Santa Clara University
Graduation Processing & Reporting System

MSIS 2602 Information Systems Analysis and Design
Fall 2018

Group 5:
Brian Frank
Swati Ramesh Gaikwad
Nate Marohl
Oleksandr Vozniuk

Table of Contents

Business Description and Problem Statement

Santa Clara University has an undergraduate population of approximately 5,800 students. Each academic year, two full time staff in the Office of the Registrar processes between 1,400 and 1,600 petitions to graduate. The majority of these petitions, between 1,100 and 1,200, are for students graduating at the end of spring quarter.

The petition to graduate process for spring quarter starts in late February with degrees being conferred in June. Currently students must physically walk paper forms around campus, interacting with department administrators, advisors, and department chairs, to get the necessary approvals and signatures. Once these forms have been signed, they are dropped off at the Office of the Registrar for further manual processing. A staff member reviews the forms and enter the data in multiple locations. As the quarter progresses, they continue to review the students looking for issues that would preclude them from graduating.

Occasionally, other departments, such as the Housing Department and Bursar's Office, ask for a list of students who are expected to graduate in order to facilitate their business processes. Other departments who may be interested in this data is the Offices of both the Provost and President, as well as Institutional Research, as they are reviewing the student completion rates.

System Request

Business Sponsor: The University Registrar

Business Need: This initiative has been created to streamline and automate the petition to graduate process to ultimately accelerate the approval and review processes by all key stakeholders. Currently,

- Completing the forms necessary to graduate is a manual process that needs to be automated.
- Processing and review of the graduation forms is tedious and can be expedited by eliminating the duplicate data entry performed by Office of the Registrar staff as well as developing a methodology to identify students with issues.
- Departments outside of the Office of the Registrar do not have access to this valuable information and must have it provided to them on demand.

Business Requirements: By creating an electronic graduation processing and review system, students will be able to efficiently submit the information needed for their degree to be awarded, including all approvals. This will be accomplished through the following functionality:

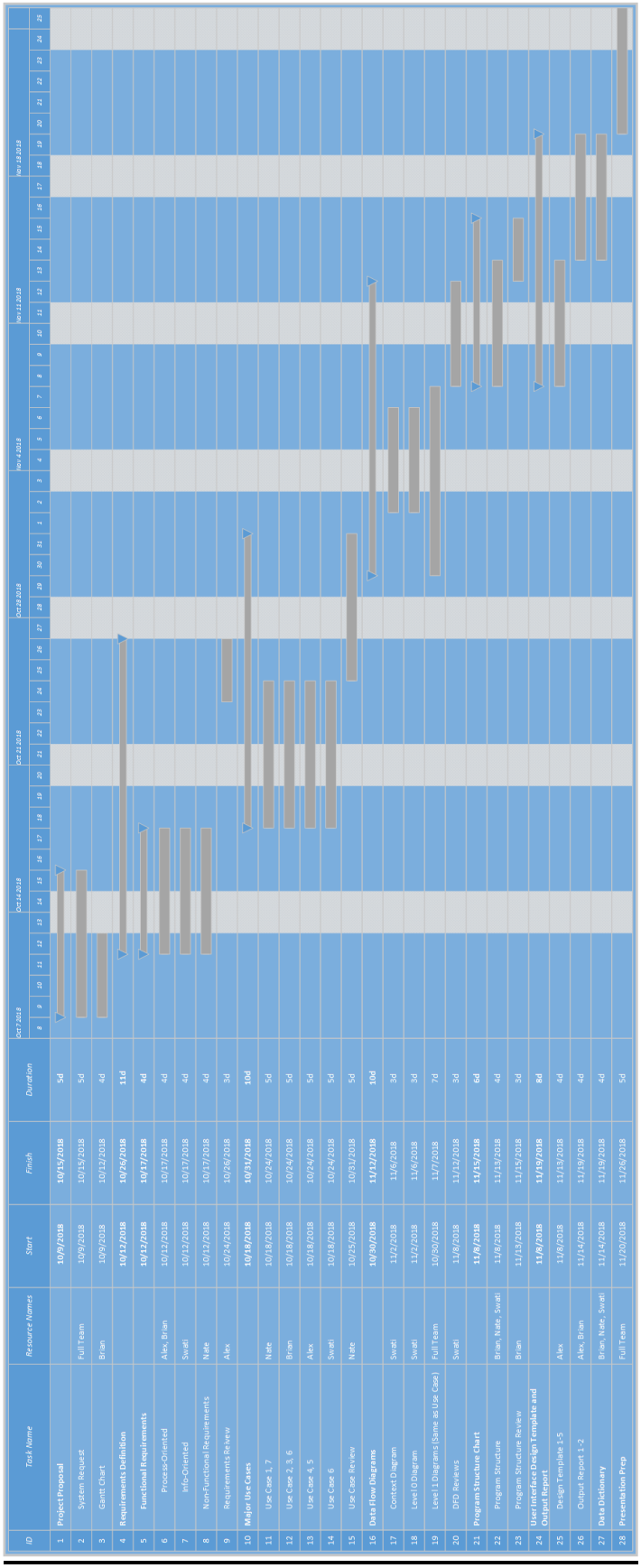
1. Identify and notify students who are eligible to graduate.
2. Assist in the submission of all information necessary for students to graduate.
3. Simplify the approval and substitution process for department administrators, advisors, and department chairs.
4. Facilitate the review and approval processes performed by the Office of the Registrar staff.
5. Support key stakeholders by providing the status of each approval process and general statistics.

Business Value: Allowing students to complete the petition to graduate via an electronic graduation processing system will have tangible value by decreasing their time and effort by 10%-25%. This will grant them more time to focus on completing senior projects, and other educational requirements with large time commitments, that are necessary to graduate. We also hope to see a 25% reduction in time for departments as they manage the approvals and a 50% reduction in processing time by the Office of the Registrar staff. These time savings will create opportunities to focus more on students who may need extra support to complete their graduation requirements, which will increase the number of students receiving a college degree at the end of four years. Removing paper from the process will expand on Santa Clara University's mission to be create a sustainable environment. Finally, this initiative will have intangible value by demonstrating to our students that they are truly attending a prestigious university.

Special Issues or Constraints: One issue that must be addressed early on is receiving support from the deans of the colleges as well as the department chairs. These changes may affect their business processes and thus we need their backing for these changes to be successful. Another issue will be how the system interfaces with our current IT system and the support we receive from the IT department. Current projects for the Registrar's department move slowly due to the pace at which IT manages them. We would like this project to be implemented as quickly as possible so that is it ready for the upcoming Spring graduation.

Scope of the System: We will limit the scope of this version to collecting the desired information for undergraduates at Santa Clara University. Information collected will include pertinent information related to graduation as well as approvals and substitutions from the appropriate individuals.

