Team Bae Squared ([baesquared@umich.edu](mailto:baesquared@umich.edu))

Irene Bashar, Medha Kapoor, Preksha Shah, Hilda Wang

12/5/2022

**UM Undergraduate CSE Override Project**

**Final Report**



# **Table of Contents**

1. Executive Summary
2. Introduction
3. Background Information
4. Methodological Overview
5. Findings
6. Recommendations
7. Conclusion
8. References

# **Executive Summary**

This past semester, we assisted the UM CSE Advising Office. The advising office assists CSE students primarily with advising for academic issues like course planning and course overrides. Since they serve both undergrads and grad students, they offer a wider variety of resources to meet the needs of both groups. One of UAO’s responsibilities, administering course overrides, was becoming a very lengthy and cumbersome process, so we were tasked with studying the current process and offering input on how to further streamline and automate it. As part of our solution formulation process, we conducted user interviews and leveraged that data to perform further thematic analysis and figure out the main problems with the current system.

We have now created this report to share our findings and further recommendations. This report contains some background information on the UM CSE Advising Office, our data collection and analysis processes, and of course, our findings and further recommendations.

# 

# 

# **Introduction**

In this report, our team, Bae Squared, examined the course override process for the University of Michigan CSE Undergraduate Advising Office (UAO). To gain a better understanding of the system and its pros and cons, we conducted multiple user interviews and multiple forms of data collection and analysis. We have now formulated our research findings and recommendations, the contents of which are presented in this report.

# **Background**

**Client Mission Summary**

Our team’s client was the UM CSE Advising Office. The advising office helps out students within the Computer Science and Engineering school with all kinds of issues, whether it’s figuring out what courses to take, determining student override eligibility for class, or finding relevant program resources, etc. They serve both undergrads and grad students, so they offer a wider variety of resources to meet the needs of both groups. For our project, we only dealt with the CSE Undergrad Advising Office.

**Client Problem Summary**

There are currently multiple steps in order to determine class override eligibility. The first of these steps, which is inputting student information, is a massive hassle as it generates numerous forms and spreadsheets. This forces someone to manually go through student information, and still requires additional steps to determine whether a student can override a class. In addition, CSE Advising Team primarily uses Google Forms, which can get very long and unmanageable, as well as not being the most secure application when it comes to dealing with potentially sensitive student information. Furthermore, WolverineAccess only grants students override permission one class at a time, which can be annoying for both the advisors and students when it comes to planning out their schedules. We were requested to provide suggestions that involve automating more of this override request process, or at least streamlining it to a point where numerous forms and spreadsheets aren’t generated each time a request occurs.

**CSE Advising and Responsibilities Overview**

At the University of Michigan (UM), the Computer Science and Engineering (CSE) department houses an advising office for undergraduates, officially known as the CSE Undergraduate Advising Office (UAO). The office assists students with a variety of concerns, such as course planning, course overriding, career planning, and industry knowledge. Furthermore, UAO’s advising team is composed primarily of three types of advisors: staff, faculty, and peer advisors. They handle advising for coursework, major/minor declarations, course overrides/special requests, and course policies. (“Current Undergraduate Students: Computer Science and Engineering at Michigan.” )

