

CIS 422 Project 2: JTAS (Jaqua Tutoring Appointment Scheduler) SRS

David Han (dh), Kassandra Morando (km), Kelly Schombert (ks), Brianna Vago (bv), and Mert
Yapucuoğlu (my)
February 12, 2022 – v1.0

Table of Contents

- 1. SRS Revision History**
- 2. The Concept of Operations**
 - 2.1. Current Systems**
 - 2.2. Justification for a New System**
 - 2.3. Operational Features of the New System**
 - 2.4. User Classes and Modes of Operation**
 - 2.5. Use Cases**
- 3. Specific Requirements**
 - 3.1. Inputs and Outputs**
 - 3.2. Functions**
 - 3.3. Usability Requirements**
 - 3.4. Performance Requirements**
 - 3.5. Building Requirements**
 - 3.6. Software System Attributes**
- 4. References**
- 5. Acknowledgements**

1. SRS Revision History

Date	Author	Description
2-12-2022	bv	Created the initial document
2-13-2022	bv	Did the Concept of Operations and wrote down some basic requirements
2-14-2022	ks	Did the Functions
2-15-2022	dh	Did input and output
2-16-2022	ks	Edited Functions
3-3-2022	bv	Modified the requirements and table of contents
3-4-2022	bv	Edited the Operational Features of the New System, Current system, and Justification for New System
3-5-2022	bv	Edited the Functions and Inputs and Outputs section

2. The Concept of Operations

2.1. Current Systems

On the University of Oregon campus there is a student athlete tutoring building called the John E. Jaqua Academic Center for Student Athletes (Jaqua). At this location they need to schedule tutoring for each student athlete depending on their gpa and year. The problem with that scheduling process is that they need to schedule each tutoring session by making sure the student and the tutor can both make it at a specific time. There are quite a few scheduling systems out there that do basic scheduling, but it is quite difficult to find one that can create schedules for this specific situation.

The current system the Jaqua is using is a web based platform that was created for scheduling the tutoring sessions. The scheduling process is done manually and then they input it into a system to display it in a calendar format. It then can create a calendar format export and they are able to send it out to athletes by emailing their individual schedule links out. This site has a schedule with all their classes, their tutoring schedules, and their normal practice schedule. However, this schedule does not take students who have to do a different practice time due to class conflicts. This makes the schedule less usable for the athletes' daily lives. The tutors have access to their full schedules by going on their site and pressing the display option. They have all their appointments for the day displayed on their login home page. The tutors also have access to all the athletes and all the other tutors schedules on their main check in screen.

2.2. Justification for a New System

A new system will make sure that all the tutoring hour rules of the school and the NCAA are being followed. The NCAA requirements include:

- Freshman and transfers need to have 8 hours of tutoring.
- People with a gpa below a 2.29 must have 8 hours of tutoring.
- People with a gpa between 2.30 and 2.59 must be assigned 6 hours of tutoring.
- People who have a gpa between 2.60 and 2.99 must have 4 hours of mandatory tutoring.

It will also allow for more privacy for the student athlete and tutors because all of the tutors should not have access to every person's daily schedule. Some student athletes have different practice times due to class conflicts with the normal practice times. The current system does not take that into account when making and displaying the schedules. This can make the schedules being displayed incorrect because of the fact that a tutoring appointment could be scheduled for them during that time. The new schedule will take each individual athlete's practice schedule into consideration to allow for less errors to occur in that regard.

2.3. Operational Features of the New System

This new system will have an input for the user to input two files. One of the files will include the required information of each athlete and the other will include the required information of each tutor. Both of those files will include the hours they are unavailable along with other requirements. The system will then take both of those files and find the optimal full schedule to give to the user. It will also specify if it meets all the NCAA and school tutoring requirements. It will also be able to save individual tutor or athlete schedules so that the user can send them out.