Gantt Chart



ID	Task Name	Resource Names	Start	Finish	Duration
1	**Project Proposal**		**10/9/2018**	**10/15/2018**	**5d**
2	System Request	Full Team	10/9/2018	10/15/2018	5d
3	Gantt Chart	Brian	10/9/2018	10/12/2018	4d
4	**Requirements Definition**		**10/12/2018**	**10/26/2018**	**11d**
5	**Functional Requirements**		**10/12/2018**	**10/17/2018**	**4d**
6	Process-Oriented	Alex, Brian	10/12/2018	10/17/2018	4d
7	Info-Oriented	Swati	10/12/2018	10/17/2018	4d
8	Non-Functional Requirements	Nate	10/12/2018	10/17/2018	4d
9	Requirements Review	Alex	10/24/2018	10/26/2018	3d
10	**Major Use Cases**		**10/18/2018**	**10/31/2018**	**10d**
11	Use Case 1, 7	Nate	10/18/2018	10/24/2018	5d
12	Use Case 2, 3, 6	Brian	10/18/2018	10/24/2018	5d
13	Use Case 4, 5	Alex	10/18/2018	10/24/2018	5d
14	Use Case 6	Swati	10/18/2018	10/24/2018	5d
15	Use Case Review	Nate	10/25/2018	10/31/2018	5d
16	**Data Flow Diagrams**		**10/30/2018**	**11/12/2018**	**10d**
17	Context Diagram	Swati	11/2/2018	11/6/2018	3d
18	Level 0 Diagram	Swati	11/2/2018	11/6/2018	3d
19	Level 1 Diagrams (Same as Use Case)	Full Team	10/30/2018	11/7/2018	7d
20	DFD Reviews	Swati	11/8/2018	11/12/2018	3d
21	**Program Structure Chart**		**11/8/2018**	**11/15/2018**	**6d**
22	Program Structure	Brian, Nate, Swati	11/8/2018	11/13/2018	4d
23	Program Structure Review	Brian	11/13/2018	11/15/2018	3d
24	**User Interface Design Template and Output Report**		**11/8/2018**	**11/19/2018**	**8d**
25	Design Template 1-5	Alex	11/8/2018	11/13/2018	4d
26	Output Report 1-2	Alex, Brian	11/14/2018	11/19/2018	4d
27	Data Dictionary	Brian, Nate, Swati	11/14/2018	11/19/2018	4d
28	Presentation Prep	Full Team	11/20/2018	11/26/2018	5d

Functional Requirements - Process

1. **Students Ready to Graduate:**
    1.1. The system should be able to check all students and identify those that are ready to start graduation process.
    1.2. The system should mark identified students as eligible to graduate.
    1.3. The system must send an email notification to the student with general information about the graduation process and a link to begin the petition process.
    1.4. The system should send a reminder email if there was no actions taken by the student after 2 weeks, 4 weeks, 6 weeks, 8 weeks.
    1.5. The system will notify the Office of the Registrar Staff of all students who have not petitioned to graduate 1 week prior to the start of classes for their expected graduation term.
2. **Submission of Graduation Materials:**
    2.1. The student must log in through a secure single sign on portal.
    2.2. The student must be able to submit their diploma name and address. The system should only allow the student to submit their diploma name once. Also , the system will notify the student that If the diploma name needs to be changed, the student must contact the Office of the Registrar.
    2.3. The system should pull current majors and minors for the student to review. The student must be able to add majors or minors.
    2.4. The system should pull the current course history and run the student's degree audit.
    2.5. The system should clearly identify incomplete requirements.
    2.6. The system will show courses and future availability that will fulfill incomplete requirements.
    2.7. The student will be able to develop a course plan and notes based on course availability.
    2.8. The system will notify respective advisor and department of submission.
    2.9. The system will display a final review page that the student must approve.
    2.10. The system will send confirmation email to the student after they have completed the submission process.
    2.11. The system will update the database to show that the student has completed the submission process
    2.12. The student should be able to update their course plan once they have registered for classes.
3. **Review and Approval by Key Stakeholders: Advisors, Department Chairs and Office of the Registrar Staff (OOTR Staff):**
    3.1. The system will allow the advisor, department chair and OOTR staff to each complete a review of the student.
    3.2. The advisor, department chair, and OOTR staff must sign on through a secure single sign on portal.
    3.3. The advisor, department chair, and OOTR staff will be presented a list of students who need to be approved.
    3.4. The system should pull evaluation information submitted by the student for the advisor, department chair, or OOTR staff.
    3.5. The advisor, department chair, and OOTR staff should be able to run student's degree audit.
    3.6. The advisor, department chair, and OOTR staff should be able to comment and verify course plan and notes for unfulfilled requirements.

3.7. The department chair should be able to provide the final approval for course substitutions.
3.8. The OOTR staff should be able to process approved substitutions
3.9. The advisor, department chair, and OOTR staff should be able to mark the petition as approved or pending.
3.10. The advisor, department chair, and OOTR staff should be able to email the student regarding any issues regarding the student's course plan, notes and outstanding requirements.
3.11. The system should send notification emails to the student and the department chair after the advisor has submitted the approval.
3.12. The system should send notification emails to the student and the OOTR staff when the department chair has submitted the approval.
3.13. The system should send a notification email to the student when the OOTR staff has submitted the approval.
3.14. The system will notify OOTR Staff weekly after first review if the student still has outstanding requirements or if the student was previously approved but is now missing a requirement.

4. Status and Statistics for Key Stakeholders:

4.1. Key stakeholders should be able to enter criteria to see a list of students and the status in the petition process.

Functional Requirements - Information

1. The system must store all the student's information entered during the petition process.
2. The system must be able to store the student's name in foreign accents.
3. The system must be able to access student's course history, majors and minors and upcoming class schedules.
4. The system should store approval information/comments submitted by respective department advisors and chairs.
5. The system should maintain data about outstanding requirements.
6. The system must include a petition eligibility criteria determined by the Office of the Registrar.
7. The system must retain student's petition history at least for previous five years.
8. The system should hold email templates for each type of email notification.
9. The student should be able to see the status of their petition.

Non-Functional Requirements

Operational:

1. The system should be able to run on handheld devices.
2. The system should connect with existing databases from other departments.
3. The system should be able to send emails to Santa Clara addresses without being blocked.
4. The system must be able to accommodate new students/classes/requirements from each department.
5. The system should be able to run on any web browser.
6. All office locations always have network connection permitting real-time database updates.
7. The system must be able to print any pages containing a final submission report, course plans, or notes.

Performance:

1. The system should be available for use 24 hours per day, 365 days per year.
2. Maximum of 1500 simultaneous users at peak use times.
3. The system should automatically save the student's graduation request every 10 seconds as they complete the forms in case of crashes.
4. The system should have a 99% uptime performance.
5. In case of system crash, the system should be back online within 24 hours.
6. Network transaction response time <=5 seconds.

Security:

1. Department Chairs can only see student's who have petitioned for a major in their departments, Advisors can only see petitions for their advisees.
2. A student petitioning for graduation can only see their request.
3. The system should allow employees of the Office of the Registrar to view the contact information and request of the student, as well as the contact information of the approvers.
4. The system will require all users to sign in with a unique password.
5. All passwords in the system should be encrypted and secured.
6. Advisors and Department Chairs must be able to approve the petition in a manner that does not allow another unauthorized individual to approve petitions.
7. The system will comply with all FERPA regulations.
8. The system will have the most updated virus and malware detection software available.

Cultural & Political:

1. The system should allow users to select in what language they would like to see their graduation request.
2. The system should allow students to use foreign accents in their name.
3. Each department is allowed to continue their own process, as long as all information and approvals are eventually entered into the system.
4. The system should display the school logo on each output from the system.

Use Cases

The major actors interacting with our system consist of the Student, Advisors, Department Chairs and Office of the Registrar Staff. The use cases chosen show how each of these actors help move the process forward by completing their steps.

Use Case 2 demonstrates how a student enters their diploma name and address. This is followed by Use Case 3, which describes how the system helps the student choose classes to fulfill their outstanding requirements. The advisor, department chair and Office of the Registrar staff complete their reviews by using Use Cases 4, 5 and 6. These three use cases are very similar, with only minor differences, but a separated because the actions take place in a sequential manner. Use Cases 1 and 7 document processes that the system must do to start and continue the overall graduation process. Both use cases are controlled by time triggers.

