**Vision Document for “MUMScheduler“**

**An online Student/Course Registration system for MIU Computer Science department**

**Team members:**

*Ayoub lachhab 611939*

*Filmon Tesfalem 612345*

*Zebib Zegherghish 112234*

*Helen Teklemariam 111816*

*Anderson Arbelaez Arias 612333*

**1. Introduction**

Several years ago in the Computer Professional MS in CS program, there were three

entries per year and student entry numbers were 20-40 per entry. Often there was just one

elective class being offered per block and all students in an entry took the same classes in

the same sequence. Scheduling of classes and faculty was done with a relatively simple

Excel spreadsheet, and students were assigned to classes via a manual process.

As Compro has grown, we now offer 4 entries per year and there are often 100 – 130

students per entry. In some blocks, we may offer 8 or 9 elective classes, plus there are

often 3 FPP classes and 5 MPP classes offered per entry. There are several areas of

specialization for classes such as:

- Web Applications

- Data Science

- SW Design

- Networking

- Operating Systems

- Compilers

- Parallel Programming, etc.

Most faculties have one or two areas of specialization and a set of classes that they

would like to teach. In addition, they have preferences for what blocks they can teach.

Faculty needs to be able to enter their profile and be able to view their scheduled classes.

Compro students should be able to view the schedule and register for classes.

A few 500 level courses have 400 level **course prerequisites**, so the 400 level courses

should be offered for each entry in their first blocks on campus.

The 500 level classes should be provided for their later blocks on campus.

Most students take 4 elective blocks on campus.

Some U.S. resident students take 9 elective blocks on campus.

Some OPT students take 5 courses on campus.

MUMSched is a new software tool that will build a Compro schedule of classes with

faculty assigned to each class and will also offer a simple tool for students to register for

those classes.\*

(\*Note – the student registration part will be kept simple for our project. It is added for

the purposes of having a separate student register subsystem – to be explained in later.)

**2. Positioning**

**2.1 Problem Statement**

**This tool will be useful to any university that uses the block system for offering classes. With small modifications it could also be used for Semester based programs.**

|  |  |
| --- | --- |
| The problem of | *managing the Compro schedule and allowing students to*  *register for classes* |
| Affects | *administrators, faculty, and students* |
| the impact of which is | *scheduling is complex, must be manually maintained, and*  *changed frequently* |
| a successful solution would be | *one tool which builds a Compro schedule that integrates the*  *business rules for faculty availability and courses needed by*  *students per entry. This tool will provide a Database and a*  *user interface that is easy to use for faculty, staff, and*  *students.* |

**2.2 Product Position Statement**

|  |  |
| --- | --- |
| For | *Any University or College program* |
| Who | *Needs To manage their scheduling of classes* |
| The (product name) | *MUMScheduler* |
| That | *Automates generating a schedule from selected inputs* |
| Unlike | *It Gonne be An Excel spreadsheet which requires manual scheduling* |
| Our product | *Creates a database for the schedule and provides a UI that can be tuned for specific types of users.* |

**3. Stakeholder Descriptions**

**3.1 Stakeholder Summary**

****

**3.2 User Environment**

*It could Web Application or user Interface work on any device mobile phone or laptop.*

**4. Product Overview**

**4.1 Product Perspective**

*The MUMScheduler will allow to register online the courses each semester for each student.  
Once the student finish to register the course the system would verify if the selected courses satisfied  
the requirements and approve or disapprove the selected course. Each student will watch the progress  
of the course until complete it. Students may drop a class up to the end of the first week of the  
semester in which the class is being taught. It is an independent system; it means that it is not a  
component of another application.*

**4.2 Assumptions and Dependencies**

*We only provide a very simple student registration interface since the student registration system has already been developed and is deployed on the Infosys system here:*

<http://register.cs.mum.edu/register/>

**4.3 Needs and Features**























**4.4 Alternatives and Competition**

[*https://sakai.cs.miu.edu/*](https://sakai.cs.miu.edu/)

[*https://www.regpacks.com/*](https://www.regpacks.com/)

[*https://www.brushfire.com/*](https://www.brushfire.com/)

[*https://www.aventri.com/*](https://www.aventri.com/)

**5. Other Product Requirements**

*By the end of this course the final development of the project must be done, we have challenge our group work together for the first time we all try to learn and execute what we learn on this project*