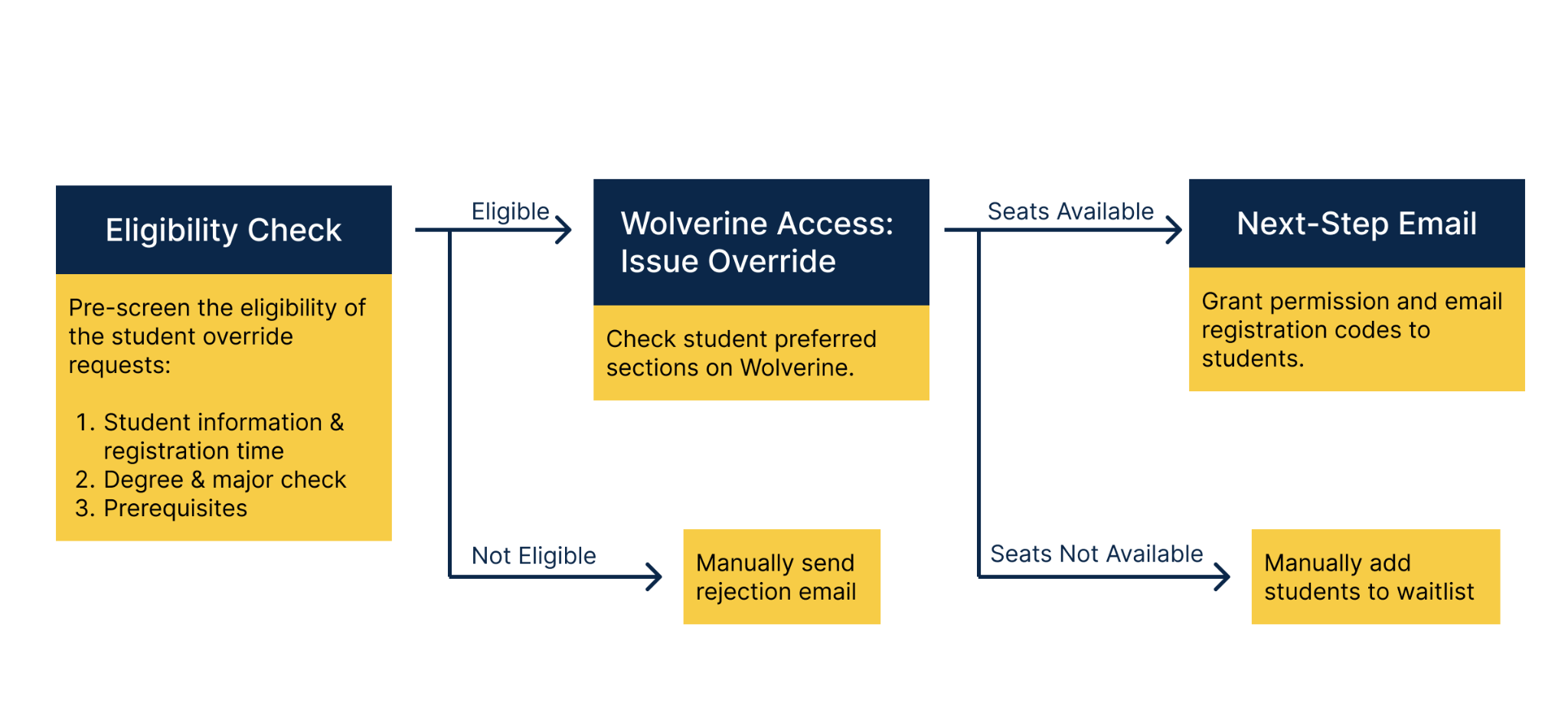
**CSE Course Registration and Course Override Request Processes**

Although UAO does its best to serve its students in an efficient and thorough manner, it is forced to handle massive amounts of student information, which has resulted in information processing systems that can be slow and inefficient. This influx of student information heavily impacts the course override system, which heavily depends on manual processing of student information, which is primarily contained in Google Forms and spreadsheets.

1. **Course Override Requests**

Course override requests are considerably less straightforward to handle than the course registration process. This is largely due to the fact that there are multiple types of override requests that can be made. For undergraduate students, these are the current types of override requests they can make:

1. Permission to take an undergraduate CSE course (EECS 400-level or below):
2. Permission to take EECS 280 without the enforced prerequisite
3. Permission to take a graduate CSE course (EECS 500-level or above)

Although each form slightly differs from each other, students are primarily asked about basic information such as a student’s name, UMID, class registration appointment, school & major, etc. They are then asked to indicate what course they would like to take and when. Finally, the form asks students to mention if they have completed the course prerequisites and to explain why they should be allowed to take their desired course (“Letter, Form, and Override Requests.”). After the form response is collected, UAO follows an override eligibility decision-making process, as shown in the chart below: 

As seen in the chart, UAO first conducts a pre-screen for the student, checking for their registration time, degree & major, and cleared prerequisites. If a student is deemed eligible for an override, UAO will check to see if there’s a class section available based on the student’s preferences. If there are still available seats, UAO will grant the student course override permission and email them a registration code. If no seats are available, students are then manually added to a waitlist. On the other hand, if the pre-screen determines that a student is ineligible for an override, UAO will manually send a rejection email to the student.

