**Exam-Scheduler**

**High Level Design**

**Submitted by**:

Roey Ashkenazy

Tal Gelber

Dor Bartov

Yosef Bader

Rephael Azoulay

Moisei Vainbaum

**Supervised by:**

Tanya Brokhman

**Table of Contents**

[Abstract 3](#_1fob9te)

[1. Introduction 4](#_2et92p0)

[1.1 General Background 4](#_1ksv4uv)

[1.2 Programming Environment 4](#_tyjcwt)

[2. Theoretical Background 4](#_3dy6vkm)

[2.1 The Problem 4](#_44sinio)

[2.1.1 The Solution 5](#_2jxsxqh)

[2.2 Our Responsibilities 5](#_z337ya)

[3. Basic System Functionalities 5](#_17dp8vu)

[4. Software Implementation 5](#_3rdcrjn)

[4.1 User Interface 5](#_3j2qqm3)

[4.2 Features 5](#_1y810tw)

[5. References 5](#_35nkun2)

# 

# Abstract

# Introduction

## General Background

The exam schedule of the Computer Science faculty in the Technion is currently built manually. This means the undergraduate secretariat schedules important university-wide courses and then the faculty’s secretariat has to fit in their particular courses with as few conflicts as possible.

This project aims to help the secretary in charge of this task build an exam schedule for the faculty in a computerized manner. This will potentially save human resources, as well as optimize the schedule based on different configurations.

## Programming Environment

Development of GUI and logic will be done in Java, with GUI using the JavaFX library, In the IneliJ Idea IDE.

Data for the application will be stored in local files.

# Theoretical Background

## The Problem

The problem depicted above entails building the exam schedule automatically, while addressing the following issues (among others):

* Some of the courses have been pre-scheduled by the Ullman secretariat and their allocation cannot be changed.
* Some courses belong to several faculties, and scheduling them might require taking theses faculties’ scheduled into account.
* Some courses of the Computer Science faculty are being taken at different semesters in different specializations within the faculty.
* There are days in which exams cannot take place.

Currently, the semesters’ representatives supply extra factors to consider, such as:

* Course popularity.
* Exams which require extra time to study for
* Courses that are usually taken together while not being in the catalog together.

An exam’s time in the day, or assigning rooms is not an issue for the faculty, and therefore will not be addressed.

### The Solution

* Our System will deal with a single semester, including both Moed A and Moed B.
* The GUI will allow for an intuitive way to input the data mentioned above, as well as official restrictions such as the beginning and ending date of each Moed.
* The system will then try to build an exam schedule which takes all this data into account, while optimizing the schedule. The definition of Optimal Solution with which we will work is an exam schedule that fulfills all the official restrictions and provides to most students sufficient amount of preparation time (3-5 days) before exam, while maximizing the preparation time whenever possible.

## Our Responsibilities

* Our project's scope is a single semester's exam period (Moed A and Moed B).
* We will only schedule Computer Science courses while attempting to allow the maximum time between exams to give the students time to prepare.
* Our system will allow an easy way to input the required data, as well as a way to output the data in different formats such as an excel spreadsheet or a txt file.
* We will aim to make the system future proof by allowing changes such as adding / removing courses, changing dates, etc.

# Basic System Functionalities

The list of recommended courses for a semester will come predefined with the application. A user can make changes in the list based on current faculty catalog.

The application will require the following input:

* Start and end dates for the two exam periods.
* courses relevant for the current schedules (which courses are given this semester, and have exams) - this input will be given by the user adding and removing courses from a list in the GUI.
* The predefined exam dates from the Ullman secretariat (txt file or manual input).
* Restrictions from related faculties (manual input).
* Groups of courses taken together (for each such group we should allow sufficient preparation time between exams).

The program will parse all the input to generate a database of courses and constraints. The user can then execute the scheduling algorithm to create the exam schedule. The output of the algorithm will be displayed on calendars, and it will be possible to modify it manually.

