

Assignment 2: BSPS Framework

Introduction

The objective of this project was to create a framework that alleviates the common problems found in the advising process for both students and advisors. While this system is generalized to meet the needs of advisors and students at all universities, the beginning stages of our design process have been specific to Illinois College. The need for this system in a university setting became clear as our team encountered the complex and time consuming current materials for advising at IC. The current system, while functional, has room for improvement in regard to efficiency. For example, the current system is complex and requires portions of the Advising Sheet to be checked manually with each advising session, which is not only time consuming, but it leaves room for human-error in the advising process. Issues with the current system like this one force students and faculty to take different approaches to the advising process, leaving the advising process nonuniform across the faculty base. These different approaches have their own disadvantages like the extra time and effort on the advisor's end or loss of information for simplicity. A solution to these problems with the current advising application is especially relevant to students and advisors at any university, especially Illinois College, as the majority of students and advisors use routine advising sessions each semester to check progress and choose future courses. Our approach to solving this issue is creating a framework that combines and simplifies all advising processes to meet our general education requirements (e.g., Illinois college BluePrint). Compared with other good advising systems, we believe our framework could provide better solutions including the user-friendly interface, clear advising sheet, and course recommendations. Our solution would not only make the advising process less time consuming and simpler for advisors, but it would also make the advising process uniform at Illinois College, take away the guesswork and possibility of error in picking future courses, and aid in student understanding of general education requirements. Although we elicited requirements from users at Illinois College and developed solutions specific to the college, the framework is generalized to be utilized for advising systems at all universities and schools.

Requirement Elicitation Design

Participating subjects

Advisors, who were professors and department heads, from five different departments participated. The departments were English, Maths, Business, World Language and Culture and Computer Science. By choosing advisors from a variety of disciplines and experience levels, we were able to gain a broader set of data for Requirements of the system. While the real system will have all the departments integrated into the database, by selecting just five departments, we will be able to capture the complexity of the framework adequately.

The next group of users and also the primary users are the students that use the advising system. We randomly choose six students from IC with at least one from the categories, Freshmen/ Sophomore, Junior/ Senior, Transfer students and International students to exhaust the groups of students that use the advising systems. We take care to choose students who would have very different experiences with advising, and focus on the different requirements.

Data Collection

Our six team members individually interviewed 2 participants (1 student, 1 advisor) in different weeks to prevent bias or skewed results. We collected data via a short 10-15 minute mainly asking open-ended questions designed before the session. Through this relaxed discussion-type interview, we collected the data with a focus on requirements and case scenarios.

Step 1: Pre-interview Questions

The questions are open-ended and are separate for advisors and student participants, attached below in this document. The questionnaire begins with general questions like asking the interviewee how the advising process goes? This makes sure that the interviewees is in control and can start the conversation on what is most apparent in their minds. Taking this top-down approach, the questions are sequenced to remove biases that can be introduced by the interviewer. Again, the questions are chosen to help interviewee think about scenarios that can later be translated to use cases and requirements. Furthermore, we also make sure to invoke wishes of the interviewees that they want to see in our new system. This means we have data from the users on their interpretation of a 'good' advising system.

An additional part of the advisor interview is a mock advising session, where the interview observes and takes notes while the advisor advises a student.

Step 2: Interview

The interview is not timed to not put pressure on the participant but generally takes about 10-15 minutes. The interview asks questions directly from the prepared questionnaire (Appendix A) and takes notes with a pen. Interviewer takes notes directly as the participant explains without interpretation as it would otherwise introduce biases. Moreover, interviewers do not comment or interrupt the participants but have good suggestive dialogues such as repeating the participant to make sure that ideas are communicated effectively or asking the participant to clarify.

Step 3: Data Crystallization

After the raw data collection from the interview sessions (Appendix B), the team attends a meeting where each interviewer explains their interview process and data to the rest of the team. The team recompiles the raw data into specific categories that we later used to create functional requirements, nonfunctional requirements, priorities, use cases, scenarios and a class diagram.

Results

After the raw data collection from the interview sessions, the team attends a meeting where each interviewer explains their interview process and data to the rest of the team. The team recompiles the raw data into specific categories that we later used to create functional requirements, nonfunctional requirements, priorities, use cases, scenarios and a class diagram.

From the data we gathered through the Requirement Elicitation, we found the greatest concerns about the current system.

System Issues-

- Can't have multiple tabs open making cross referencing advising materials difficult.
- Adding and removing courses each semester makes it hard to prepare ahead of time.

