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

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. Project Plan Revision History

Date	Author	Description
2-12-2022	ks	Created the initial document and wrote previously discussed and recorded elements of the project plan.
2-13-2022	ks	Completed management plan, work breakdown schedule, and monitoring and reporting sections.
2-14-2022	ks	Completed build plan and finalized document.
3-3-2022	ks	Revision based off feedback for final submission.
3-5-2022	ks	Completed addition of meeting notes.

2. Management Plan

2.1. Organization and Roles

Each group member is assigned the following role which includes the outlined responsibilities:

- Writing lead: Brianna Vago
    - Verify the quality and completion of the writing in project documents such as the SRS and SDS.
    - Assign writing related work to other members while monitoring progress.
- Record keeping: Kassandra Morando
    - Ensure that task assignment and completion is being recorded in a complete and timely fashion.
    - Summarize any decisions made or issues brought up during group meetings.
    - Ensure any revisions to records or documents are properly noted and archived.
- Design lead: Kelly Schombert
    - Monitor progress and verify completion of assigned design tasks.
    - Log design issues and changes that need to be made and present them to other members for discussion.
    - Routinely check the quality of the code for style and maintainability.
    - Referred to when any minor design decisions that are yet to be discussed need to be made.
- Code lead: Mert Yapucuoğlu
    - Monitor progress and verify completion of assigned implementation tasks.
    - Upon completion of a task, assign members to begin a new task or assist in ongoing implementation tasks.
    - Check the quality of the code for performance issues and will check for communication issues between modules.
- Test lead: David Han
    - Develop a thorough and robust set of program test cases for all debugging stages.
    - Check throughout the development process that the currently implemented components are being robustly tested by group members.
    - Ensure that any issues or bugs that have been raised are corrected in a timely fashion and will assign debugging tasks.

The type and number of responsibilities for each member will be flexible to accommodate any issues or imbalances in workload or unforeseen circumstances that may arise. Every member will participate in routine meetings and communicate with each other outside of meetings to make decisions, be assigned tasks, and track the progress as the project goes on.

When a large decision (any decision that may impact other parts of the development process) must be made, the relevant lead will decide on a course of action individually and then propose their verdict to the group. The group will discuss if there are any alternatives that may be preferred or if there are concerns with the decision. If there are no concerns or alternatives, the lead's original decision is carried out. Otherwise, a group decision is made and carried out instead. Smaller decisions that will not impact other pieces to the development process will be left up to the appropriate lead.

2.2. Meetings and Communication

Group members will attend regular meetings at the following times:

- Sundays @ 4:00 pm over Discord
- Wednesdays @ 8:00 pm in the John E. Jaqua Academic Center of Student Athletes
- *Additional meetings will be scheduled as needed*

Meetings will last two hours as time permits for individual members. A record of every meeting including but not limited to when and where the meeting was, the agenda, what was reported, what was accomplished, the start and end time, and who attended, will be kept by the record keeper.

Group members will have discussions and report their progress outside of meetings via the following communication methods:

- Discord
- Group MMS messaging

Members are expected to read such messages to stay up to date on any updates or issues that might arise during development.

3. Work Breakdown Schedule

- Week 1 (2/7 - 2/15)
    - Write and submit initial proposal (due 2/9)
    - Determine system requirements and develop subsequent build plan and scheduling algorithm
    - Create github and discord channel
    - Write and submit initial documentation, including a project plan, SRS, and SDS (due 2/15)
- Week 2 (2/16 - 2/22):
    - Begin development of test cases based on user cases and edge cases.
    - Write skeleton code for each system module.
    - Assign and build system modules.
    - Reconfigure file reading and writing modules from Project 1 cold-call system project.
- Week 3 (2/23 - 3/1)
    - Finish building system modules.
    - Complete modular and inter-module testing and debugging.
    - Combine system modules.
- Week 4 (3/2 - 3/6)
    - Complete system testing and debugging.

- If ahead of schedule, consider implementing additional functionalities.
- Review and update documentation.
- Project due 3/6.

4. Monitoring and Reporting

Every task and its completion state will be recorded and tracked on a shared Gantt chart spreadsheet. This chart will also note the group member in charge of completing this task. The chart will be frequently updated to show every self-reported task completion. The responsibility of monitoring progress of each milestone and verifying task completion will be shared amongst all group members. Progress updates and the assignment of new tasks will be done at every meeting.

5. Build Plan

5.1. Plan Details

As further outlined in the SDS, the system will consist of 7 modules. Those modules are builder, schedule, fileI/O, appointment, tutor, athlete, and managerInterface. These will be used to store data, execute the scheduling algorithm, and provide basic UI for file management to managers utilizing the system.

One set of modules to be implemented are those that will interact with the user and the input data. Specifically, the fileI/O module to manage both file input and output, the tutor module to hold the attributes of each tutor, the athlete module to hold the attributes of each athlete, and the managerInterface module to manage the tkinter windows that allow for user interaction. The fileI/O will include the integration of file I/O modules from the previous cold-call system project.

In tandem, the set of modules dealing with information flow and the implementation of the scheduling algorithm will be made. These are the builder module to initialize and control the flow of the system and all necessary data, the schedule module to run the scheduling algorithm and hold information about an individual schedule, and the appointment module to hold information about every appointment object made by the schedule module.