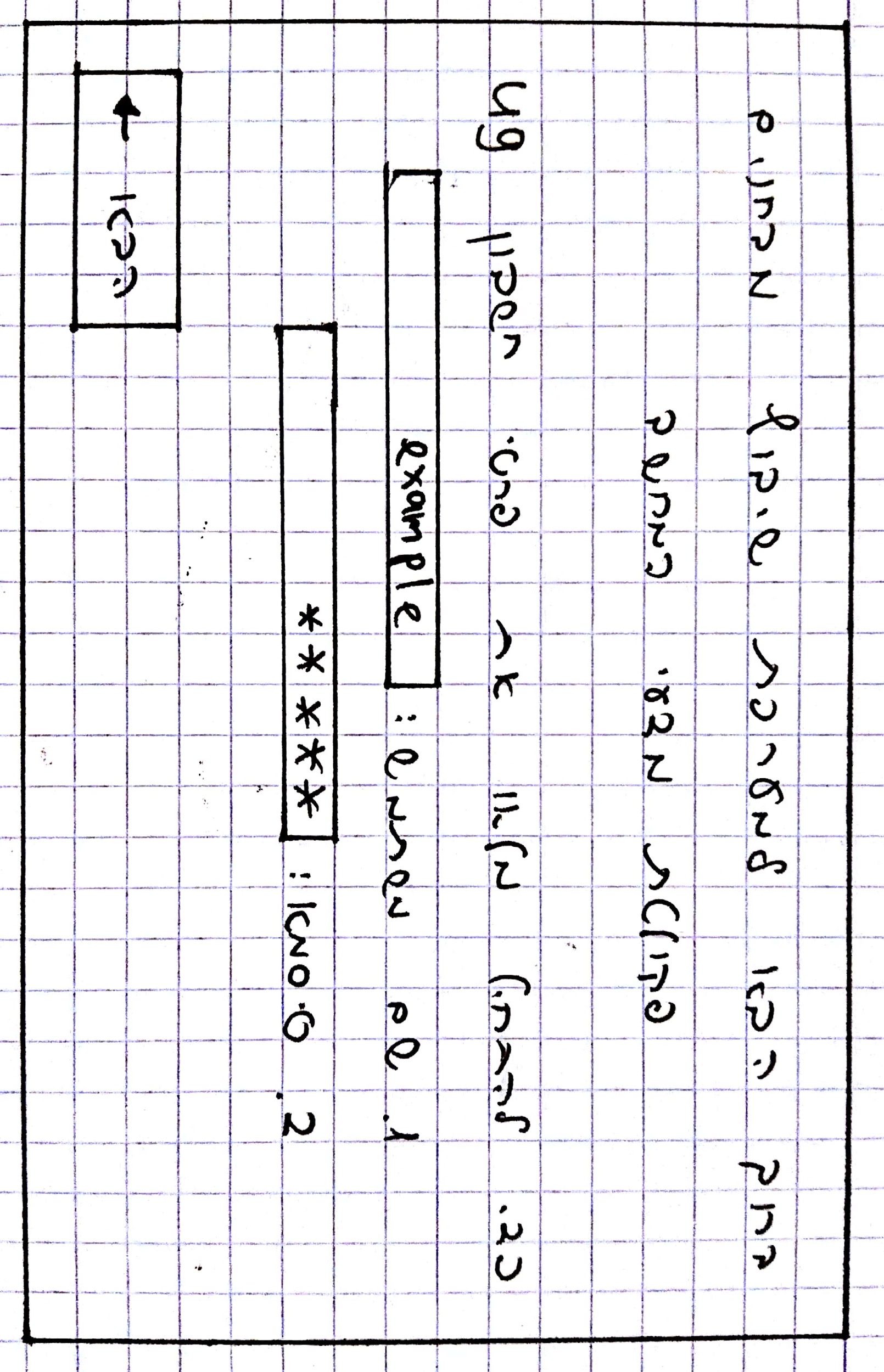
After the exam schedule is finalised, the application will allow a choise of format in which to output the results.