Information Issues-

- Need to manually check for blueprints fulfilments, such as foundation, exploration, and embedded credits.
- Not knowing if certain classes are offered every semester or every other semester, and it might be required for certain majors.

Interface Issues-

- Interface is not clean and not well organized, it is also difficult to navigate.
- No way of knowing what you should take next, for your major or minor.
- New Blueprint 2.0 doesn't tell you what you still need.

The greatest concern about the current advising materials was that the system is repetitive and requires manual checking which makes advising sessions time consuming. There are two different blueprints now, and the old one is very cluttered compared to the new one, but the new one still doesn't update the classes taken and still requires fulfilments to be checked manually, increasing the time and risk for human error in advising. Aside from the issues we found with the current system, advisors had other concerns about the framework after implementation. It was clear that our biggest concern was to create a clean interface that was easily navigable and could give the student and advisor all the information they may need without needing multiple tabs or windows.

Solutions

From these results, we moved on to creating our functional requirements that resolve user issues with the current system, nonfunctional requirements, priorities, use cases, scenarios and a class diagram. As mentioned, a school's educational advising system is important to a student's academic career. There are various rules and measures that are taken when building a student's

possible schedule. Thus, most of these rules had to be incorporated into the BSPS, then translated into the functional objectives of the system.

The system's functional objectives were based on the different components of the system. The primary system functional objective was the ability to generate a recommendation report for students. This report offers students possible course selections that can fulfill their missing major (if declared) and academic requirements. Other system functional objectives consist of the ability to generate a Student Transcript, which holds all of a student's academic progress. The system must be able to generate an Advising Report, which shows a student's missing academic and major (if declared) missing requirements. Also, the system should be able to generate a Master Course which holds all of the course information including course offerings, descriptions, meeting times and course academic fulfillments. Finally, the system should allow the Registrar to update a course's information including the academic fulfillments, course number, name, description and course schedule. Because most of these system components exist in many different colleges today, we decided to devote our focus into creating the Recommendation Report system component. Thus, this system component will be discussed more in detail.

The use case diagram for this system is shown in Figure 1. The actors include the

Student, Registrar and Advisor. Along with these actors, two abstract entities are also represented, the Generic User Actor and the Generic Faculty Actor. Starting from the top, the use case diagram states that only advisors can use the *Check Students Requested Courses* system component. The component's name describes the component's action. Advisors are part of the Generic Faculty Actor. Thus, they inherit the components which this actor can perform, which will be discussed later.

Next, on the diagram is the

Registrar actor. Shown by the diagram, only the registrar is available to

use the *Update Master Schedule* and *Update Course Credits* system components. The Registrar

