Smart Schedular

An MIS 310 Project

Contents

[Executive Summary 3](#_Toc344288798)

[The Project Charter 3](#_Toc344288799)

[Project Purpose: 3](#_Toc344288800)

[Anticipation Completion: 3](#_Toc344288801)

[Approved Budget: 3](#_Toc344288802)

[Key Participants: 3](#_Toc344288803)

[The System Scope Document 4](#_Toc344288804)

[Problem Description: 4](#_Toc344288805)

[Anticipated Business Benefits: 4](#_Toc344288806)

[System Capabilities: 5](#_Toc344288807)

[Preliminary Alternatives 6](#_Toc344288808)

[Recommendation 6](#_Toc344288809)

[Context Diagram 7](#_Toc344288810)

[Project Plan (WBS) 8](#_Toc344288811)

[Predictive 8](#_Toc344288812)

[Adaptive 10](#_Toc344288813)

[Analysis Report 11](#_Toc344288814)

[Event table 11](#_Toc344288815)

[Subsystems and the use case for each subsystems 12](#_Toc344288816)

[Data Model 13](#_Toc344288817)

[Entity Relationship Diagram (ERD) 13](#_Toc344288818)

[Data Flow Diagram (DFD) Fragments 14](#_Toc344288819)

[Level 2 DFD Fragments 17](#_Toc344288820)

[System Requirements 23](#_Toc344288821)

[Functional Requirements 23](#_Toc344288822)

[Non- functional requirements 24](#_Toc344288823)

[Alternative solutions considered 24](#_Toc344288824)

[Summary 24](#_Toc344288825)

[Methods used for requirements 25](#_Toc344288826)

[Appendix 26](file:///E:\Project%20Requirements\The%20Project%20(Smart%20Schduler).docx#_Toc344288827)

# Executive Summary

This is a report to purpose a project that can help both advisors and students alike. This project will. This project reduces errors caused by students and advisors, reduces the time needed to schedule and register for classes, increases options of which classes the students can take, plans for which classes to take in upcoming semester/s, helps schools to add certain classes for a particular subject, and develops simple programs which students can take for the whole degree.

# The Project Charter

Project Name: Smart Scheduler

Project Purpose:The projects purpose is to design a system that can be added to a schools website, as well as being integrated into the schools date base of classes and previous semesters’ classes.

Anticipation Completion: Within 3 months of project initiation.

Approved Budget: 3000 KD

## Key Participants:

|  |  |  |
| --- | --- | --- |
| Participant | Position | Primary Responsibility |
| Anwar Alzalzalah | Project Manager | Manage the entire project |
| Ali Nawabdin | Director | Supervise project manager  Check status weekly  Serve on oversight committee |
| Anwar Alzalzalah | Director of Human Resources | Employ qualified personnel |
| Ali Nawabdin | Employee Supervisor | Manage Employees  Supervise Employees |
| Anwar Alzalzalah | Marketing Director | Offer schools the systems |
| Ali Nawabdin | Customer Service Director | Manage any Customer Service problems |

# The System Scope Document

## Problem Description:

Many students and advisors find it difficult to keep track of the credits, classes, and what the best route to take is. The program helps students and advisors alike, to manage their class schedules from start to finish. The system also recommends which classes to take as electives to help in the case of Minors and even Double Majors. This system improves how students schedule their classes and minimizes the risk of taking an irrelevant class.

## Anticipated Business Benefits:

The primary business benefit for “Smart Scheduler” is that it eases the job of a student and advisor alike. More immediate benefits include the following:

* Reduce errors caused by students and advisors.
* Reduces the time needed to schedule classes.
* Increases options of which classes the students can take.
* Plan for which classes to take in the following semester.
* Helps schools and universities to add a certain class for a particular subject.
* Develop simple programs for the whole degree.
* Easily check grades and GPA.
* Everything can be done through the systems web space.