# 

# 

# 

# 

# 

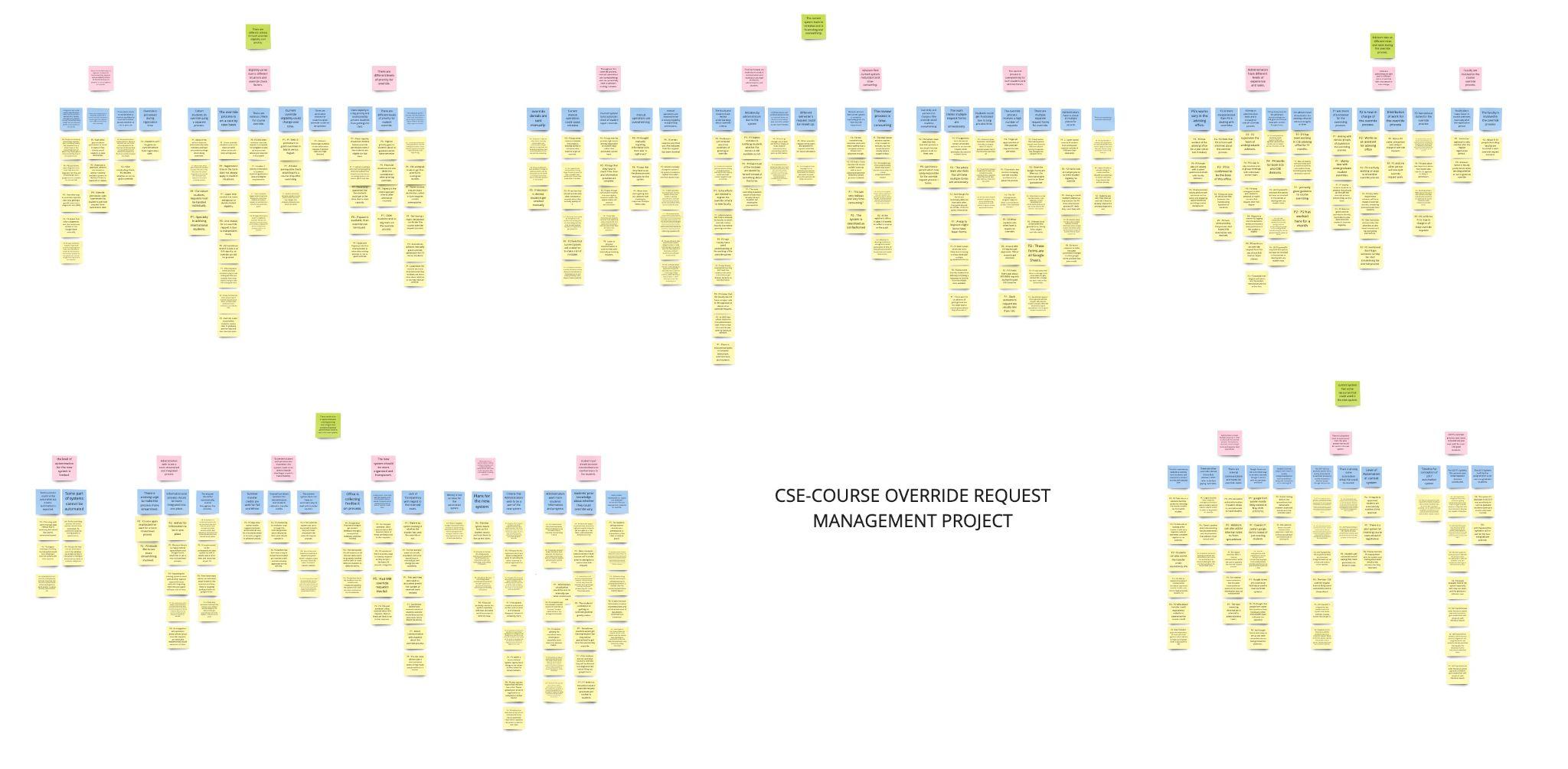
# 

# **Methodological Overview**

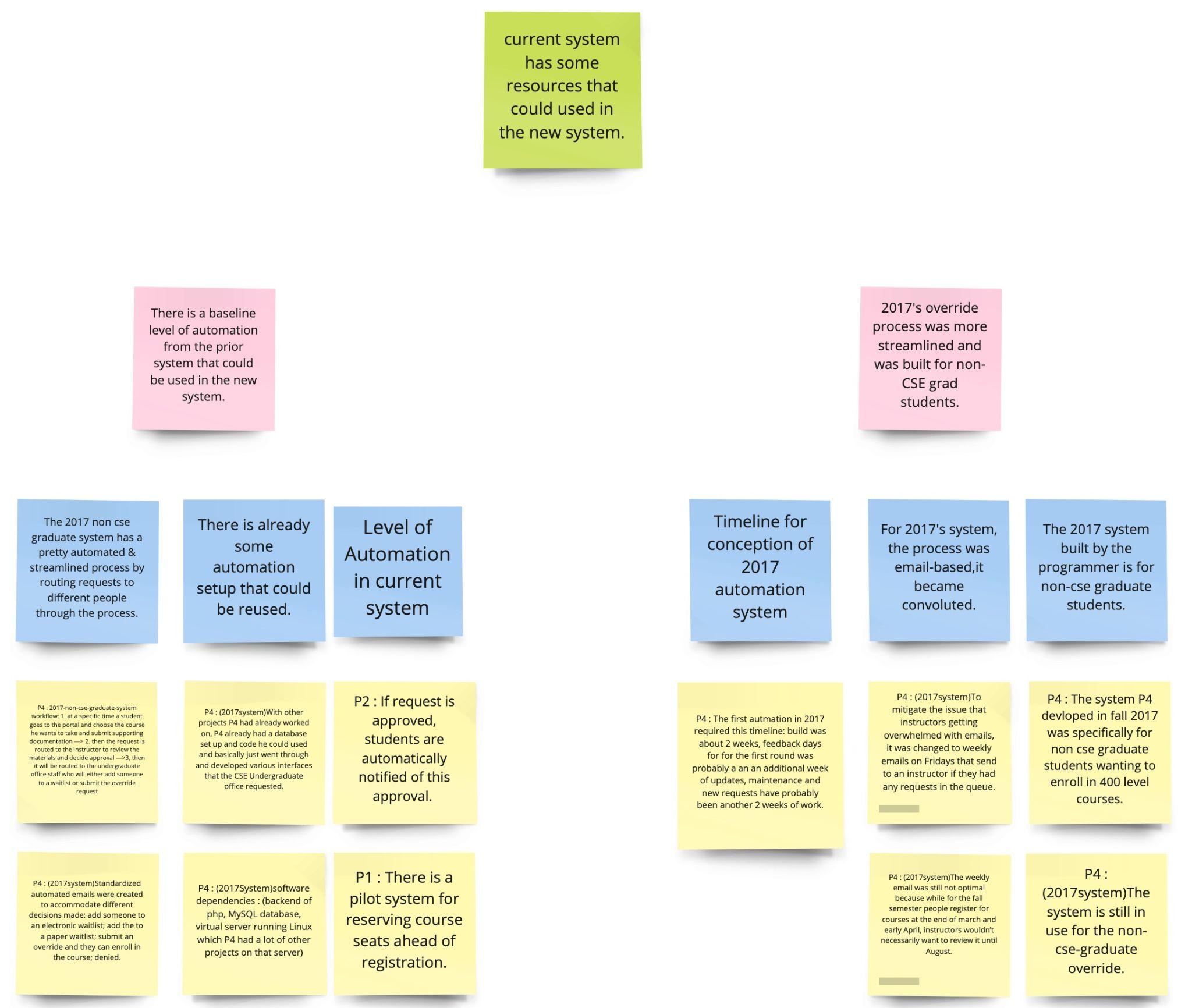
**User Interviews**

In order to help us begin finding a solution, we first conducted multiple interviews with UAO staff members. This included a group consisting of office administrators, advisors, and a programmer who has previously worked with the office before. We were able to ask for their thoughts on the current override system, previous systems that had been implemented in the past, and features they would like to see implemented in a potential override system. We also took a look at the current system, and were able to go through various relevant spreadsheets and forms that the administrators use and interact with on a daily basis.

**Data Analysis - Affinity Wall**

We turned our interview notes into one-sentence summaries for each line of notes. Each summary was then transferred onto virtual yellow sticky notes so that we could start grouping notes and observing recurring themes and patterns amongst them. Based on the initial groups that were created, we then created more high-level groupings, represented by the blue notes, so that we could go one step further and identify overarching themes within those initial groups. We repeated this step one more time, represented by the pink notes, which then helped form the basis for our findings. Because we had a high number of pink notes, we repeated another round of groupings with those, which resulted in our final round of findings, as represented by the green notes. These findings are not actual solutions themselves; rather, they helped guide our solution ideation process by reminding us of the core issues and themes that our solution needed to address. For reference, we’ve attached our chart, which is also commonly referred to as an affinity wall in UI/UX terms:****