It should be noted that though the implementation is categorized into two separate groups, both groups should have close communication throughout and make sure that functionality and requirements that cross different modules are planned for and do not become compromised.

Once each component can execute its individual functionalities and have been tested for appropriate requirements, the modules will be connected to reach a buildable state. Afterward, remaining ‘must-have’ requirements will be implemented and the viability of implementing remaining ‘should-have’ or ‘nice-to-have’ requirements will be evaluated.

Unless optional requirements and functionality are being added, all resources should be focusing on testing and debugging to reach a stable release candidate by the project deadline.

For a temporal breakdown of these phases, refer to the work breakdown schedule.

5.2. Rationale

An object-oriented approach to designing this system was taken in light of the fact that there are numerous attributes associated with every tutor and athlete that must be compared by the scheduling algorithm and used to build the appropriate appointment objects. An OO design will help organize all input and make it easy to access when performing the logic to build a schedule and when writing and exporting the final scheduling data.

The fileI/O module will follow an integration and configuration process of the file I/O modules from the previous cold-calling system project. This is because the same functionalities needed by a user with the cold-calling system will also be needed by the user of the AWW system. These functionalities include choosing files for import, deciding on a location for output files, and specifying what outputs are needed.

We expect team members to have more difficulty in creating and integrating the modules dealing with information flow and the scheduling algorithm. We will have to assess the difficulty and efficiency of implementing each module early on and reallocate team members accordingly. In addition, even members who are not part of the initial development of these modules should make themselves familiar with the design and rationale for design in the case that they are called upon for assistance later on.

6. Meeting Notes

Check-In
When: 2-9-22 @ 11:20
Where: Allen Price Science Commons and Research Library
Duration: 1 hour
What was done:

- Basic rules to system (i.e. up to 3 students per appointment, separate subjects, available classrooms, GPA requirements)
- “when2meet” schedule to help decide on future group meetings

Group Meeting
When: 2-11-22 @ 9:00
Where: Discord
Duration: 2 hours
What was done:

- Mapped out flow of program events from importing input files, reading data, running the scheduling algorithm, to outputting complete schedule and individual schedules for each tutor and athlete
- Decided on an interface display and import prompts

Group Meeting
When: 2-12-22 @ 12:00
Where: Allen Price Science Commons and Research Library
Duration: 3 hours
What was done:
- Divvied out roles for initial documentation
- Designed system architecture and module architecture
    - Decided on creating appointment objects and storing them in a schedule object as well as using two priorities queues for tracking assigned athlete hours
    - Created a scoring system to find the best schedule of each run instead of using a recursive system

Check-In
When: 2-13-22 @ 21:00
Where: Discord
Duration: 30 minutes
What was done:
- Reported on progress made with initial documentation

Check-In
When: 2-14-22 @ 11:20
Where: Allen Price Science Commons and Research Library
Duration: 40 minutes
What was done:
- Clarified functionality of interface and its relationship to builder and schedule modules

Group Meeting
When: 2-15-22 @ 21:30
Where: Discord
Duration: 40 minutes
What was done:
- Prepared for class presentation

Group Meeting
When: 2-16-22 @ 20:00
Where: John E. Jaqua Academic Center for Student Athletes
Duration: 2 hours
What was done:
- Determined what test cases will be made – small tests to catch edge cases and a big test case with all the current Jaqua athletes and tutors

- Made skeleton code for each module
- Divvied up coding responsibilities for each module and recorded task assignments in Gantt chart

Group Meeting
When: 2-20-22 @ 16:00
Where: Discord
Duration: 1 hour
What was done:
- Decided to prioritize assigning individual appointments over making larger group appointments
- Reported progress made with coding and any setbacks
- Decided on shuffling athlete and tutor data with each call to the scheduling algorithm to ensure a difference in schedule iterations

Group Meeting
When: 2-23-22 @ 20:00
Where: John E. Jaqua Academic Center for Student Athletes
Duration: 2 hours
What was done:
- Reported progress made with coding and any setbacks
- Decided on formatting of input csv files
- Discussed possible formatting options for output files

Group Meeting
When: 2-27-22 @ 16:00
Where: Discord
Duration: 1 hour, 20 minutes
What was done:
- Reported progress made with coding and any setbacks
- Decided on new method to create the large test case with all current athletes and tutors as current method of find information manually was taking too long
- Assigned final documentation assignments and recorded these in Gantt chart

Group Meeting
When: 3-2-22 @ 18:00
Where: John E. Jaqua Academic Center for Student Athletes
Duration: 1 hour, 30 minutes
What was done:
- Reported on all current bugs within the system and the completion status of each module
- Decided on next meeting and the timeline for final submission
- Reported progress on documentation and adjusted task assignments to match needs of group members

Check-In

When: 3-4-22 @ 14:00
Where: Discord
Duration: 30 minutes
What was done:

- Reported progress on fixing bugs within the system and on status of documentation work
- Decided against implementing a way to allow athletes to request more hours than they are required to take because it would require massive changes to the code and the timeline was too short

Group Meeting
When: 3-5-22 @ 20:30
Where: John E. Jaqua Academic Center for Student Athletes
Duration: 4 hours, 30 minutes
What was done:

- Reported progress on fixing bugs within the system and on status of documentation work
- Finished debugging, testing, and documentation work
- Submitted project

7. Acknowledgements

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