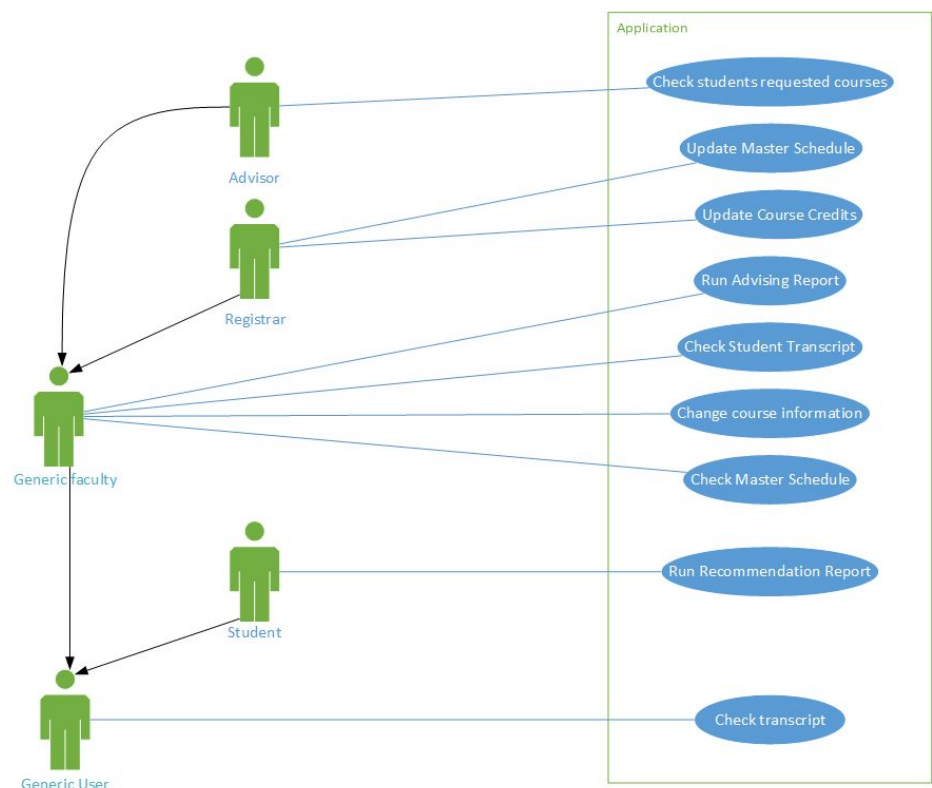


Figure 1: BSPS Use Case Diagram

actor also inherents the ability to perform the Generic Faculty’s granted components. Next, is the Generic Faculty actor. This actor is able to perform the *Run Advising Report*, *Check Student Transcript*, *Change Course Information*, and *Check Master Scheduler* components of the system. As mentioned, the Registrar and Advisor actors both inherit the Generic Faculty’s granted components, thus, they also can perform the same operations as the Generic Faculty actor. There is also a Generic User abstract entity which is inherited by all actors in the diagram. This actor has one system component assigned to it which is *Check Transcript*. Thus, because every actor inherits the Generic User, all actors within the diagram can perform the *Check Transcript* system feature. Finally the last actor in the diagram is the Student actor. Only this actor has the ability to perform the *Run Recommendation Report* system component, which, as mentioned, is the main system component which was integrated into the BSPS.

Use Case Name:	GenerateRecommendationReport
Actors:	Student
Summary:	A student wishes to check recommended courses for next semester based on the Blueprint fulfillments.
Basic Flow:	<ol style="list-style-type: none"> 1. Student signs into their institution’s respective webpage 2. Student navigates to, and opens the advising application 3. Student selects the ‘Recommendations Report’ button/field 4. The application fetches the master course schedule, course info, and student transcript from the institutions database 5. Application returns a report displaying possible courses for the next semester that fulfill the student’s missing Blueprint and Major requirements 6. Application return Recommendation Report as PDF.
Alternative Flow:	<p>Step 4:</p> <ul style="list-style-type: none"> • If the institution's database is down, Application will fail to fetch the needed transcript, master schedule, and course info. As such, the application will display an error: “failed to fetch future courses”. Application will return to the home page. <p>Step 5:</p> <ul style="list-style-type: none"> • If student hasn’t declared major, Recommendation Report recommends courses that fulfil only Blueprint Requirements
Extension Points:	none
Precondition:	<ul style="list-style-type: none"> • <u>Student</u> must be able to sign into their institution’s online resources. • Student must be a current or incoming IC student • Application must have access to institution’s database
Postcondition:	<ul style="list-style-type: none"> • Application must return a form displaying course recommendations

Figure 2: Recommendation Report Use Case Table

The recommendation report is the main system component because this component mainly focused on helping students. The report's use case table is described in Figure 2. Unlike the other mainstreamed components, this component creates a new change to alleviate the burden of choosing future classes. Thus, this component can affect many of the actors described in the diagram (even though it is only available to students, shown in Figure 1 and Figure 2).

Remaining Required BluePrint Requirements		
Section	Needed	Suggestions
Explorations	<u>4 of 8 completed</u> <ul style="list-style-type: none"> - Creative Expressions: Studio - Science in Society: Lab - Social Spiritual Philosophical 	<ul style="list-style-type: none"> - TH231 - SO101
Transformations	<u>0 of 1 completed</u> <ul style="list-style-type: none"> - Senior Capstone Experience 	<ul style="list-style-type: none"> - CS485
Ethical/Responsible Actions	<u>1 of 3 completed</u> <ul style="list-style-type: none"> - Major Course - Ethics course / Course with ethics component 	<ul style="list-style-type: none"> - CS485 - CO220
Major Courses	<u>8 hours still required</u>	<ul style="list-style-type: none"> - CS350 - CS380 - CS410 - CS420 - CS440 - CS460

Figure 3: Recommendation Report File

A visual representation of the recommendation report is shown in Figure 3. As mentioned above, the main focus of this report is to give students suggestions about possible courses that can complete certain academic requirements. Thus, the form is separated into three columns. The first column addresses the section of the academic requirement. The second column addresses the academic requirement needed. Finally, the third column addresses a list of courses which can be taken to accomplish some of the academic requirements listed in the second column of the form.

The main goal of the BSPS is to alleviate the process of choosing a student's schedule. Thus, the main goal of the system's components is to provide assistance to all of the actors which are involved in this process.

