютера диспетчера, таких как операционная система Windows и ограниченный объём оперативной памяти, для разработки графического интерфейса был выбран язык C# в сочетании с Windows Presentation Foundation (WPF). Учитывая опыт работы автора с WPF, данная технология стала удобным выбором для разработки пользовательского интерфейса.

Для хранения данных о пожеланиях преподавателей и их дальнейшей валидации в ходе составления расписания было решено использовать Entity Framework в сочетании с SQLite. SQLite был выбран в связи с тем, что это лёгкая и встраиваемая база данных, которая подходит для локального хранения информации в рамках небольших приложений. Инструмент для работы с расписанием является таковым, так как в среднем количество групп равняется 80, количество преподавателей 400, а количество курсов близко к 1700 [11].

Структура базы данных повторяет формальную модель пожеланий преподавателей, которая была доработана в процессе тестирования:

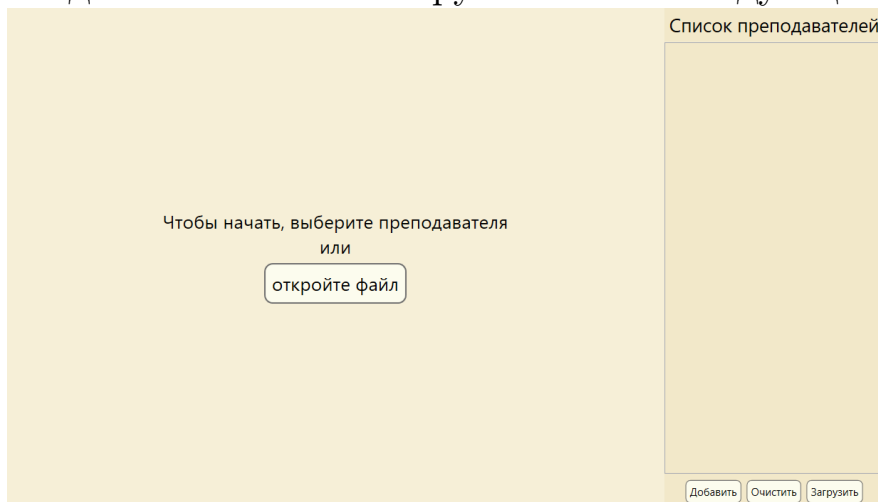


Итоговая архитектура приложения получилась следующей:



4.3. Предоставляемые возможности

На данный момент инструмент имеет следующие возможности:



1. Кнопка "загрузить" позволяет открыть файл педагогической нагрузки. С помощью синтаксического анализатора происходит заполнение базы данных преподавателям и курсами, которые они ведут.

2. Кнопка "добавить" позволяет добавить преподавателя вручную, если его по каким-то причинам не оказалось в файле педагогической нагрузки.

The form is titled "Введите данные преподавателя" (Enter teacher's data). It contains four input fields: "Фамилия" (Surname), "Имя" (Name), "Отчество" (Patronymic), and "Кафедра" (Department, which is a dropdown menu). At the bottom, there are two buttons: "Назад" (Back) and "Сохранить" (Save).

3. Выделение преподавателя в списке открывает меню редактирования его предпочтений.

The form is titled "Петров П. П." and is divided into two main sections. The left section, "Общие пожелания" (General preferences), has a sub-section "Пожелания по времени" (Preferences by time) with two rows of time slots. Each row has a day selector (currently "Понедельник"), start and end times, a priority dropdown (currently "Обязательно" and "Недопустимо" respectively), and a "Удалить" (Delete) button. Below these is a "Добавить пожелания по времени" (Add preferences by time) button. The right section, "Список преподавателей" (List of teachers), shows a list of teachers: "Иванов И. И. Кафедра небесной механики" and "Петров П. П. Кафедра астрономии". The second teacher is highlighted. At the bottom, there are buttons: "Добавить предпочтение" (Add preference), "Сохранить" (Save), "Удалить" (Delete), and "Отмена" (Cancel).

4. Кнопка "откройте файл" позволяет открыть файл расписания.
5. После открытия файла с расписанием и его синтаксического анализа выводится список ошибок, в которых указаны ячейки, которые не удалось разобрать.

Найден лист с неожиданным названием 'комп'. Он будет пропущен

Найден лист с неожиданным названием 'матмех'. Он будет пропущен

Преподаватель не найден для Русский как иностранный

Аудитория не найдена для Русский как иностранный

Аудитория не найдена для АНГЛИЙСКИЙ ЯЗЫК (пр. з.) Русский как иностранный

Преподаватель не найден для Программирование и вычислительный практикум (1 пгр)

Аудитория не найдена для АНГЛИЙСКИЙ ЯЗЫК (пр. з.) Русский как иностранный

Аудитория не найдена для АНГЛИЙСКИЙ ЯЗЫК (пр. з.) Русский как иностранный

Список преподавателей

Назад

ОК

Добавить

Очистить

Загрузить

6. Данные ошибки могут быть проигнорированы нажатием кнопки "ок". После этого интерфейс выводит список всех найденных несоответствий между расписанием и пожеланиями преподавателей.

Преподаватель Иванов И.И. не найден в расписании.

Преподаватель Петров П. П. должен иметь пары в диапазоне времени с 09:30 до 11:05 в понедельник, но они не найдены.

Преподаватель Петров П. П. не должен иметь пары в диапазоне времени с 17:00 до 18:45 в понедельник, но они найдены.

Список преподавателей

Иванов И. И.
Кафедра небесной механики

Петров П. П.
Кафедра астрономии

Назад

Добавить

Очистить

Загрузить

Заключение

В ходе выполнения данной работы были получены следующие результаты.

1. Реализован синтаксический анализатор педагогической нагрузки.
2. Реализован синтаксический анализатор расписания занятий на семестр.
3. Реализована валидация расписания с учётом имеющихся пожеланий преподавателей.
4. Разработан пользовательский интерфейс для взаимодействия диспетчера с инструментом.
5. Реализованный код протестирован и выложен на GitHub [12].

Список литературы

- [1] URL: <https://github.com/Azatic/schedule>.
- [2] URL: <https://github.com/spbu-se/spbu-curriculum-tool>.
- [3] URL: <https://timetable.spbu.ru/>.
- [4] URL: <https://pbprog.ru/catalog/all/376>.
- [5] URL: https://solutions.1c.ru/catalog/asp_spo.
- [6] URL: <https://www.mmis.ru/programs/avtor>.
- [7] URL: <https://rector.spb.ru/>.
- [8] URL: <https://sofrare.ru/windows/hronograf-30-master>.
- [9] URL: <https://elib.spbstu.ru/dl/3/2023/vr/vr23-3496.pdf/en/info>.
- [10] URL: https://docs.google.com/document/d/1rVlLm7ZiB8cfroR024W1k7b_yKP_JCa30A_ypdfSEuc/edit.
- [11] URL: <https://docs.google.com/document/d/1Iw4e8o9tacFwdSpCT0IW9iXo-rWTDXd0048xNbW1azo/edit?usp=sharing>.
- [12] URL: <https://github.com/sayumikko/Spbu-timetable-tool>.