## System Capabilities:

To obtain the business benefits listed previously, the system shall support the following capabilities:

* Calculate the number of credits needed to finish the major.
* Look at previous years classes offered to anticipate which classes will be offered.
* Tell the school how many students will want to enroll into a particular class.
* Show classes a student has completed and the classes needed for the major.
* Determines all pre-requisites to go on.
* Calculates how long the student needs to complete their program.
* Provide a description of the class.
* Shows grades.
* Shows GPA.
* Creates major sheet.
* Generate managerial report.
* Generate students’ semester report.
* Generate faculty report.
* CRUD capabilities allowing it to create, read, update, and delete, classes’ courses and such.

In the previous points students are referred to students and advisors alike.

# Preliminary Alternatives

The first preliminary alternative prior to the project completion, the school may be willing to thoroughly direct students to advisors. Advisors can manually access the system and correlate the student’s major sheet with the upcoming classes. The advisor should have knowledge of past experience with students attending those classes. The advisor should also be granted the right to register the students into classes. The advisor should also study and learn how to complete the major in the least amount of time, or the path that would be the lightest burden on the student themselves.

The second alternative is to have the registration department do all the advisors roles which are stated in the first preliminary alternative above.

The final alternative would be to create a temporary alternate department to take the advisors roles stated in the first preliminary alternative above.

# Recommendation

The school should create a system known as the “Smart Scheduler”, which will be stated in the following analysis. This Smart Scheduler will take all of the roles stated in the first preliminary alternatives stated in the previous section and more. This however, does not exclude the advisors role in helping the student to register or choose the classes they would like to register for. This will save much time and effort for the advisor, making their role as an advisor manageable.

# Context Diagram

CURD Student Database

Student Database

Access System

Student/ Advisor

Smart Scheduler System

Access System

CURD Class Database

Registration

Class Database

Generate Management report

Generate Faculty report

Read Major Database

Management

Faculty

Major Database

# Project Plan (WBS)

## Predictive

|  |
| --- |
| Tasks |
| -1 Project Planning |
| -1.1 Define the problem |
| 1.1.1 Meet the users |
| 1.1.2 Determine scope |
| 1.1.3 Write problem description |
| 1.1.4 Identify business benefits |
| 1.1.5 Identify system capabilities |
| 1.1.6 Develop context diagram |
| -1.2 Produce the project schedule |
| 1.2.1 Develop WBS |
| 1.2.2 Estimate durations |
| 1.2.3 Determine sequences |
| 1.2.4 Develop Gantt chart |
| -1.3 Confirm project feasibility |
| 1.3.1 Identify intangible cost/ benefits |
| 1.3.2 Estimate tangible benefits |
| 1.3.3 Estimate cost/ benefits |
| 1.3.4 Organizational feasibility |
| 1.3.5 Technical feasibility |
| 1.3.6 Evaluate resource availability |
| 1.3.7 Risk analysis |
| -1.4 Staff the project |
| 1.4.1 Develop resource plan |
| 1.4.2 Procure project team |
| 1.4.3 Produce employee liaison |
| 1.4.4 Conduct training |
| -1.5 Launch the project |
| 1.5.1 Make executive presentation |
| 1.5.2 Procure facilities |
| 1.5.3 Procure support resources |
| 1.5.4 Conduct kickoff meeting |