Future Work

In conclusion, this system allows institutions to easily and efficiently have their student body choose their courses. This is done by the generalization and standardization of the course recommendation systems, while also making the interface look clean and understandable for the end user. This allows for many institutions of many levels to create cohesive learning plans. Through this a student can maintain similar planning strategies throughout their whole education thus mitigating: confusion, unnecessary courses, time, money, etc. Furthermore, the framework with which we used to set up the program can be reformed with any title and then be sorted in whichever way the programmer would like. It doesn't have to be classes or majors, it could easily be branches in a business or districts in a city. Through these simple recommendation tenements we could then set up AI to further develop recommended courses, a system based on a student's likes and dislikes, a system based on the students' reviews of an instructor, a system based on what courses the students' friends liked, or Students of the same major liked. The framework is customizable to the goals of individual institutions. All in all this system leads to a more effective advising process, which leads to happier students.

Appendices

Appendix A: Requirement Elicitation Questions:

https://docs.google.com/document/d/1PGZKoPy51hw-8DL-6SYuG64sHH8_c65hNG85wqAR3TE/edit?usp=sharing

Requirement Elicitation Design

Interview with Advisors

1. Ask following questions, probing for more specific answers

- Describe an average advising session.
- What is the biggest issue with the blueprint?
- What are some other problems/annoyances/ greatest difficulties within the average advising session?
- Do you like advising on the computer? Why?(navigability etc)
- What features should be automated
- Are there any differences in advising different students(freshman vs upperclassmen)? Any differences in advising?
- What are typical advising questions students ask(picking major, finding graduation date etc, classes for next semester), and what would help you answer them?
- What features would you like to have?

2. Ask them to do a mock advising session, observe and make notes of any other problems that they did not mention in interview portion

Interview with Students:

1. Ask following questions, probing for more specific answers

- Describe the typical things that goes on in your advising sessions.
- What do you find annoying?
- How easy is it for you to set the time for advising?
- Do you usually have to take more than one session to complete? What part is the most time consuming

Appendix B: Requirement Elicitation Data/Notes:

<https://docs.google.com/document/d/1XCp6xFu4NbuU9waCGPByvZVOFzk1d3KASwoTeddqiZw/edit?usp=sharing>

Advisor Interview Notes:

Advisor- English department chair, advises freshman-senior, and honor's department students:

- Freshman advising sessions- normally just for showing resources on connect2(advising worksheet, blueprint grid, blueprint poster, courses by semester, catalog), checking course selections for next semester, checking transfer credits

- Not about telling what classes to take, just making sure classes they pick fill blueprint requirements or introduce possible major (Students come with course selections for next sem usually)
- Upperclassmen advising sessions- goes through advising sheet checks for requirements not met, finds classes to fill them, checks major requirements, finds classes to fill
 - At the beginning she checks graduation year, gpa, convos, catalog year
 - If GPA is high she tells them about Phi Beta Kappa and the course requirements for that
- Advising Sheet
 - She prints this out and marks on paper; has to print new with each advising session as requirements are filled
 - New information security laws about having paper copies of student info will probably change this in the future
 - Goes through top- bottom checking each section and counting manual verification sections in order to find missing course req.
 - **Biggest issue is manual verification because with each advising session(same student each year) she has to check the same section again(redundant)- she suggested if they have to have manual verification at least let her update once she checks the first time to eliminate redundancy. Or sending these edits to the registrar to confirm before updating.**
 - A lot of the sections could be counted automatically because the information is there, they just aren't
 - Section with 8 courses from different sections and embedded experiences are the most trouble (especially with two majors)
- English Major Specific
 - Concentration requirements require manual checking esp
 - certain courses are not put into major categories because the course tags change each year like 388
 - Independent studies don't get put in
 - Course offerings each year affect students ability to meet certain requirements, so the department usually substitutes courses.

Advisor: (Head of Math Department)

Time: 3:58 - 4:13 (2/26/20) Wednesday

Notes:

- 1.) Deciding Major Undeclared (Freshmen) or Major Declared (non-freshmen)
- 2.) (a) Major Undeclared: Ask for what interest the student has. For instance, a student has a passion for Arts. Students should do what they love. At the same time, there are options for matching different majors and minors so students can still major in arts and take a minor in business.
- (b) Major Declared: Take at least one class of their major. For Math Major, students take a placement test that helps them choose where to start. Typically starts from Calc 1 but

some take Intro to functions or Calc 2 depending on their maths background in high school

Note: For the first year, try to complete the basics blueprint requirements like Speaking and Writing classes and First-Year Seminars.