2.4. User Classes and Modes of Operation

This system is being made for the employee that is responsible for scheduling all the tutoring sessions at the jequa. It will allow this employee to have an easier time doing their job due the many factors involved in the creation of the schedules. This system will have three different modes of operation that the employee will use:

1. The employee can use this system to create a full completed schedule which includes:
 - All the appointments that are scheduled for the week with the classrooms that are being used by each one.
2. The employee can use this system to get a desired athlete's individual schedule by typing their name into the search bar.
3. The employee can type a tutors name into the search bar to get a desired tutors individual schedule.

All these files will be exported in the form of a csv file which will be able to be viewed by the employee or sent to the tutor or athlete.

2.5. Use Cases

Use Case: Get a full schedule to show at a faculty meeting.

Description: This use case describes how to create the full schedule that includes all the tutoring appointments at every time on each day of the week.

Actors: An employee.

Preconditions:

1. The employee has access to a computer that has the necessary files downloaded. This can be done by following the installation instructions to make the scheduler work.
2. The employee will have the tutor information file on their computer in the proper formatting. This file needs to be a csv file.
3. The employee will have the athlete information csv file in the proper formatting on their computer.

Steps to Complete the Task:

1. The employee starts up the scheduler system on their computer by running it on their terminal.
2. They will click on the “File” button at the top left corner of the screen and then click the “Import” button.
3. A file finder window will open on the employees screen. Once it opens, navigate to their athlete file and select it.
4. Click the “Open” button at the bottom of that screen to import the athlete file.
5. A second file finder window will open. The employee should then navigate to their tutor file and select it.
6. To import the tutor file click the “Open” button at the bottom of that screen.
7. Run the scheduling process by clicking the “Create Schedule” button.
9. Look for the folder called “Schedules” folder for the .csv file that will have the full schedule written on it. It will be called “schedule.csv”.
10. Open the file.
11. The file will have each day of the week in a separate column and the rows will be organized by time. Each of the rows will have an appointment that is scheduled for that time and the columns split those up by day An example of this is shown here:

Time	Monday	Tuesday
8	Pinto, Chanaya ANTH274 with: Charlie, Early 201	Ellis, Scott ANTH145 with: Carleigh, Ocon 201
8	Wong, Nicholas HPHY323 with: Cooper, Gardner 202	Barbieri, Alice WR122 with: Teresa, David 202
8	Crocker, Isaah/Cromwick, Josiah ASL102 with: Jen, Gerould 203	Clayton, Ella Advising with: Dietrich, Moore 203
8	Jeffers, Jaylan ART101 with: Carleigh, Ocon 204	Cleveland, Bella ACTG211 with: Abigail, Smith 204
8	Doyle, Eric EC201 with: Kellie, McSween 205	Avina, Cassidy EDST225 with: Katelyn, Howery 205
8	Barbieri, Alice CH221 with: Ally, Cook 206	Moriyama, Yuki GEOG342 with: Morgan, Scafdi 206
8	Zajca, Mara MATH241 with: Sabrina, Schimscheimer 207	Gravante, Madeline SOC370 with: Nick, Lougee 207
8	Afaese, Maceal Advising with: Jennifer, Jackson 208	Johnson, Isaac SOC207 with: Cameron, Hartquist 208
8	Ayon, Isaac ENG110M with: Leilyn, Miles 209	Hosendove, Taylor J397 with: Audrey, Kalman 209
8	Tyus, Ella MATH111 with: Erin, Speltz 210	Lokar, Madi MATH241 with: Eli, Molloy 210
8	McClellan, Daley SOC204 with: Juliana, Weinerth 211	Nakato, Kohana EC202 with: Dominic, Vaccher 211

Postconditions: The file is ready to be shown at a faculty meeting and is ready to be used. No further action by the employee is needed.

Use Case: Scott Ellis wants to get their tutoring schedule emailed to them.

Description: This use case describes how an employee can get an athlete's individual tutoring schedule so that they can email it to the athlete. As well as how to read that file

Actors: An employee. A student athlete named Scott Ellis.