## Adaptive

# Analysis Report

## Event table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Event | Trigger | Source | Use case | Response | Destination |
| Check classes | Class inquiry | Student/ Advisor/ Registration | Look up class | Classes available for student | Student/ Advisor/ Registration |
| Register for class/es | Registration inquiry | Student/ Advisor/ Registration | Register class | Register student | Class database |
| Calculate no. of credits | Student progress inquiry | Student/ Advisor | Calculate credits | No. of credits | Student/ Advisor |
| Look up anticipated classes | Future semester class inquiry | Student/ Advisor/ Registration | Anticipate class | Anticipated classes | Student/ Advisor/ Registration |
| Look up anticipated students | No. of students for class inquiry | Registration | Anticipate students | Anticipated students | Registration |
| Show the completed classes | Completed classes inquiry | Student/ Advisor | Completed classes | Completed classes | Student/ Advisor |
| Show needed classes left for major | Classes left until graduation | Student/ Advisor | Needed classes | Classes left for student | Student/ Advisor |
| Check prerequisites for a class | Prerequisite inquiry | Student/ Advisor | Prerequisites | Prerequisites for class | Student/ Advisor |
| Check length of time till graduation | Length until graduation inquiry | Student/ Advisor | Length for student | Length of time in semesters | Student/ Advisor |
| Show grades | Class grades inquiry | Student/ Advisor | Grades | Grades for each class per each semester | Student/ Advisor |
| Show GPA | GPA inquiry | Student/ Advisor/ Registration | Calculate GPA | GPA for each semester and/ or current GPA | Student/ Advisor/ Registration |
| Show major sheet | Major sheet inquiry | Student/ Advisor | Major sheet | Student information | Student/ Advisor |
| Generate managerial report | Report inquiry | Management | Managerial report | Collective data and information | Management |
| Generate student report | Report inquiry | Student/ Advisor | Student report | Students report per semester/ total | Student/ Advisor |
| Generate faculty report | Report inquiry | Faculty | Faculty report | Collective faculty report | Faculty |

## Subsystems and the use case for each subsystems

Student Subsystem

* Look up class
* Register class
* Grades
* Calculate GPA
* Major sheet

Class Subsystem

1. Prerequisites
2. Completed classes
3. Needed classes
4. Anticipate class
5. Anticipate student

Smart Subsystem

* Length for student
* Managerial Report
* Student Report
* Faculty Report

1. Calculate credits

## Data Model

### Entity Relationship Diagram (ERD)

Faculty

Name

Faculty ID

Department

Major

Adviser Status

Office Number

Classes

Student Data

Name

ID

Major

Complete Classes

Number of Classes

Number of Credit

Grades

GPA

Advisor

Name

Faculty ID

Department

Major

Student

Name

ID

Major

Gender

Grades

Class

Location

Time Department

Class ID

Faculty

Number of Student

Major Department

Head Department

Name

Majors

Major sheet

### Data Flow Diagram (DFD) Fragments

#### Student Subsystem

1

Major Data

Student Data

Registration

Student

Class

Student

C

#### Class Subsystem

Major Data

Registration

Student

Class

Class

2

Class Data

Student Data

#### Smart Subsystem

Smart

Major Data

Management

Student

Faculty

3

Class Data

Student Data

### Level 2 DFD Fragments

#### Student Subsystem

B

C

D

E

F

G

H

A

N

M

L

K

J

I

Class Data

Grades

Calculate GPA

Major Sheet

Look Up Class

Register Class

1.5

1.4

1.3

1.2

1.1

Student Data

Student

Class

Registration

Major Data

#### Student Subsystem Index

1. Register Student
2. Put Student Name
3. Get Class Data
4. Look for class to register
5. Look for class
6. Look at grades
7. Look at GPA
8. Look at GPA
9. Search for pre-requisites
10. Pull up class info.
11. Pull up student info
12. Pull up student info.
13. Pull up student info.
14. Pull up student info

#### Class Subsystem

P

O

N

M

L

K

J

I

H

G

F

Major Data

Student Data

Class Data

Class Data

E

D

C

B

A

Student

Registration

Needed Classes

Anticipate Class

Anticipate Student

Completed Classes

Pre-requisites

2.5

2.4

2.3

2.2

2.1

Class

#### Class Subsystem Index

1. Look up class
2. Pull up completed classes
3. Pull up needed classes
4. Pull up classes anticipated
5. Pull up No. of Students Anticipated
6. Pull up class data
7. Pull up major data
8. Pull up major data
9. Pull up student data
10. Pull up student data
11. Pull up class data
12. Pull up class data
13. Pull up major data
14. Pull up major data
15. Pull up student data
16. Pull up student data