To also help visualize what this hierarchy of notes looks like, we’ve also included a close-up example of a full grouping. Here, we have yellow notes at the bottom, which represent direct quotes and observations we pulled from our user interviews. Next, our blue notes mention the inferences we were able to pull from yellow notes. After that, we have pink notes, which are inferences we pulled based on our blue notes. Finally, we have a single green note that contains our main finding for this group of notes, and represents the culmination of all the inferences we’ve pulled from our user interview notes.

****

# 

# **Findings**

**Finding 1: There are different criteria for both override eligibility and priority.**

**1.1 Eligibility varies due to different situations and override check factors.**

When override requests are received on the administrator’s side, there are three pieces of information to check in order to determine the student’s eligibility for the override:

1. Student information, especially the registration time. This will determine when the overrides are issued, since overrides can’t be issued prior to the registration date.
2. Student’s degree program/major. It is important to know whether students have declared their major or not in order to request certain major-restricted courses..
3. Prerequisites requirement. This checks to see whether or not students meet any prerequisite requirement, and may also require double checking.

In some situations, however, the predefined process mentioned above could not be followed due to various reasons. For example, the Shanghai cohort students request overrides using a separate process, and while most cohort students are in the same situations, their requests must be handled individually on the administrator side. The unfamiliarity with the override system has also caused many of the cohort students to make mistakes and have misunderstandings when doing overrides. Sometimes, there are also diagnostic tests students must takes to satisfy override requirements, which are separate from the standard eligibility checking process.