For Math Majors:

- 3.) Think more about filling in the General Education section of the blueprint.
- 4.) Take about five maths classes by the end of sophomore year: Calc 2, Discrete Maths, in first spring, then Linear Algebra in next fall and Calc 3 in spring.
- 5.) Math majors are required to take 40 credit hours of math classes.
- 6.) Core classes: MA201 (Discrete), MA213 (Calc 1), MA 223 (Calc 2), MA233 (Calc 3), MA323 (Linear Algebra)
- 7.) Two from proof-based classes: MA302 (Survey Geometry). MA373(Real Analysis), MA 383 (Abstract Algebra)
- 8.) Two from other electives.
- 9.) Senior Seminar or capstone

Additional comments:

- 1.) Students typically visit when they want to know what classes to register for next semester or when they want help deciding a major.
- 2.) Even when recommending GE, prerequisites are taken into account. A physics class would be recommended to someone who has a developed maths background.
- 3.) The advisor pulls up the blueprint and connect2 course search on her computer every time she does advising.
- 4.) The current problem is connect2 disconnects and gives an error message when opened on different windows at the same time. The advisor would like to be able to have different parts of the interface laid out at the same time.
- 5.) Higher Maths classes are offered every other spring or fall.
- 6.) Cases of new classes and dropped classes are possible.
- 7.) More intelligent filtering and organization of course search function.

Advisor: Prof. John Rush

1. Check what courses they have already taken. What their interests are. Pull up the advising sheet and look at general ed classes they have already taken. Then talk about what major they would decide on and work towards the blueprint at least. Then take a few classes around different majors to see what they're interested in if they're undecided and new.
2. You have to pull up connect2 to check the blueprint. And not all classes are offered every semester. So you have to make assumptions as to what will be taught the semester after the next semester. But being able to offer a list of courses the student hasn't taken would be useful.
3. Implementing a system that the students can use on their own would be scary because then you risk losing the student to professor interaction. And having that interaction

could be helpful if something new comes to mind. Or being able to throw suggestions and reasoning during the advising session is good. There are some students that already just follow the course plan in the catalog and sometimes they just go through the courses in an awkward order or they take the classes and decide later that they don't want to continue in that major. So having the advising session in some way would be great, at least in the beginning.

4. For advising sessions differing between the freshmen and seniors is more like. Giving the freshmen a broad array of choices so that they can have a general idea for each major before having to choose one. And for a senior, sometimes they only do the courses for that major and have trouble finishing enough classes to gain enough credits to graduate. So what I recommend is to have a wide variety at the beginning, then around sophomore and junior year we can go through major courses, and near the end once a major is declared and you are close to finishing. You can then take more broad classes that would assist in your field of study.

Additional Notes

Currently the best way to check what blueprints you have are to go through connect2 and to use the spreadsheet that they provide us to check all the courses. But that spread sheet doesn't tell you what will be offered in which semester so unless you know for sure it's being offered then you have trouble scheduling anything farther than one semester in advance. If we can have something that allows us to see what class covers what blueprint and keep it consistent enough to provide a solid plan, then we can cut down on memorizing what class is what by their number.

Advisor : Dr. Devin Bryson (World Languages and Culture)

- Software is not really easy to navigate. Mentioned about the different pages and reports that have to be open in order to show student what courses they have taken and they could possibly take
- Interfaces (and software) are not very helpful, they are outdated and not easy to use
- Hard to navigate the right forms and pages that are necessary to give a student advice about what classes they should take
- Would like to automate the process of choosing classes for a student
 - Possibly give students a variety of classes that they can possibly choose, but not also add flexibility to this feature. Do not force students to choose a class!
- Likes for students to have background knowledge of the blueprint
 - Possibly implement a system which will give students a variety of classes to choose concerning their current class history, that way, Professors just have to give the "okay" for students to take.
- Ease the process of doing the research to find a student's future schedule.
- Would like to somehow corporate the Catalog within the system

- Allow students to get a description of a certain class?
- Connects to creating a system that would allow student to know what classes they should take
- Implement a process where we can know what classes a student can possibly can take in the future concerning their class history and their selected major (if one is chosen)
 - Students with different majors can find the classes they need in the future semesters.
 - Professors usually have to direct students to other advisors due to their lack of knowledge in other fields (concerning what class the student should take next, usually a First Year Problem)
- CLEANER INTERFACES.