#### Smart subsystem

A

B

C

D

E

P

M

L

O

N

K

J

I

H

G

F

Student Data

Major Data

Class Data

Management

Student

Faculty

Student Report

Faculty Report

Management Report

Length for student

Calculate Credits

3.5

3.4

3.3

3.2

3.1

#### Smart Subsystem Index

1. Pull up credits completed
2. Pull up length until completion
3. Pull up student report
4. Pull up faculty report
5. Pull up management report
6. Pull up student data
7. Pull up student data
8. Pull up student data
9. Pull up major data
10. Pull up major data
11. Pull up major data
12. Pull up class data
13. Pull up class data
14. Pull up student data
15. Pull up major data
16. Pull up class data

# System Requirements

## Functional Requirements

There are several requirements that the smart scheduler should be able to do as mentioned above. The smart scheduler should be able to do the following corresponding with its capabilities:

* Calculate the number of credits needed to finish the major. With this capability, the system should take the students data and their major’s data, then calculate the number of credits they have finished corresponding with their major.
* Look at previous years classes offered to anticipate which classes will be offered. With this capability, the system should look at previous classes that were offered in previous years of that semester and anticipate if that class is offered, unless otherwise stated.
* Tell the school how many students will want to enroll into a particular class. With this capability the system should pull up all the students they would anticipate to enroll in the class by calculating if they need the class or not.
* Show classes a student has completed and the classes needed for the major. With this capability the system should pull up student data and show their completed classes with the grades they have gotten, as well as the classes that are left to complete their major.
* Determines all pre-requisites to go on. With this capability the system should look at all major pre-requisites that the student needs to take to graduate as soon as possible.
* Calculates how long the student needs to complete their program. With this capability the system should pull up the student data and the major data as well as the anticipated classes to determine what the fastest route to graduation is.
* Provide a description of the class. With this capability the system should provide a description of the class when they are viewing classes.
* Shows grades. With this capability the system should pull up the student’s data and shows all their grades for each class by semester, if stated otherwise, the system has a set of function such as, alphabetical.
* Shows GPA. With this capability the system should pull up the students data and calculate the MGPA (Major’s Grade Point Average) as well as the their Grade Point Average (GPA).
* Creates major sheet. With this capability the system should pull up the student’s data and create a major sheet by filling in all of the students data, such as year enrolled, grades, and all the class details for their major.
* Generate managerial report. With this capability the system should pull up all of the data connected to the system to generate a report for the management department.
* Generate students’ semester report. With this capability the system should be able to pull up all of the students data and generates a report.
* Generate faculty report. With this capability the system should pull up the faculty’s information and generates a report.
* CRUD capabilities allowing it to create, read, update, and delete, classes’ courses and such. With this capability the system should be able to create, read, update, and delete all classes and courses involved in the school.

## Non- functional requirements

The non- functional requirements are quite simple. The system is accessible through the internet and hence, it should have its own website. The system should be implanted into a server connecting all its required databases to that server. The system should also have great processing power to do all of the capabilities mentioned in its functional requirements. The system should also have maintenance that supervises its capabilities at least on a weekly basis. Due to the fact that this would be a critical element in a university, the previous point should be taken under extreme consideration.

# Alternative solutions considered

There are no alternative solutions for this system because the system also helps with time consumption. Any other alternative will use up much time or would create a more complicated system. However, if alternatives want to be taken, only the alternatives mentioned in the alternative section exist.

# Summary

This is a report/ proposal for the smart scheduling project. This system has several capabilities that would ease students, advisors, faculty, registration, and management’s life if implemented. This system can save a great deal of time for all its stakeholders. This report cover all its considerations and stakeholder, how it should work from point A to point B, and all its capabilities. This report also covers the systems context diagram, Work Breakdown Structure, and recommendations. This system is highly recommended for not only GUST as a university, but also for universities all around the world.

# Methods used for requirements

The requirements collected for this report and its capabilities were collected from several students a professor as well as an advisor. As being students ourselves, most of the ideas were suggested from each other. However, no surveys were conducted. The professors gave many suggestions, but not many could have been implemented into the system. The student however contributed many ideas and requirements to the system, some were chosen, and some were deemed irrelevant.

# Appendix