Moreover, there are also other unpredictable factors in determining eligibility that can occur. Sometimes, the eligibility of the student’s request can be altered due to failing prereqs after permission is granted or students just decide to no longer take the course after submitting the request. Scenarios like the latter caused one of our interviewees to mention that “it might be too premature to grant overrides in March versus August.“

**1.2 Even if students have passed the pre-screening requirements, there are different levels of priority for granting override.**

After the students have satisfied all the requirements, it is also important to note that permission is not guaranteed due to reasons like limited classroom capacity. During an interview, one of our interviewees showed us the administrators’ internal Google Spreadsheet, where administrators discussed how to prioritize override permissions amongst different requests. For example, one interviewee mentioned that, “the highest priority goes to students who are about to graduate within the same semester and financial situations are also taken into consideration when granting overrides.” Many factors are taken into account and there is no absolute standard as to which priority goes first, which slows the entire process down.

**Finding Implications**

* The undergraduate override is very complicated, so semi-automation might be more realistic than creating a totally automated system.
* Since some checks require external sources, such as the transfer credit site, the office would benefit from the integration of external resources.
* Priority rankings are also very complicated and should be more standardized.

**Finding 2: The current system is frustrating and overwhelming, leading to mistakes.**

**2.1 There is a lot of manual work that the administrators and students must do.**

Different types of manual operations are required, starting from checking eligibility to granting permission. If the overrides are denied in the checking stage, the rejection emails are also manually sent by the administrator. All these manual steps are time consuming and mistake-prone, as administrators must collect and process a large volume of information in multiple forms and spreadsheets. As one staff member put it, “Manually inputting information into the form was ‘a nightmare’.”

**2.2: There is a lack of communication and clarification amongst professors, administrators, and students.**

As mentioned earlier, the undergraduate override system is complicated with various criteria and procedures. Therefore, administrators are frustrated because of the amount of work they have to deal with, while the faculty and students are also confused about the override criteria. P5 mentioned that “Professors and students get confused about the conditions of granting an override.” These misunderstandings often lead to more work on the administrative side. For example, instructors sometimes grant permission directly to students without considering their classroom capacity, which forces administrators to manually fix this issue. When students are not clear about the process and requirements, their override request form answers are usually missing a lot of those details, causing administrators to have to reach out to individual students for their missing information.

**2.3: The override process is redundant.**

Many of the interviewees have complained about the current system, saying that it is “outdated”, “tedious,” and “redundant.” Many different files and sites are required to complete the checking process, and all of this manual operation has caused the review process to become very time-consuming and inefficient. During our interview analysis, we found that multiple advisors often take on different roles and tasks during the override process, which can sometimes be very chaotic. We believe that this extra work shows that the process is redundant and requires further streamlining.

**Finding Implications**

* Since the current process is time-consuming and redundant, a more streamlined process is needed.
* It might be better if the division of labor could be better clarified among the UAO staff.
* Since the current override process is causing problems for both administrators and students, the system should be improved on both the student’s and administrator’s sides.

**Finding 3: The current system contains limited automation, and has a lack of integration.**

**3.1 The automation level of the current system is almost negligible.**

There are no automated features such as autofill, automatically retrieving the student data, or linking the keywords of the previous request to the present one. When a student fills out the application, each field requires manual input, instead of being based on a retrieval system that uses a unique identity from UM’s data warehouse. One UAO staff member mentioned that “[she wished] for a streamlined system with just one link on the front end, that guides [the student] through automated sections and the final output is in the form of just one worksheet which [the staff member] can look at and decide.”

**3.2 Integration is desired by all the interviewees.**

The interviewees are currently looking forward to a system which is more streamlined and updated with current technology. They currently work with numerous spreadsheets and Google Forms that are designated to specific courses. Each course request goes through the same process, but none have been automated. Furthermore, these request forms have remained separate instead of being integrated into one form. This results in multiple entries within multiple Google Forms. Moreover, there seems to be no integration with Wolverine Access, where all the override requests are finally approved or declined.

**Finding Implications**

* Automation on different levels has been overlooked for a long time, and as a result, a redundant system has been in use instead.
* The undergraduate override system is highly unorganized and unstructured.
* The workload seems to be overwhelming, and the frustration with this workload was evident during the interviews.
* Having to manually enter data causes the system to be mistake-prone.