Use Case 1:

Use Case Name: Notification of Eligibility	**ID:** UC-1	**Priority:** High
Brief Description: This describes how the system detects that a student needs to submit a petition to graduate and notifies the student		
Actor: System		
Trigger: It is 8am on the second week of school **Type External Temporal**		
Preconditions: 1. Student is an active in the academic system. **2.** The biographic and academic datastores is up to date.		

Normal Course	**Information for Steps**
1. The system pulls the student's bio/academic data from the Biographic and Academic Info datastore	<=Academic Data
2. The system checks if all requirements are met and updates the student's profile in the petition datastore with a petition to graduate approval (ie student is marked as eligible)	=>Petition to Graduate Approval
3. The system pulls the student's contact information from the Biographic Info datastore	<=Contact Info
4. The system sends an email to the student notifying them that they need to petition to graduate	=>Petition to Graduate Email
5. The system provides the time at which the email was sent to the petition datastore	=>Time of Email
6. Every two weeks from the time of email, the system will verify if the petition datastore has received a student account trigger (noted in Process 2)	<=Student Account Trigger
7. If no petition process approval is present the system repeats steps 4 and 5	<=Contact Info =>Petition to Graduate Email

Alternative Course(s):

A1. The system does not identify a petition process approval in the petition datastore after 6 weeks of the time of email (**Branch at Step 8**)

1. If no petition process approval is present the system repeats steps 4 and 5
2. The system includes that student's contact info in an email to the Record Analyst
3. The Record Analyst contacts the student by phone to notify them of the requirement to submit a petition to graduate

Post Conditions:

1. The system has validated that the student has qualified to submit a petition to graduate
2. The system has updated the Bio/Academic Info datastore to denote the student is qualified to submit a petition to graduate
3. The student has attended for one week the classes that would necessary to require a petition to graduate
4. The system has sent an email to the student notifying them that they need to submit a petition to graduate

Exceptions:

E1. The student has not reached the necessary requirements to begin the petition process (**Branch at Step 3**)

1. The system identifies that the student in question does not meet the necessary requirements to petition to graduate
2. No other steps in the process are performed, system moves on to the next student

Summary:

Inputs	Source	Outputs	Destination
Academic Data	Academic Datastore	Petition to Graduate Approval	Petition Datastore
Contact Info	Biographic Info Datastore	Petition to Graduate Email	Student
Student Account Trigger	Petition Datastore	Time of Email	Petition Datastore

Use Case 2:

Use Case Name: Student Submission of Information	**ID:** UC-2	**Priority:**High
Brief Description: This describes the process a student must take to submit their information that the system uses to process their petition to graduate		
Actor: Student		
Trigger: Student has decided it is time to complete the petition to graduate. **Type External Temporal**		
Preconditions: 1. The student has been marked as eligible in the Petition datastore. 2. Student has received an email with petition to graduate information. 3. The biographic and academic datastores are online and up to date.		

Normal Course	**Information for Steps**
1. The student logs on through a secure single sign on portal.	<=Student Credentials
2. The system verifies that the student is eligible to petition to graduate in the petition datastore (via the petition to graduate approval).	<=Petition to Graduate Approval
3. The system provides a student account trigger to the Petition datastore to identify that the student has started their petition to graduate.	=>Student Account Trigger
4. The system pulls biographic and academic data for the student	<=Bio/Academic Data
5. The system displays an acknowledgement page for the student to approve.	<=Acknowledgement Approval, =>Validated Ack. Approval
6. The system presents the diploma name page, which is pre-populated with the student's primary name, for the student to edit and approve.	<=Updated Diploma Name, =>Validated Diploma Name
7. The system presents the diploma address, which is pre-populated with the student's preferred address, for the student to edit and approve.	<=Updated Diploma Address, =>Validated Diploma Address
8. The system presents all majors and minors, which is pre-populated from the student's academic record, for the student to approve.	<=Updated Majors/Minors, =>Validated Majors/Minors
9. The system presents the quarter in which the student is expected to graduate, which is pre-populated from the student's academic record, for the student to edit and approve.	<=Updated Grad Term, =>Validated Grad Term
10. The system displays all information gathered for final approval by the student.	<=Final Approval, =>Validated Final Approval

Alternative Course(s):

A1. The student needs to edit their majors or minors (**Branch at Step 8**)

1. The student is taken to another page which allows them to add, drop or switch their majors and minors.
2. ADD:
   a. Requests to add a major or minor are sent to the department of the added major or minor for approval.
   b. The major or minor is added to their record with a note pending approval from department.
3. DROP:
   a. Requests to drop a major or minor are done automatically.
4. CHANGE:
   a. Requests to change a major to minor (or minor to major) are sent to the department for approval.
   b. The major or minor is added to their record with a note pending approval from department.
5. SWITCH:
   a. Requests to switch a first and second major are done automatically.
6. (Follows completion of any path above) Student is returned to the normal course step 8.

Post Conditions:

1. The student has submitted their diploma name.
2. The student has submitted their diploma address.
3. The student has submitted their majors and minors.
4. The biographic, academic and petitioning datastores are up to date.

Exceptions:

E1. The system cannot verify the student is eligible to petition to graduate (**Branch at Step 2**)

1. The system displays the follow message: "You are not eligible to petition to graduate. Please contact the Office of the Registrar for more information."
2. The system terminates the use case.

Summary: Inputs	**Source**	**Outputs**	**Destination**
Student Credentials	Student	Student Account Trigger	Petition Datastore
Petition to Graduate Approval	Petition Datastore	Validated Ack. Approval	Petition Datastore
Bio/Academic Data	Bio/Academic Datastore	Validated Diploma Name	Biographic Datastore
Acknowledgement Approval	Student	Validated Diploma Address	Biographic Datastore
Updated Diploma Name	Student	Validated Majors/Minors	Academic Datastore
Updated Diploma Address	Student	Validated Grad Term	Academic Datastore
Updated Majors/Minors	Student	Validated Final Approval	Petition Datastore
Updated Grad Term	Student		
Final Approval	Student		

Use Case 3:

Use Case Name: Degree Audit Check and Course Planning	**ID:** UC-3	**Priority:**High
Brief Description: This describes how the system runs the degree audit, shows unsatisfied requirements, shows courses to fulfill requirements, and notifies advisors and departments		
Actor: System		
Trigger: Student has completed the final approval from UC-2 **Type External Temporal**		
Preconditions: 1. The student has submitted their diploma name. 2. The student has submitted their diploma address. 3. The student has submitted their majors and minors. 4. The academic datastores is up to date.		

Normal Course	**Information for Steps**
1. The system runs a degree audit, based on the updated academic datastore, and identifies all outstanding requirements.	<=Academic Data
2. The system displays each outstanding requirement with classes that will satisfy the requirement, and their next availability. (Repeats through each requirement)	=>Available Classes
3. The student will select the desired course and the quarter they plan to take it. a. As the system repeats through step 3, the system will display all classes selected and the planned quarter at the top of the page for reference.	<=Class and Quarter
4. The student will add any notes for their advisor and department and will submit their final plan.	<=Course Plan/Notes
5. The system updates the petition datastore to show that the student has completed the course plan/notes.	=>Course Plan/Notes Completion
6. The system will update the academic datastore with course plan and notes.	=>Verified Course Plan/Notes
7. The system will send a confirmation email, including all submitted information, to the student.	=>Confirmation Email
8. The system will send a notification email to the student's advisor and department	=>Advisor Notification Email

Alternative Course(s):

A1. The system does not find any available classes for the student to take (**Branch at Step 3**).

1. The student will add notes detailing how the requirement will be fulfilled (return to normal course step 3).

A2. The student does not have any outstanding requirements (**Branch at Step 2**).

