Requirements gathering report

Group 4

The first method for requirements gathering was an interview with a faculty member. Our goal with this interview was to get insight into the process of creating a schedule, which started with getting a very simple list of steps needed from start to finish. We can then narrow into each of these steps to find out which steps took the most amount of time and which steps were subjectively the most tedious to complete. We would then try to see if it is possible to save time or improve the user experience for those particular steps, and offer a solution to a problem. These would give us insight into the “should” and “could” portions of our requirements table. The remaining questions focused on gathering data on the “must” and “wont” portions of our table.

We found that the most time-consuming step by far is resolving conflicts in the schedule after the conflict has already been detected. An important discovery is that conflict resolution cannot be automated and requires face-to-face communication with faculty members, as there will almost always be ripple effects and the “human factor,” where moving just one course into a different time slot can cause issues to appear with other courses. A requirement by the ministry of education is that courses must be offered in the morning, afternoon, and evening; and many faculty members either do not want to or cannot do mornings or evenings for personal reasons, such as driving their kids to school in the mornings. It is rare to resolve a conflict without some compromises being made. In almost all instances, faculty members do not get their preferred time slots, and a compromise here may include having to teach a class outside of your preferred time just one day out of the week. Full automation is not something that can be achieved for this reason, and we cannot automate away face-to-face meetings when resolving scheduling conflicts.

We found that the biggest time-save we can offer is offering multiple potential schedules, and providing a UI for the user of the software (department chair, associate dean, etc.) to see and compare different permutations. We can do this by creating a calendar view with the option to only show required courses for a particular program, and separating different courses by color. This gives a simple and fast way to visually see if two required courses are impossible to take together in one semester. The user would repeat this for all the different programs available by the college, and then repeat this for all the different potential schedules drafted by the software or by the user.

The user can create multiple draft schedules, and one suggestion by the interviewee was to use AI models to detect possible conflicts (but not to attempt to use AI to resolve these conflicts for the reasons explained above).

We see UI design as one of the most important challenges for this project, specifically for seeing and comparing different schedules to spot course conflicts. We need the UI to fit a lot of information (one program can include many required courses) without it being overwhelming. We need to carefully find the balance between showing all the information needed at a glance so important details are not hidden behind mouse clicks, while not showing too much which can create a cluttered-looking UI and potentially not save much time over current methods/solutions on the market by other scheduling software.

# Observation of existing systems

## Software: Appointy

Appointy has a feature where a business staff can schedule classes with students. During initial step up it asks the owner to enter the information for all the staff and all the classes. In the class scheduling, teachers schedule their own classes which is different than ours, but a few useful features were seen. All invalid times were greyed out. (all hours the business was closed). Then if the class was 50 minutes long. The next class could only be scheduled 10 mins later. There is also an easy way for managers to see how many classes are in total scheduled and how many students are in it and who booked the class.

## Software: Wix

Wixs allows users to add and remove classes in their calendar view. The user has to select a time slot of 15 mins interval and then a pop up shows up for asking the type of time block it is. A topic is chosen and then a start time, duration and if this repeats or not. This seems to be a very intuitive way to schedule a few classes but may not be ideal for scheduling 100s of classes.

# Requirement Gathering via Documentation

## Software: Creatrix Campus

Creatix is specifically designed for colleges, but we could not get access to it aside from its webpage marketing as it is behind a paywall. We found many features here that we can see will be useful and gave us a good bases to start are project on. It has features which allow for substitute teachers (which we do not need at Douglas), automated scheduling, attendance management, online calendar syncing and editing the schedule to allow for classes cancellations. These features were determined to be not needed as Douglas already has ways to handle these situations. The most useful feature we saw is the AI conflict resolution. We were already planning to add such a feature, and this one gave us ideas on how to go about it. It has a UI for easy conflict diagnoses and automatically tries to resolve all conflicts.