**Finding 4: Incomplete and Insufficient information**

**4.1 Students are unaware of basic course override eligibility policy and protocols.**

Students do not have a clear idea regarding how the override requests work, and have not received proper guidance regarding when to apply and how to apply for them. These results in issues such as:

* Applying for an override request prior to the actual desired timeline. There have been cases of UAO receiving Winter semester override requests prior the beginning of Fall semester.
* Impression of a guaranteed override approval.
* Unawareness regarding the existence of a possible override based on the student’s already completed courses. Some students seem to be unaware about what prerequisites will get them into a particular course and then miss the opportunity to apply for an override.
* Sheer random requests by students, as if trying their luck on override request eligibility. P5 said it’s surprising how “students will also apply as a ‘Hail Mary,’ without possibly doing the pre-requisites.”

**4.2 Input criteria is not clear.**

The input fields on the student’s end of the forms are vague and have very unclear instructions. This results in students uploading partial or incorrect documents. Furthermore, this also means that the administrator has to contact the student, resulting in emailing back and forth so that they can retrieve more information about the student to help determine their eligibility. For example, in one form field that requests the student’s reason behind requesting an override, students often reply with vague answers such as, “Summer Transfer credit.” Answers like these show the importance of having more specific input field instructions.

**4.3 There is a lack of transparency and categorization.**

There is no system tracking for whether or not the student has used the override code given to them. Sometimes classes have reserved seats, but students can be misled because the class looks full on Wolverine access- students are unable to see reserved seats. This leads to confusion and inequity, as students won’t realize that they’re actually able to register for the course.

**4.4 Course transferability is an intricate process.**

There are multiple resources to be looked at while approving transfer credits. Every time a transfer credit document is attached, UAO needs to go through Google Forms, a transfer credit equivalency website, a faculty preference and requirement spreadsheet, a record-keeping spreadsheet and a request status spreadsheet. This is a very complicated process and takes up a lot of time on the administrator’s end.

**Finding Implications**

* A complete guide to applying for an override is not available.
* The fields for information input need to be more to the point and have an organized flow.
* Checking for transfer credit equivalency is currently a very spread out system, which causes a lot of back and forth due to referring to multiple websites to help determine a student’s override eligibility.

**Finding 5: There are features in the current system, as well as other existing technologies, that are preferred and loved by the administrators.**

**5.1 Semi-automation setup that could be reused for the undergraduate system**.

The current non-CSE Graduate override request system has resources that can be used, and has been working great so far. The programming code for achieving this level of semi-automation is already available, as this system is currently implemented.

The non-CSE Graduate override request system features are as follows:

* Well-defined student input fields. These fields have dropdown menus and autofill availability.
* Streamlining on the administrator’s end. There’s just one spreadsheet and specific details.
* Automated emails to students. Regardless of whether a student got approved, denied, or waitlisted, they are automatically notified about their override request status once a decision’s made.
* Reminder emails two weeks prior to the semester’s start date.
* Minimal financial resources required to implement this.

**5.2 Google Forms can be more of an asset when used properly.**

Google Forms are easy to set up and are convenient due to being an interactive platform. The staff was comfortable using these forms, so their only obstacle was the quantity of forms that had to be looked at. They wanted to continue with Google Forms if there was a possibility of one form with all the required fields.

**5.3 Automated emails are popular with the UAO staff and have been recommended to incorporate in the new override system.**

Automated emails with a registration code are sent via Wolverine Access solely if a request is approved. UAO is looking forward to a new system with the same automated email feature, but for both approved and denied requests.

**5.4 Slack Channels are used for specific override cases.**

The use of Slack as a platform to discuss unique cases and help maintain internal communication is a productive initiative on UAO’s part. This helps keep everyone updated in a more timely manner, while clearing out internal miscommunication due to system incompetency. It is easy to use, available to all UAO staff, and is a creative way to involve everyone.

**Finding Implications**

* The existing system actually does have some good features.
* There is most likely a possible solution that has been implemented already.
* UAO is also aware of their needs and has been looking into making new changes, while still considering their current preferences.

# 

# 

# **Recommendations**

By analyzing the problem posed by different advisors and administrative staff, as well as our synthesis from the interviews, we have come up with one main recommendation for the CSE UAO, as well as some other future steps to look into. We believe that implementing our main recommendation would greatly help the UAO with managing their course override problem.

Initially, our team tried to formulate multiple solutions to solve each of the course override problems, only to realize that these problems could potentially be solved using a system that UAO is already familiar with. That would be the non-CSE course override system, which is newer yet trustworthy, and the advisors and administrators who are familiar with it have mentioned that they would be happy to utilize a similar system. Therefore, we present the following solution, as well as our reasoning behind implementing it:

**Major Solutions: Implementing the non-CSE override system**

The non-CSE system has been successful for the following reasons:

1. **It eliminates the downsides of Google Forms.**

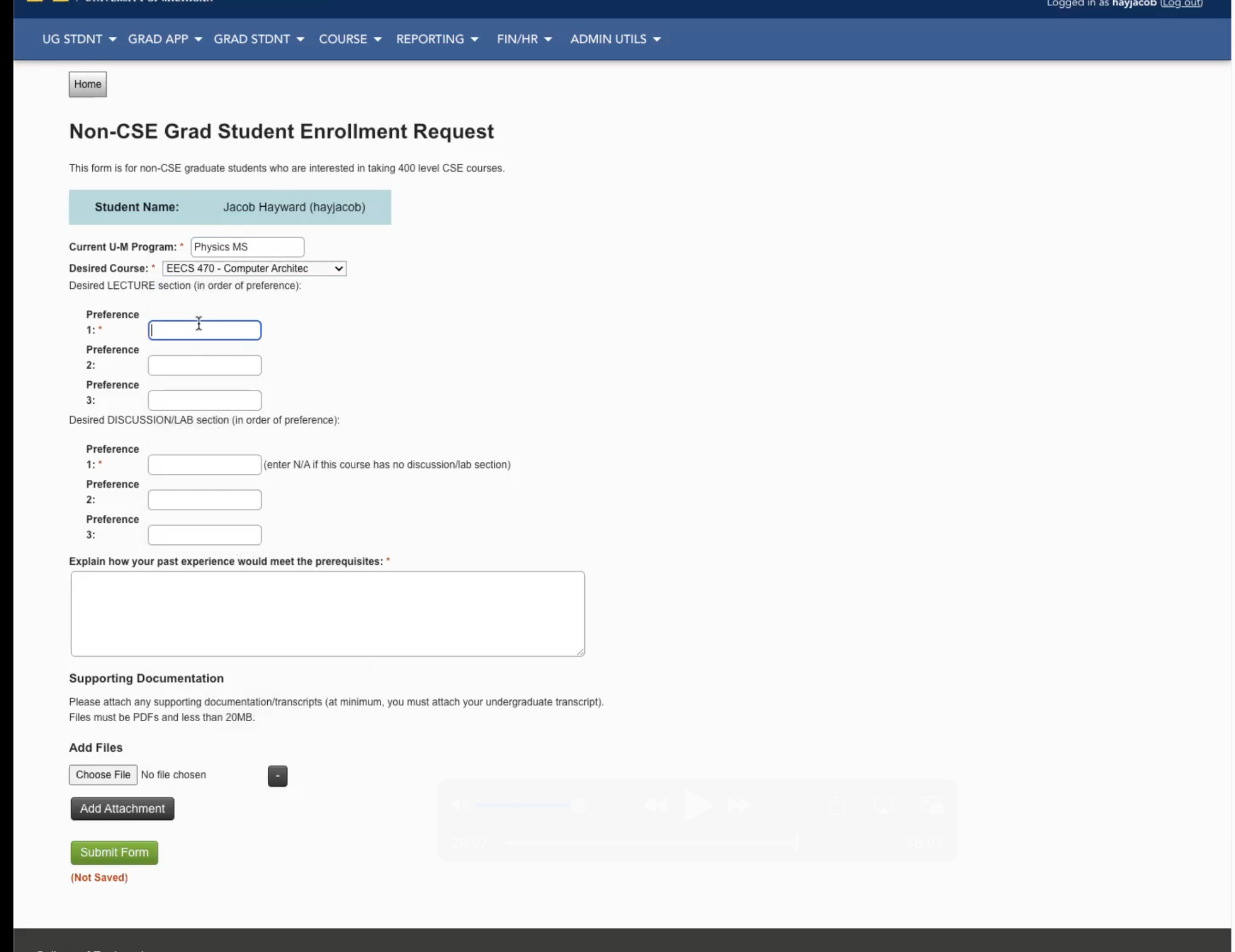
One of the major complaints UAO had was that they had too many Google Forms to keep track of. It seemed difficult to decrease the number of forms due to these forms having different formats, which caused UAO to become overwhelmed. They had to learn not just what different Google Forms contained, but also what was expected of students who came from different fields.

These Google Forms were primarily created based on different course needs. The initial plan didn’t include having multiple forms, but these multiple forms became a necessity over time. The new non-CSE system, however, was able to account for all these needs in a more central form, thus eliminating the problem of using multiple forms.

By implementing a new system based on the current non-CSE override system, this system will be able to directly organize different types of information, lessening that effort on the administrator’s part.

1. **It reduces the chances of making manual errors, due to having fewer fields to fill out.**

In the non-CSE system, the input information fields are more concise and the questions are better-framed and organized. This makes it easier for both the advisors and students to fill and understand the form, thus greatly reducing the chances of making an error.