Preconditions:

1. The employee has access to a computer that has the necessary files downloaded. This can be done by following the installation instructions to make the scheduler work.
2. The employee has run the program and created the full schedule as described in the use case "Get a full schedule to show at a faculty meeting".
3. The employee has Scott Ellis's email on their computer.
4. Scott Ellis has access to a computer or phone with their email on it.

Steps to Complete the Task:

1. The employee opens their computer and starts up the program using the terminal.
2. On the employee's screen there should be a text box. Click on the text box.
3. The employee can then type whatever athlete name they would like into the text box and hit enter. In this case they would type Scott Ellis into the box.
4. Navigate to the "Schedules" folder on the employees computer. The employee should then find a file called "mySchedule.csv".
5. The employee will then take this file and attach it to an email and send it to Scott Ellis.
6. Scott Ellis can then open the file on his computer and it will have just his tutoring appointment schedule inside of it. An example of this file text is shown:

The screenshot shows a Google Sheets interface with a spreadsheet titled "Scott_Ellis". The spreadsheet has a single sheet named "Sheet 1". The visible data is a table with the following structure:

Time	Monday	Tuesday	Wednesday	Thursday	Friday
8	Scott,Ellis ANTH145 with:Carleigh,Ocon 201	Scott,Ellis ANTH145 with:Carleigh,Ocon 201			

Only the times that Scott Ellis has tutoring appointments will be displayed on this schedule file.

Postconditions: The file is ready for use and requires nothing else from the employee or student athlete.

Use Case: Show a tutor their schedule and send it to them.

Description: This use case describes how to show a tutor their schedule and then send it to them.

Actors: An employee.

Preconditions:

1. The employee has access to a computer that has the necessary files downloaded. This can be done by following the installation instructions to make the scheduler work.
2. The employee has run the program and created the full schedule as described in the use case "Get a full schedule to show at a faculty meeting".
3. The employee has the tutor's email on their computer.

Steps to Complete the Task:

1. The employee opens their computer as well as starts up the program using the terminal.
2. The employee should then see a text box that they need to click on.
3. Once that is done the employee can then type the tutors name in that text box and hit enter.
4. The employee will then navigate to the "Schedules" folder on their computer. The employee should then find a file called "mySchedule.py".
5. The employee can then open that file to show the tutor on their computer. The schedule should be displayed as shown:

Kelly_Schombert

100% Zoom

Formula Table Chart Text Shape Media Comment

Sheet 1

A B C D E F

Table 1

Time	Monday	Tuesday	Wednesday	Thursday	Friday
10		Brylie, Hoover MATH111 with: Kelly, Schombert zoom			
11	Timon, Davis MATH111 with: Kelly, Schombert 203				
14		Abby, Mulvey MATH111 with: Kelly, Schombert 307	HsinYu, Lu MATH111 with: Kelly, Schombert 308		
16			Ella, Tyus MATH111 with: Kelly, Schombert 205		
19		Jade, Bernal MATH111 with: Kelly, Schombert 302			

Only the times that the tutor has appointments will be displayed on this schedule file.

- The employee can then take this file and attach it to an email and send it to the tutor for their use.

Postconditions: The file is ready for use and needs no further work from the employee or the tutor.

3. Specific Requirements

3.1. Inputs and Outputs

Input: Student Athlete File

Description:

A .csv file containing the relevant data for all qualifying student athletes. Data contained within the student athlete file includes the name, UO ID, Year, GPA, number of required tutoring hours, required subjects, and time availability of each athlete. The data will be used by the program to build student tutoring schedules that accommodate their required needs and time restrictions of both tutor and athletic parties.

Source of Input:

The user will provide the input to the program.

Valid Range of Input:

A .csv file will constitute a valid file input.

Units of Measurement:

Not applicable.

Data Format:

Each line of the .csv file must be formatted as the following:

<First Name>, <Last Name>, <UO ID>, <GPA>, <Year>, <# of Hours Wanted>,
<Subjects>, <Available Times>

An exception lies in the first line of the file, where the headers of each column will exist in order to provide context to the relevant columns of information.