1. The system will notify the student they do not have any outstanding requirements (return to normal course step 6).

Post Conditions:

1. Student has submitted a course plan and notes.
2. System has sent a confirmation email to the student
3. System has sent a notification email to the student's advisor.
4. The academic datastores is up to date.

Exceptions: N/A

Summary:

Inputs	**Source**	**Outputs**	**Destination**
Academic Data	Academic Datastore	Available Classes	Student
Classes/Quarter	Student	Course Plan/Notes Completion	Petition Datastore
Course Plan/Notes	Student	Verified Course Plan/Notes	Petition Datastore
		Confirmation Email	Student
		Advisor Notification Email	Advisor/Department

Use Case 4:

Use Case Name: Review by Advisor	**ID:** UC-4	**Priority:**High
Brief Description: This describes how the advisor reviews and approves the pre-graduation evaluation information sent by the student		
Actor: Advisor		
Trigger: Advisor wants to review and approve student`s pre-graduation evaluation information. **Type External Temporal**		
Preconditions: 1. Student has completed their pre-graduation evaluation form in the system. 2. Advisor has received a notification email. 3. The academic and petition datastores are up to date.		

Normal Course	**Information for Steps**
1. The advisor signs on through a secure single sign on the portal.	<=Advisor Credentials
2. The system pulls the list of students that have completed their evaluation information from the datastore.	<=Petitioning Student List
3. The system displays a list of students who need to be approved for the advisor.	=>Displayed Petitioning Student List
4. The advisor chooses the student to review.	
5. The system pulls the course plan/notes for the chosen student.	<=Saved Course Plan/Notes
6. The system displays the course plan/notes submitted by the student for advisor.	=>Displayed Course Plan/Notes
7. The system pulls the student's academic data which is presented to the advisor as a degree audit.	<=Academic Data, =>Displayed Academic Data
8. The advisor compares the course plan/notes to the degree audit, provides necessary substitutions, and approves the course plan for the student.	<=Advisor Approval
9. The system provides the saved advisor approval to the petition datastore	=>Saved Advisor Approval
10. The system notifies the department chair	=>Department Notification Email or =>Student Notification Email (A1)
11. The system notifies the student that the advisor has completed the review.	=>Completed Review Email

Alternative Course(s):

A1. Advisor identifies outstanding requirements **(Branch at Step 8)**

1. The advisor identifies that there are outstanding requirements to be fulfilled by student.
2. The advisor submits substitutions for unfulfilled requirements.
3. The advisor provides additional comments about outstanding requirements.
4. The advisor changes application status to "incomplete" via the advisor approval
5. The system sends an email to the student.

Post Conditions:

1. The advisor has updated the status to "Approved".
2. The advisor has identified outstanding requirements and substitutions, and provided additional comments.
3. The department chair has been notified of the application status of the student's petition by an email.
4. The student can view the application status of their petition online

Exceptions: N/A

Summary:

Inputs	**Source**	**Outputs**	**Destination**
Advisor Credentials	Advisor	Displayed Petitioning Student List	Advisor
Petitioning Student List	Petition Datastore	Displayed Course Plan/Notes	Advisor
Saved Course Plan/Notes	Petition Datastore	Displayed Academic Data	Advisor
Academic Data	Academic Datastore	Saved Advisor Approval	Petition Datastore
Advisor Approval	Advisor	Department Approval Notification	Department Chair
		Student Notification Email	Student
		Completed Review Email	Student

Use Case 5:

Use Case Name: Review by Department Chair	**ID:** UC-5	**Priority:**High
Brief Description: This describes how the department chair reviews and approves the pre-graduation evaluation information sent by the student and also approves substitutions.		
Actor: Department Chair		
Trigger: Department Chair wants to review and approve student`s pre-graduation evaluation information and substitutions. **Type External Temporal**		

Preconditions:

1. Student has completed their pre-graduation evaluation form in the system.
2. Advisor has approved the student`s course plan and substitutions
3. The academic and petition datastores are up to date.
4. Department chair has received a notification email.

Normal Course	**Information for Steps**
1. The department chair signs on through a secure single sign on the portal.	<=Department Chair Credentials
2. The system pulls the list of students that have completed their evaluation information from the datastore.	<=Petitioning Student List
3. The system displays a list of students who need to be approved for the advisor.	=>Displayed Petitioning Student List
4. The department chair chooses the student to review.	
5. The system pulls the course plan/notes for the chosen student.	<=Saved Course Plan/Notes
6. The system displays the course plan/notes submitted by the student for advisor.	=>Displayed Course Plan/Notes
7. The system pulls the student's academic data which is presented to the advisor as a degree audit.	<=Academic Data, =>Displayed Academic Data
8. The department chair compares the course plan/notes to the degree audit and approves the course plan and substitutions for the student.	<=Department Chair Approval
9. The system provides the saved department chair approval to the petition datastore	=>Saved Department Chair Approval
10. The system notifies the OOTR staff	=>OOTR Notification Email or =>Student Notification Email (A1)
11. The system notifies the student that the advisor has completed the review.	=>Completed Review Email

Alternative Course(s):

A1. Advisor identifies outstanding requirements **(Branch at Step 8)**

1. The department chair identifies that there are outstanding requirements to be fulfilled by student.
2. The department chair submits substitutions for unfulfilled requirements.
3. The department chair provides additional comments about outstanding requirements.
4. The department chair changes application status to "incomplete" via the advisor approval
5. The system sends an email to the student.

Post Conditions:

1. The department chair has updated the status to "Approved".
2. The department chair has identified outstanding requirements and substitutions, and provided additional comments.
3. The OOTR staff has been notified of the application status of the student's petition by an email.
4. The student can view the application status of their petition online

Exceptions: N/A

Summary:

Inputs	Source	Outputs	Destination
Department Chair Credentials	Department Chair	Displayed Petitioning Student List	Department Chair
Petitioning Student List	Petition Datastore	Displayed Course Plan/Notes	Department Chair
Saved Course Plan/Notes	Petition Datastore	Displayed Academic Data	Department Chair
Academic Data	Academic Datastore	Saved Department Chair Approval	Petition Datastore
Department Chair Approval	Department Chair	OOTR Approval Notification	OOTR Staff
		Student Notification Email	Student
		Completed Review Email	Student

Use Case 6:

Use Case Name: Review by OOTR Staff	**ID:** UC-6	**Priority:**High
Brief Description: This describes how the system notifies status of application to OOTR staff and OOTR staff updates the status after review**.**		
Actor: OOTR Staff		
Trigger: OOTR staff wants to review and approve graduation petition. **Type External Temporal**		
Preconditions: 1. Student has completed their pre-graduation evaluation form in the system. 2. Advisor and department chair has approved the student`s course plan and substitutions 3. The academic and petition datastores are up to date. 4. OOTR staff has received a notification email.		

Normal Course	**Information for Steps**
1. The OOTR staff signs on through a secure single sign on the portal.	<=OOTR Credentials
2. The system pulls the list of students that have completed their evaluation information from the datastore.	<=Petitioning Student List
3. The system displays a list of students who need to be approved for the OOTR staff.	=>Displayed Petitioning Student List
4. The OOTR staff chooses the student to review.	
5. The system pulls the course plan/notes for the chosen student.	<=Saved Course Plan/Notes
6. The system displays the course plan/notes submitted by the student for OOTR staff.	=>Displayed Course Plan/Notes
7. The system pulls the student's academic data which is presented to the OOTR staff as a degree audit.	<=Academic Data, =>Displayed Academic Data
8. The OOTR staff enters substitutions approved by the department chair into the academic datastore.	<=Substitutions
9. The OOTR staff compares the course plan/notes to the degree audit and approves the course plan for the student.	<=OOTR Approval
10. The system provides the saved department chair approval to the petition datastore	=>Saved OOTR Staff Approval
11. The system notifies the student that the OOTR staff has completed the review.	=>Student Notification Email

Alternative Course(s):

A1. Advisor identifies outstanding requirements **(Branch at Step 9)**

1. The department chair identifies that there are outstanding requirements to be fulfilled by student.
2. The department chair submits substitutions for unfulfilled requirements.
3. The department chair provides additional comments about outstanding requirements.
4. The department chair changes application status to "incomplete" via the advisor approval
5. The system sends an email to the student.