Advisor: Soma taken 2/27/20 from 2:50 to 3:20

- An average advising session includes: going over transcripts and planned graduation date. Then selecting necessary major courses. The student is free to select whichever other non-major or blueprint courses they want, "I'm not going to tell you what to take" - tough nurse mentality.
- Soma does not use the Online resources, they are confusing and it takes too much time. Rather she uses the secondary handout and expects the student to keep and update this form.
- The main differences between upper and lower classmen is whether or not they know their plan.
- The most important thing is making sure students graduate on time. This usually results in Soma making a long term plan.
- Things to fix: cleaner UI, fix autofill forms so they actually autofill. Reorganize blueprint for readability.

Student Interview Notes:

Junior, Economics and Accounting major

- Advisor sends out a google calendar with times students can book advising session
- Advisor pulls up the advising sheet on the computer. And only looks through what he needs then the advisor suggests classes to meet requirements. Go over courses student picked for next semester
- Didn't like how sections that are completed don't show as completed. (manual check portions)
- The sections that aren't necessary if you have others completed (like creative non studio) show as incomplete. Would be easier if they weren't shown
- Counting for each section (Explorations) is the most time consuming and can get confusing with certain requirements.

- Knowing what classes fit the unfulfilled requirements would be easier if they had options of the form
- He prefers having it on the computer over paper, but I observed that he can't/doesn't mark off what's finished.
- Knowing future semesters
- Heavy courses
- Checking grade level and high level course

Student: Business Major (International Transfer Student)

- 1.) Overall, find it easy to communicate with the advisor.
- 2.) He spoke praises for the advisor who prepared all the requirements and pointed out all the available classes he can take for the coming semester. He had already drawn out a complete list of courses he will be taking until graduation.
- 3.) He typically goes to visit the advisor for asking about what classes to take in the next semester.
- 4.) Advisor pulls up all the courses in a list available on Connect2 and he picks the ones he like for GE.
- 5.) The advisor does not use the advising sheet but he sometimes checks it but finds it really confusing. There are a lot of blanks for some category even though only a few courses are required there.

Student (Transfer)-

1. I met with my advisor, my advisor knows the best courses so he gave me the best courses I needed to take for my major and minor. Give me a list of courses and I would take those courses to fill my major or minor and my blueprint. A couple for major and minor then a couple, like selective courses.
2. There was a course I could have taken next Fall but it's only offered in Spring.
3. The program would help students be able to pick the courses, then go to the advisor afterwards and talk with them about what you might take. And you can judge what courses would be hard and not overload yourself with 4 hard courses. Balance out the workload per semester.
4. Coming in as a Freshman, I had a general idea as to what I wanted to do but not specifically. Now I know my major and it helps set up my classes now. When you have your major you don't have waste time on a class that you might drop and even if you don't drop it, it might not allow you to work towards the major that you decide on.

Additional Notes:

When major requirements change, previously required classes should still hold high priority because it used to be required and it would still help you. Will still benefit towards your major. Recommendation system like what is most

popularly taken, highest average gpa class, and an advisor recommended list and order of classes that you should take. If the program is offered to students publicly then make more of a push towards making advising sessions a requirement after the student knows what class they plan on taking the following semester.

Student :

- Process of accessing blueprint is very slow
 - Some interfaces are very faulty and slow.
 - Hard to understand
- Process of choosing classes takes too much time
- Comprehension of the Blueprint is difficult
- Similar issues to Professor's concerns

Student: Female, Sophomore MIS major, taken 2/27/20 from 10:50 to 11:10

- Student does not attend advising sessions
- Student prefers to use the online resources, however the auto-generated reports do not fill all fields and as such are confusing and annoying.
- One main annoyance is navigation to find courses offered in regards to times, days, semesters, and blueprint requirements.

Appendix C: Class Diagram:

https://drive.google.com/file/d/1XHCHL3je7vF_e4iRGJgEXlAj5uP1psew/view?usp=sharing

checkStudentsRequestedCourses
getReqCourses(): Array

updateMasterSchedule
newCourse(): String
removeCourse(): String

updateCourseCredits
updateCourseCredits(): String

runAdvisingReports
runAdvisingReport(): advisingReport

checkStudentTranscript
checkTranscript(): Transcript

changeCourseInformation
updateCourseName(): String
updateCourseDescription(): String

checkMasterSchedule
checkMasterSchedule(): MasterSchedule

runRecommendationReport
getRecommendationReport()
: recommendationReport

checkTranscript
getTranscript(): Transcript