****

Student Request Form Interface

1. **The form better clarifies what student information needs to be inputted.**

The improved clarification helps students understand what information is needed from them, which eliminates the need for any guesswork. It helps both the student, who is the one filling the form, and the administrator, who is the one reviewing the form answers later on, since it ensures both sides will be on the same page in terms of the information provided. This non-CSE system asks for important information, but it does forget to ask about details such as:

* The student’s specific reason for applying for a course override.
* For transfer credits, it doesn’t ask for necessary information such as the other school’s location (city and state), which turns out to be necessary later on in the override process, as advisors have to individually search for these schools when checking for credit equivalency.

Therefore, having a system that clearly states what is needed of the student will help eliminate:

* The constant back and forth emails between UAO and students, which can delay the override process.
* The lack of sorted requests, as students providing their reasoning for requesting an override will help UAO better determine what kind of override case this is.

1. **It helps brings a large amount of automation to the override process.**

The increased amount of automation helps multiple parties involved in the override process for the following reasons:

* Instructors are automatically notified about approving/rejecting requests once students submit their forms
* If the instructor approves a request, UAO is automatically notified, allowing them to quickly check for class capacity.
* Unlike the current CSE system, where every rejection email has to be drafted from scratch, the non-CSE system has an easier system. Override approval and rejection emails can be sent by utilizing a drop-down email menu, which has a set of personalized email templates.

****

Administrator Processing Interface

1. **The non-CSE system is on a more personalized and secure platform, compared to an external tool like Google Forms, which can also pose a potential security risk.**

Many students and work teams all over the world utilize Google Forms. Their popularity is understandable - Google Forms are not only intuitive to use, but also easy to create, modify, and collaborate with others on. They also help organizations save money, since Google Forms allows for the creation of questionnaires, as well as questionnaire response spreadsheets.

These forms, however, do come with their own set of problems. Although they are easy to share, this ease can cause a threat to security, as the IP address of any person filling and editing the form can be traced back with relative ease.

Furthermore, potentially sensitive student information such as grades and more personal information could be easily exposed. Utilizing a more robust and secure system could help prevent such possible data breaches. The non-CSE system is contained on Wolverine Access, which has Multi-Factor Authentication (MFA), a higher-strength security clearance, as opposed to Google Form, which depends on Single Sign-On (SSO), a relatively weaker security clearance.

**Other Future Steps**

1. **More integration for the administrator’s side of data management.**

One area of the override process that could be improved on is Branched File Management. This would provide the system with more clear categorization, such as moving all the winter course overrides under one area labeled “Winter Override Requests,” while fall courses would fall under another area labeled “Fall Override Requests.” This would help administrators stop having to deal with override requests that have been submitted way more in advance than they needed to be, while still setting them aside in a more organized manner.

**Integration of external and internal eligibility checking resources.**

A lot of time is wasted by needing to refer to different sites for further evaluation of student documents. This is especially true in the case of transfer courses. Here, advisors have to look at a student’s previous schools and verify the schools’ authentication, consult a transfer credit equivalency website, and further review a student’s educational documents, like transcripts. This process remains lengthy even for courses that can be waived off due to taking and clearing diagnostic tests at UM. Allowing administrators to access all of these external resources within one page would help save them more time and effort, while still providing them with instant access to all of these resources.

# **Conclusion**

While the CSE UAO genuinely tries their best to efficiently go through course override requests and get back to students as soon as possible, the sheer volume of course override requests is still enough to overwhelm the current course override system in place. As we have previously mentioned, the Non-CSE course override system has proven to be a more systematic and secure course override process, and is less difficult to implement, since a similar system could be created based on the existing framework and code. We also believe that our two further suggestions, integration for administrator-level data management and integration for internal & external eligibility checking resources would further streamline the course override process. Implementing more administrator-level data management would help the UAO staff organize the requests in a more orderly fashion, since they could categorize requests under more appropriate labels. Furthermore, being able to access built-in eligibility checkers like the transfer credit website would save time and effort for the staff since they could check these things in one place, versus having to go back and forth between multiple tabs.

We recognize that it can be difficult and time-consuming to get building such a system to be approved and implemented. Given the immense volume of course override requests that come in each semester, however, we believe that implementing a system similar to the current non-CSE one would help transform a lengthy and costly process into a much more manageable and efficient one.

# **Sources**

“Current Undergraduate Students: Computer Science and Engineering at Michigan.” *Computer Science and Engineering*, The Regents of the University of Michigan, https://cse.engin.umich.edu/academics/for-current-students/advising/.

“Enrollment Policy: Computer Science and Engineering at Michigan.” *Computer Science and Engineering*, The Regents of the University of Michigan, https://cse.engin.umich.edu/academics/for-current-students/advising/enrollment/.

“Letter, Form, and Override Requests.” *Computer Science and Engineering*, The Regents of the University of Michigan, https://cse.engin.umich.edu/academics/for-current-students/advising/letter-form-and-override-requests/.