Post Conditions:

1. The OOTR staff has updated the status to "Approved".
2. The OOTR staff has identified outstanding requirements and substitutions, and provided additional comments.
3. The student has been notified of the application status of the student's petition by an email.
4. The student can view the application status of their petition online

Exceptions:

E1. Student form information and audit results do not match.

1. OOTR comments about the mismatch in the comments section.
2. The system sends an email to student about comments on his petition application.
3. The System terminates the use case.

Summary:

Inputs	Source	Outputs	Destination
OOTR Credentials	OOTR Staff	Displayed Petitioning Student List	OOTR Staff
Petitioning Student List	Petition Datastore	Displayed Course Plan/Notes	OOTR Staff
Saved Course Plan/Notes	Petition Datastore	Displayed Academic Data	OOTR Staff
Academic Data	Academic Datastore	Saved OOTR Staff Approval	Petition Datastore
Substitutions	Academic Datastore	Student Notification Email	Student
OOTR Approval	OOTR Staff		

Use Case 7:

Use Case Name: Ongoing review of Students	**ID:** UC-7	**Priority:**High
Brief Description: This describes how the OOTR Staff are notified and up date students who have not resolved issues on their petition to graduate.		
Actor: System		
Trigger: It is 6pm local time on Sunday (runs weekly). **Type External Temporal**		
Preconditions: 1. The OOTR staff has updated status to either "Approved" or "Incomplete". 2. OOTR staff has identified outstanding requirements if any and commented about the same. 3. Student has been notified about his/her application status by an email.		
Normal Course 1. The system pulls a list of student who have petitioned to graduate for the current quarter and their petition status. a. If the student was marked approved or pending less than 7 days prior to the current date, the student will be skipped. 2. The system runs a degree audit for each student, based on the updated academic datastore, and identifies all outstanding requirements. 3. If the student's petition status was approved, but an outstanding requirement was found in the degree audit, the system will change the petition status to pending. 4. The system will send a list of all students with a pending status (including a list of outstanding requirements) to the OOTR Staff by 8am Monday morning. 5. The OOTR Staff will follow Use Case 5 for each student.	**Information for Steps** <=Petitioning Student List <=Academic Data =>Updated Petition Status =>Pending Status List	
Alternative Course(s): A1. Differing Student Statuses and outstanding requirements.: 1. (a) If the student's petition status was approved, and no outstanding requirements were found, the status would stay as approved. 1. (b) If the student's petition status was pending, the system will not make any changes to the status regardless of whether the student has any outstanding requirements.		
Post conditions: 1. The system has identified students with outstanding requirements who have active petitions to graduate 2. The system has sent an email to the OOTR Staff with a list of all pending students. 3. The OOTR Staff have followed Use Case 7 for each student.		
Exceptions: N/A		
Summary:		

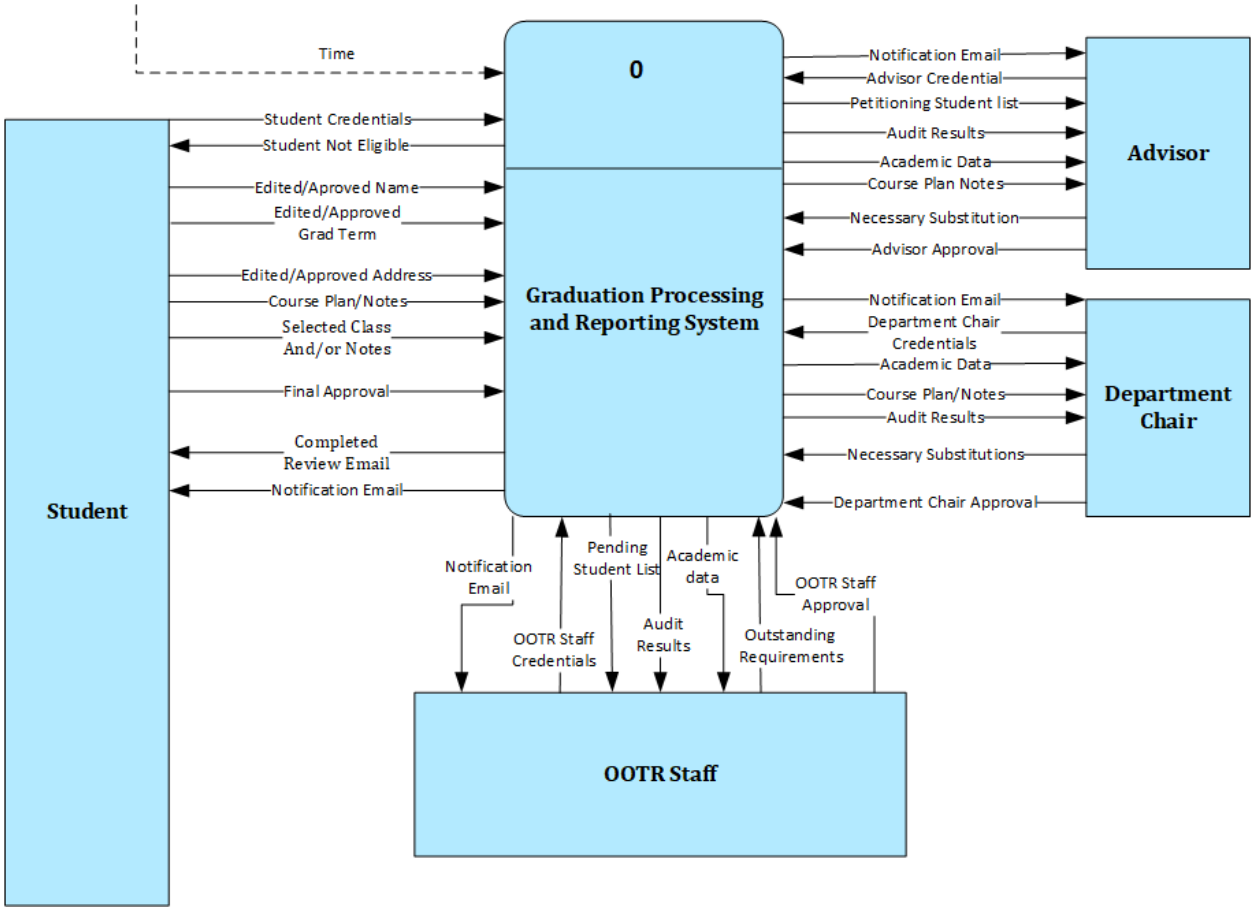
Inputs	**Source**	**Outputs**	**Destination**
Petitioning Student List Academic Data	Petition Datastore Academic Datastore	System Generated Approval Pending Status List	Petition Datastore OOTR Staff