Input: Tutor File

Description:

A .csv file containing the relevant data for all qualifying student tutors. Data contained within the tutor file includes the name, UO ID, maximum number of work hours, specialized subjects, and time availability of each tutor. The data will be used by the program to build student tutoring schedules that accommodate the required needs and time restrictions of both tutor and athletic parties.

Source of Input:

The user will provide the input to the program.

Valid Range of Input:

A .csv file will constitute a valid file input.

Units of Measurement:

Not applicable.

Data Format:

Each line of the .csv file must be formatted as the following:

<First Name>, <Last Name>, <UO ID>, <# of Hours Wanted to Work>,
<Subjects>, <Available Times>

An exception lies in the first line of the file, where the headers of each column will exist in order to provide context to the relevant columns of information.

Output: Full Schedule File

Description:

A .csv file containing the relevant data for all appointments scheduled between student athlete and tutor. Data contained within the file includes the name of the

students, subject being tutored, name of the tutor, and classroom being used. This data will be used to show the complete schedule of appointments being done in the Jaqua.

Source of Output:

The program will return the output to a folder called Schedules.

Valid Range of Input:

A .csv file will constitute a valid file input.

Units of Measurement:

Not applicable.

Data Format:

The first line will contain the names of the days of the week in, leading with a blank cell input to denote the time column. The headers will be ordered as so:

◇, Monday, Tuesday, Wednesday, Thursday, Friday

The following lines will all lead with a number representing the beginning of an hourly period in the day, denoted in military time. The leading time number will be followed by a tutoring appointment that begins at that designated time.

Output: Individual Schedule File

Description:

A .csv file containing the relevant data for an individual person's appointments schedule. Data contained within the file includes the name of the students, subject being tutored, name of the tutor, and classroom being used. This data will be used to show a student or tutor the schedule of appointments they are assigned in the Jaqua.

Source of Output:

The program will return the output to a folder called Schedules.

Valid Range of Input:

A .csv file will constitute a valid file input.

Units of Measurement:

Not applicable.

Data Format:

The first line will contain the names of the days of the week in, leading with a blank cell input to denote the time column. The headers will be ordered as so:

<>, Monday, Tuesday, Wednesday, Thursday, Friday

The following lines will all lead with a number representing the beginning of an hourly period in the day, denoted in military time. The leading time number will be followed by a tutoring appointment that begins at that designated time.

Output: Appointments File

Description:

A .txt file containing the relevant data for all appointments that have been scheduled. Data contained within the appointments file includes the name of the students, the time of the appointment, the day, the subject being taught, the name of the tutor, and the room being used. The data will be used by the program to build individual tutoring schedules.

Source of Output:

The program will return the output to a folder called Schedules.

Valid Range of Input:

A .txt file will constitute a valid file input.

Units of Measurement:

Not applicable.

Data Format:

Each line of the file will contain the information for a single appointment. The file will be ordered as so:

<Athlete Name> <Time> <Day> <Subject> <Tutor Name> <Room #>

Each of the items will be divided with a space.

Output: Error Log File

Description:

A .txt file containing any errors that the system caught during the running process. Information contained within this file will have the file the error is from as well as the reason for the error. This data will be used for the user to understand where the error is coming from.

Source of Output:

The program will return in the program folder.

Valid Range of Input:

A .txt file will constitute a valid file input.

Units of Measurement:

Not applicable.

Data Format:

Each error found in the file will have a date on the first line and then below it will be all the errors found on that run of the code. An example of an error that can happen if the GPA in the athlete file is a “5” would be:

2022-03-05

GPA can't below 0 or above 4. At line 3 of athlete file

Each error will have its own line after the date.

3.2. Functions

The system uses specific procedures utilizing functions from different components to provide needed functionalities. To collect the needed information from files, the system has two options:

1. Check to see if there is an existing save file:
 1. The module ScheduleSystem will call the function readSave() to check to see if there is an existing appointments .txt file.
 2. If the file is found then the system will use loadSave() to load information to make a schedule.
 3. If there is no file found then the files will need to be imported
2. Import files:
 1. ManagerInterface calls signalSchedule() to get the file paths.
 2. ScheduleSystem calls readFiles() to read the data.