# Software Implementation

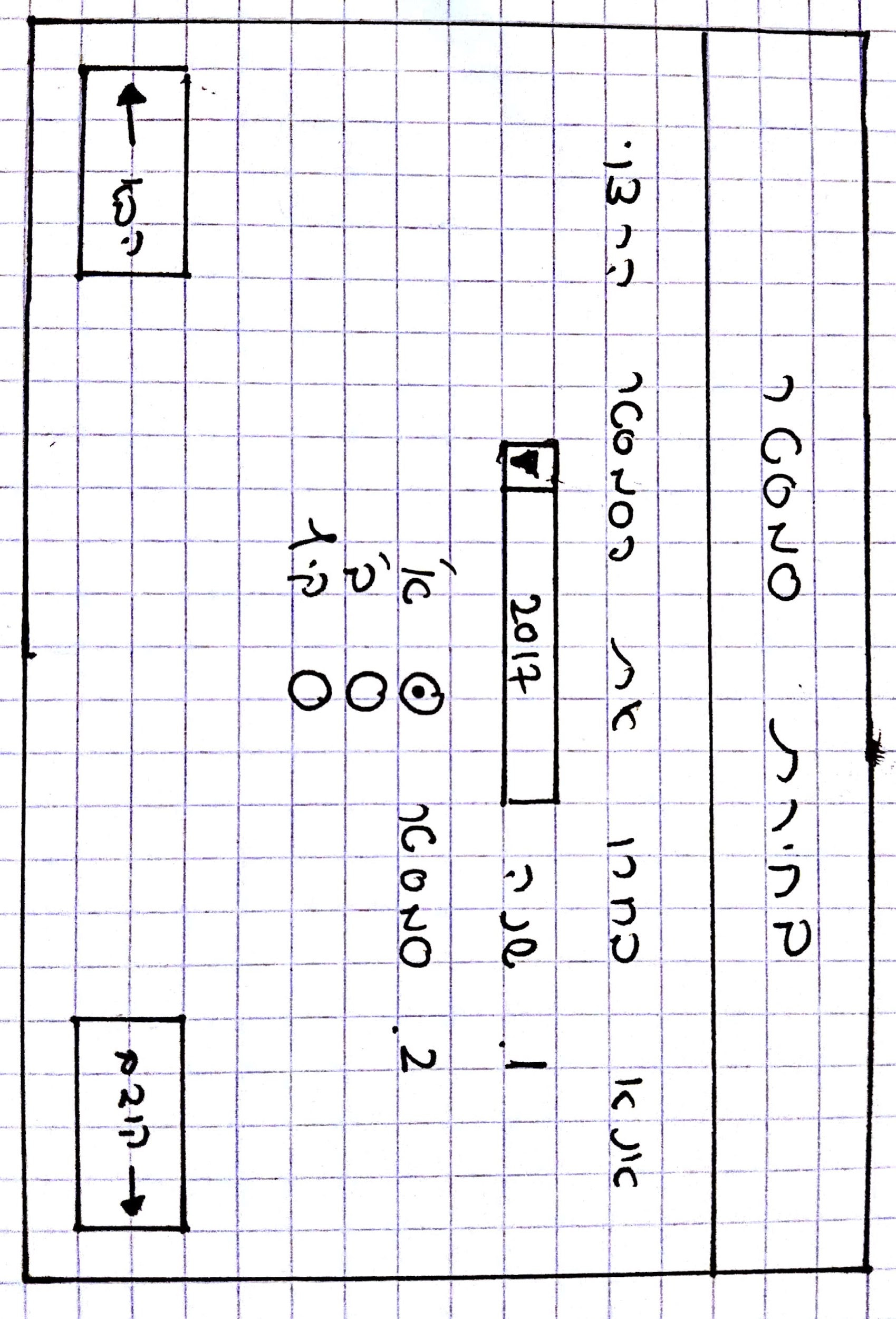
## 4.1 User Interface

The user interface will include:

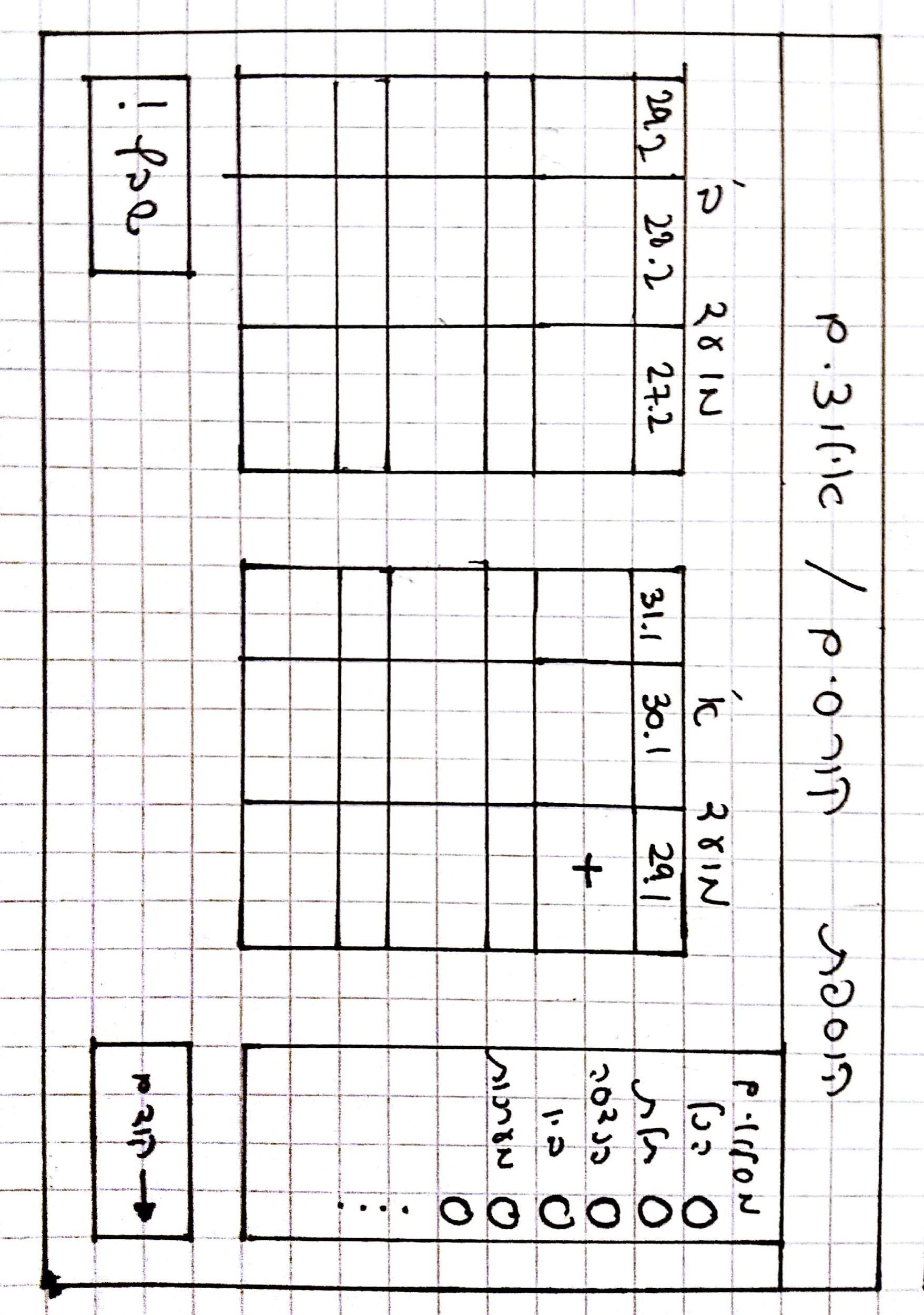
* Login window to be able to connect to ug servers



* A window for selecting the relevant semester



* A main window that will contain two calendars, one for each exam period



* + When the “+” is pressed a window will pop up for the user to add a constraint such as:
    - Ulman pre-scheduled exam.
    - Specific day that exams will not be scheduled in it.

Changes like those will appear on the calendars.

* + When the “שבץ” is pressed the scheduling algorithm will start working and the final scheduling will appear on the calendars.
* A window for the user to be able to add or remove courses from the chosen semester (default courses will be according to the catalog) or mark a course with no exams.
* Another button to upload the exam schedules to UG server.

## 4.2 Features

Our project will include the following features:

* We aim to make the flow of the program clear and intuitive.
* The application will allow for a comfortable way to input all of the required and optional input. This includes the user being able to define constraints on the scheduler, like a predefined date for a specific exam.
* The main algorithm of the program will aim at optimization of CS faculty exams scheduling. The focus will be on maximizing the time between two exams that are supposed to be taken in the same semester, according to the catalog.
* The algorithms’ result will be displayed in a calendar, making it easy to understand.
* The user will be able to make manual changes on the exam time table at any time after exam schedule generation (Add or remove a course, move an exam to another date etc).
* The application will allow for a choice of output formats (such as an excel spreadsheet or a txt file).

# 5. References

1. Mika Shapira

The CS faculty secretary currently in charge of scheduling the exam dates manually.