Once the files have been imported, the system will make the schedule. There are two different ways that files will be exported from the system:

1. Files will be saved automatically by the ScheduleSystem calling saveFiles() from fileIO. That function will call writeSave() to create two different files:
 1. A .txt file to be loaded for the next run.
 2. A .csv file containing all of the scheduled appointments to be viewed.
2. Files that will be exported if one of the following conditions is met:

1. If a name is given to the system:
 1. ManagerInterface calls generateIndividual() to signal ScheduleSystem.
 2. ScheduleSystem calls exportIndividual().
 3. That function calls individualSchedule() from fileIO. This will generate an individual schedule in a .csv file.
2. If an error is found while running the system:
 1. When reading the code, readFiles() returns an error log.
 2. ScheduleSystem calls createErrorReport() from fileIO. This will create and write an error log .txt file.

3.3. Usability Requirements

Must Have:

- If a student has above a 3.0 gpa the system will take the amount of hours they imputed.
- If a student has between a 2.6 and a 2.99 gpa the system will assign them 4 hours of tutoring.
- If a student has between a 2.3 and a 2.59 gpa the system will assign them 6 hours of tutoring.
- If a student has between a 2.29 gpa or below the system will assign them 8 hours of tutoring.
- If a student is a freshman or transfer the system will assign them 8 hours of tutoring.
- If a student asks for more than 8 hours of tutoring then the system will only assign them 8 hours.
- If a student asks for no hours of tutoring and they don't need mandatory tutoring then the system will skip them.
- Must give each appointment a classroom for the session to meet at.
- Must be able to generate a schedule with less than 10 clicks.
- Students should only have access to their own schedules and not anyone else's.

Should Have:

- If none of the schedules hit all of the mandatory rules, then the system will say that.

Could Have:

- Be able to change the classrooms based on availability or closures.

3.4. Performance Requirements

Must Have:

- Runs scheduler 100 times every time the full schedule is being built.
- Be able to make an optimal schedule in less than five minutes.

Should Have:

- All schedules should start making appointments for a different athlete each time the schedules are built.
- All data saved onto the main schedule should not be lost or overwritten by the system or lost due to the system.

3.5. Building Requirements

Must Have:

- Must work on MacOS 11.2.3.
- The code is written in Python 3 and can use the standard Python library.
- The code is written using pandas.
- All files being imported into the system are .csv files.
- All readable files being exported from the system are .csv files.
- Score is kept on each schedule and the most optimal version is exported.
- The tutor file being imported will have each tutor's name, the classes they teach, the hours they are available, and the hours they are going to work.
- The student file being imported will have each athlete's name, the classes they are in, the year they are in, the hours that they are available, their gpa, and the hours of tutoring they want.
- The system must be able to accommodate any size group of athletes and tutors.

Should Have:

- All schedules exported in a .csv file will have the days in the columns and the times on the rows.
- There will be two priority queues, one will be required and the other will be optional.
- A whole number will be added to the score when someone from required is scheduled and a decimal number will be added when someone from optional is scheduled.
- If a person types a name into the display they will get a file exported with that person's individual tutoring schedule in a .csv file.

Could Have:

- There could be a preferred time for a tutoring session to be scheduled which will affect the scheduler.
- It could display the times of the upcoming tutoring sessions on the main display along with the tutor and athlete's name.
- Be able to open schedules in a web browser with a login.
- Could export 3 different schedules: a tutor schedule, an athlete schedule, and a combined schedule.
- A way to send the athlete's schedule to their advisors to coordinate what

needs to be done.

- Export the top three scoring schedules.

3.6. Software System Attributes

Should Have:

- Reusability
- Reliability
- Maintainability
- Privacy

4. References

SRS Template by Anthony Hornoff

<https://classes.cs.uoregon.edu/22W/cis422/Handouts/Template-SRS.pdf>

5. Acknowledgements

The formatting of this document was based on a Software Requirement Specification template provided by Professor Anthony Hornof.