Data Flow Diagrams

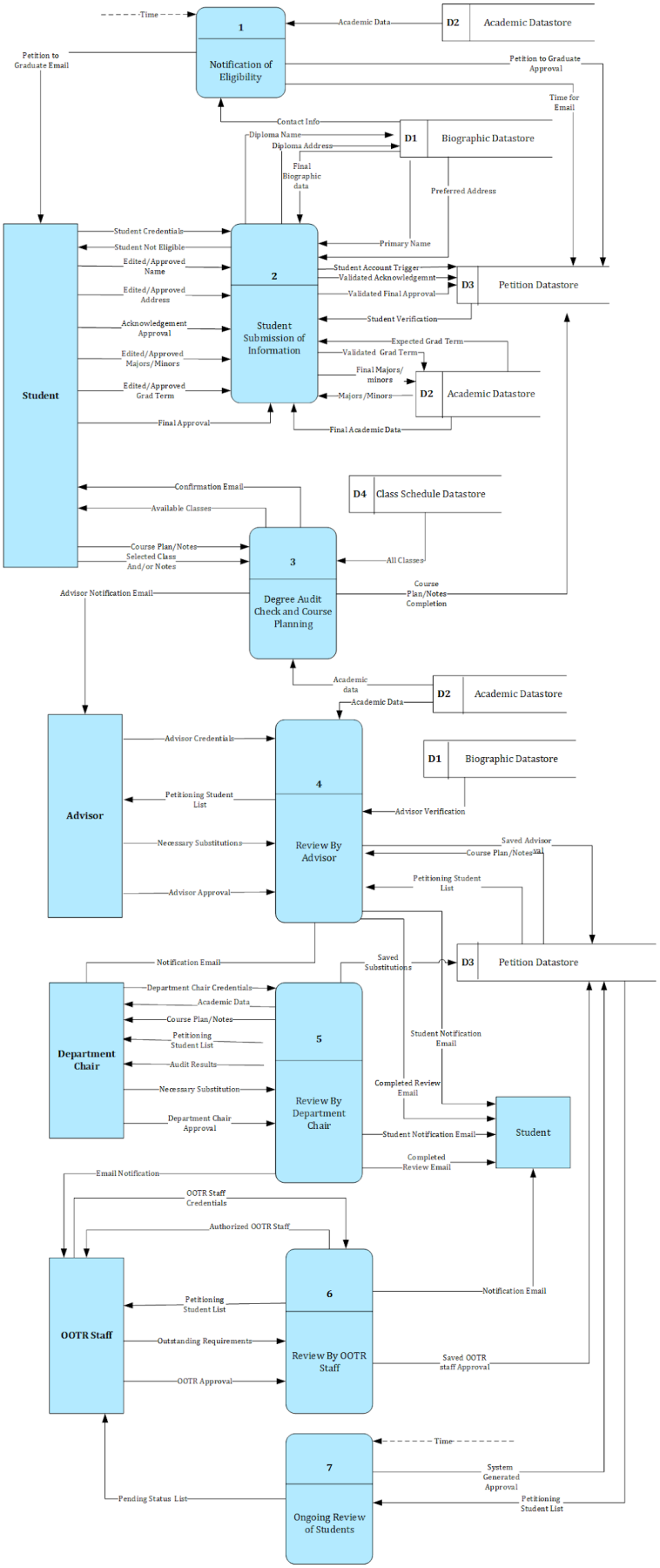
The most important data flowing through the system is the student's academic data, outstanding requirements and the course plan and notes. Other important data is the student's diploma name and address.

Student academic data is passed back and forth throughout the entire process. It is mainly used when a degree audit needs to be run, which happens in almost every phase of the process. Outstanding requirements are also a necessary piece of data as this is what is preventing a student from graduating. Course plan and notes show the student's current method for completing their outstanding requirements and must be shown to every actor in the process. Student's might say that the diploma name and address are the most important pieces of data because their diploma needs to be mailed to the correct address so it can be framed to hang on their wall.

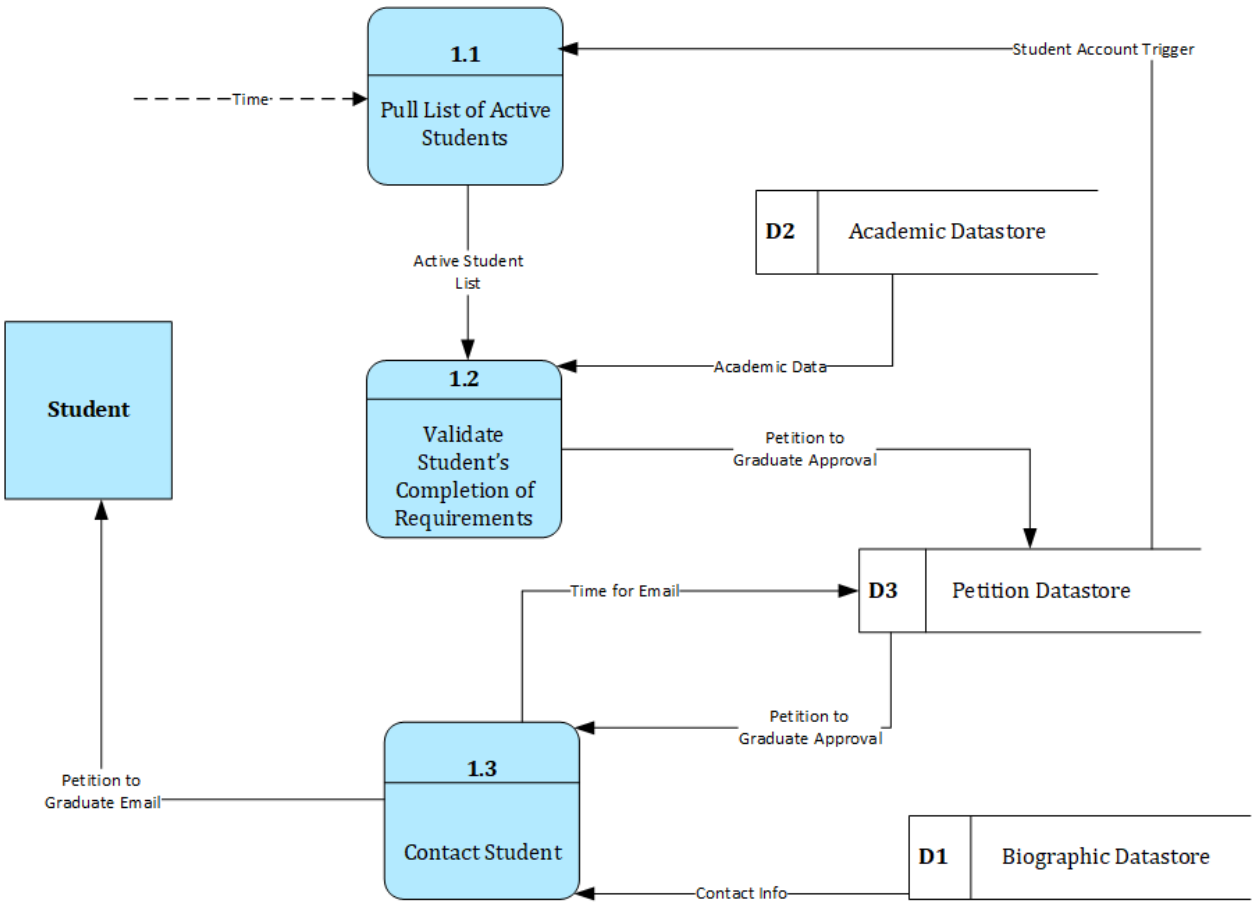
Context Diagram:



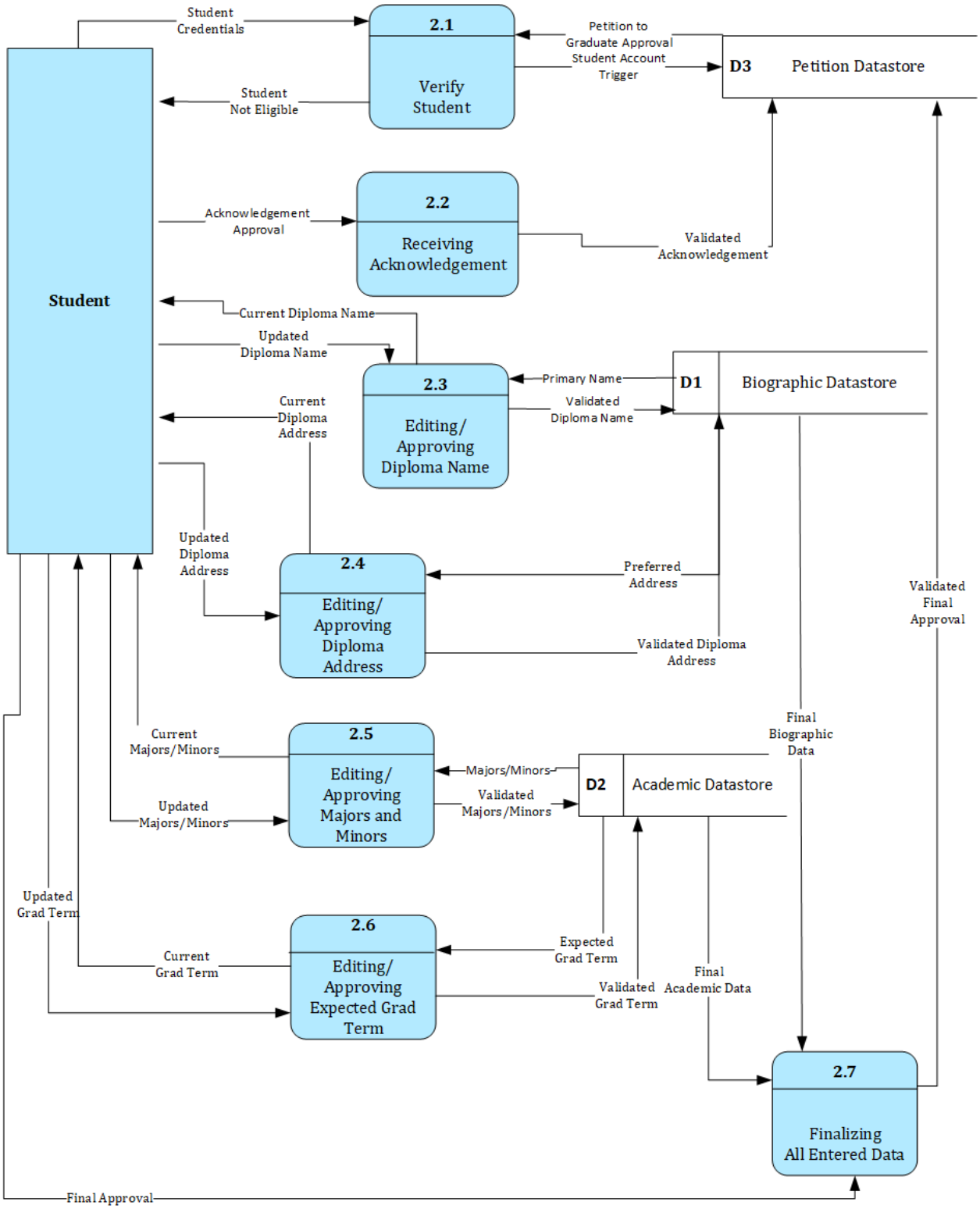
Level 0 Diagram:



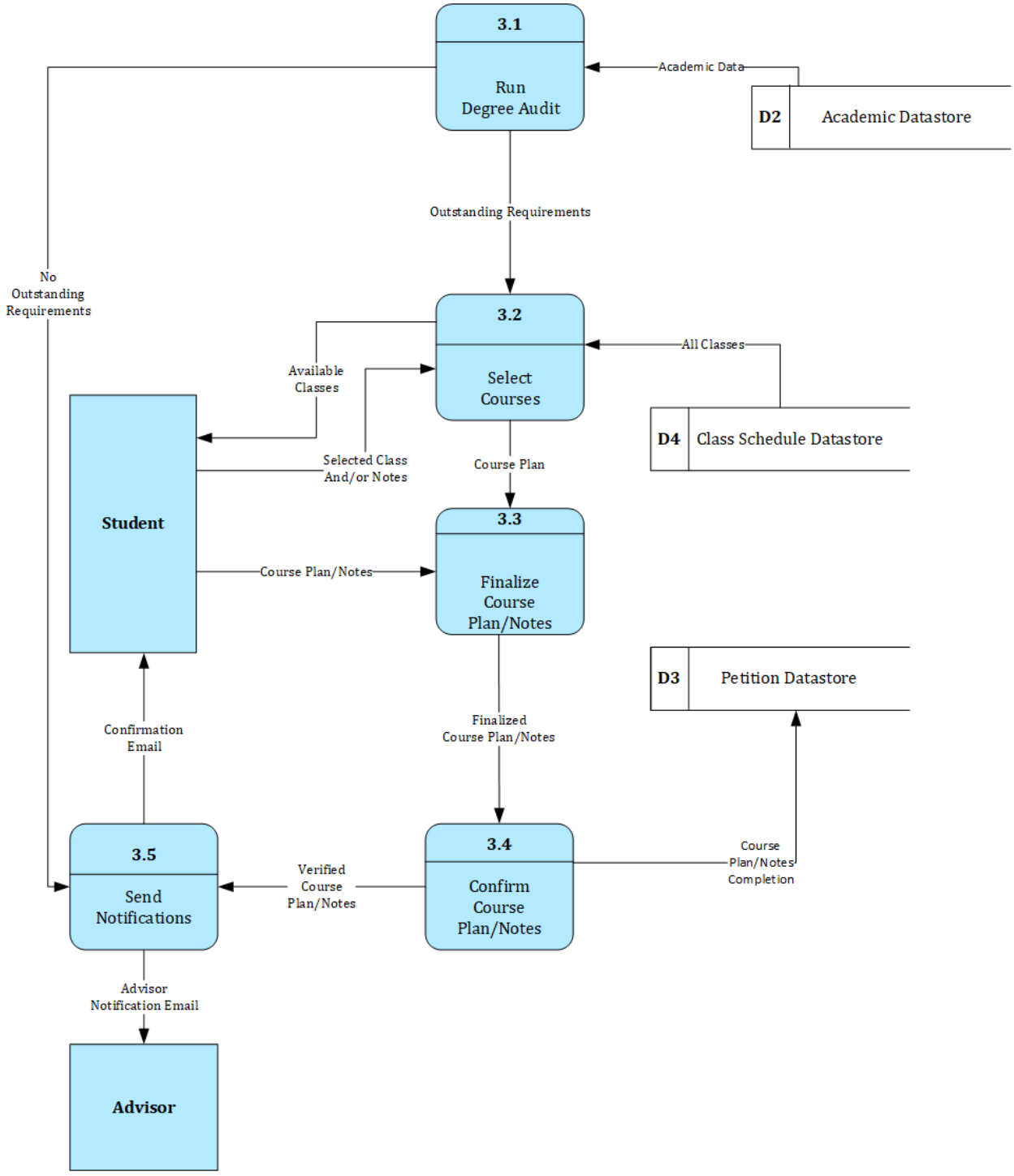
Level 1 Diagram for Process 1:



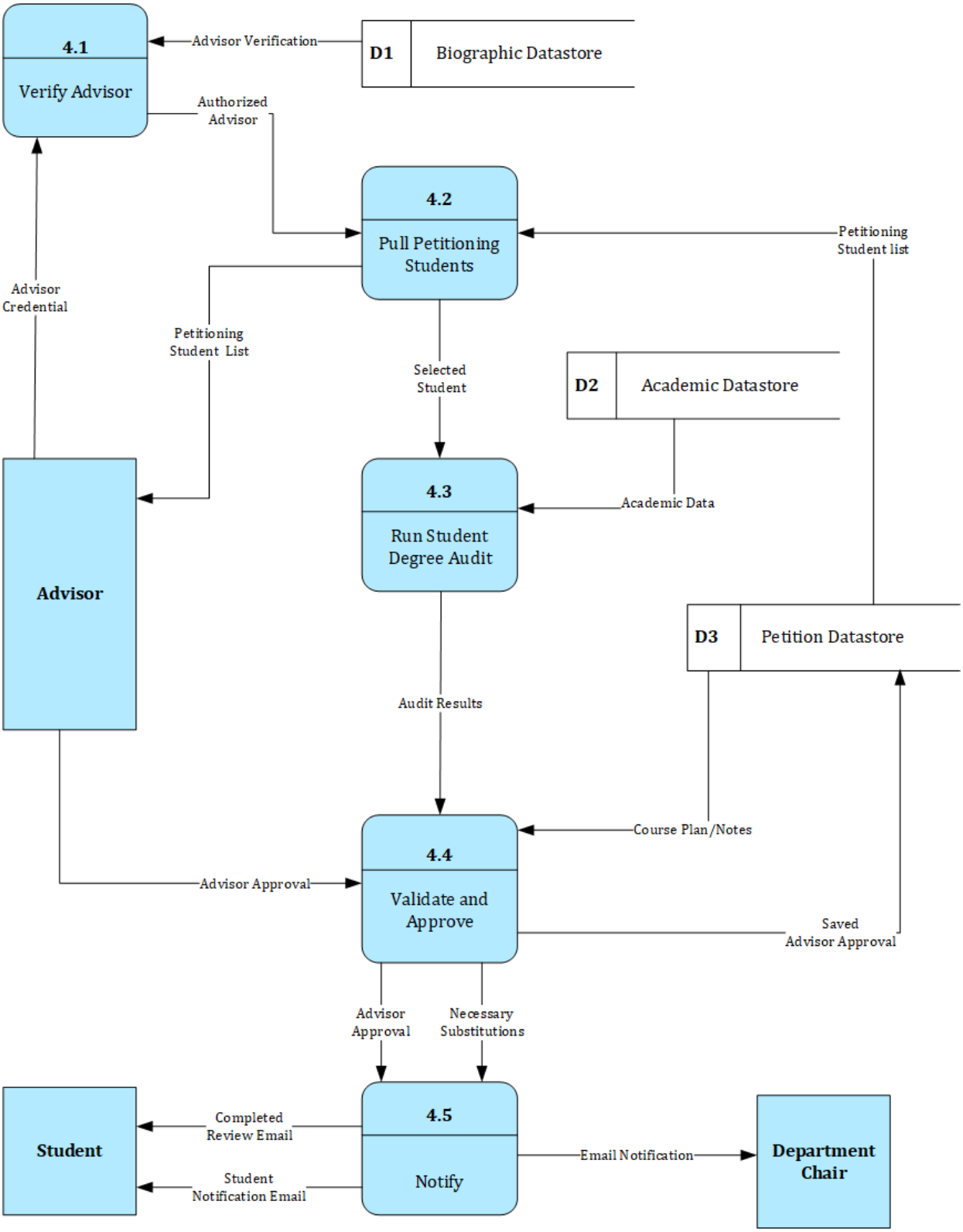
Level 1 Diagram for Process 2:



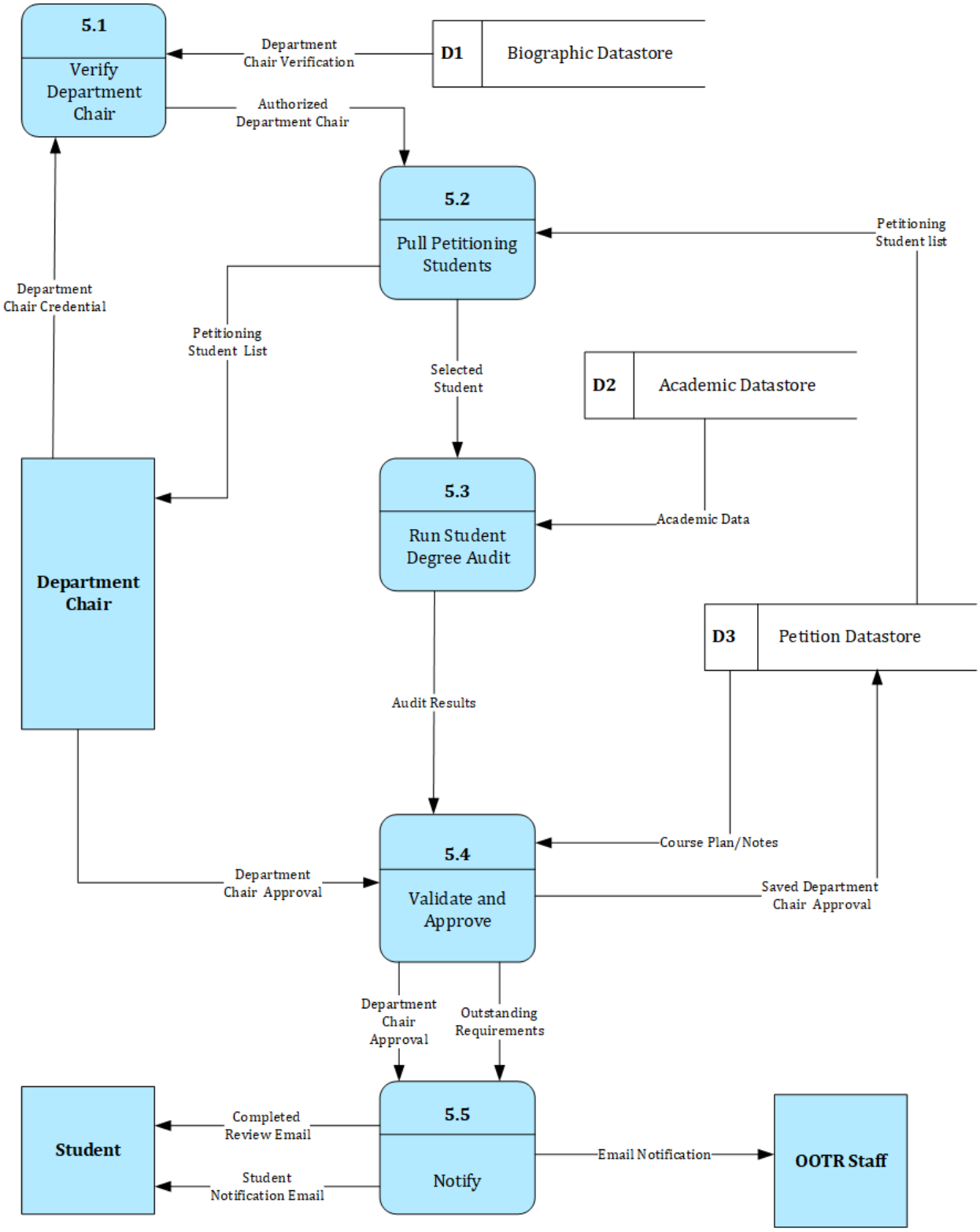
Level 1 Diagram for Process 3:



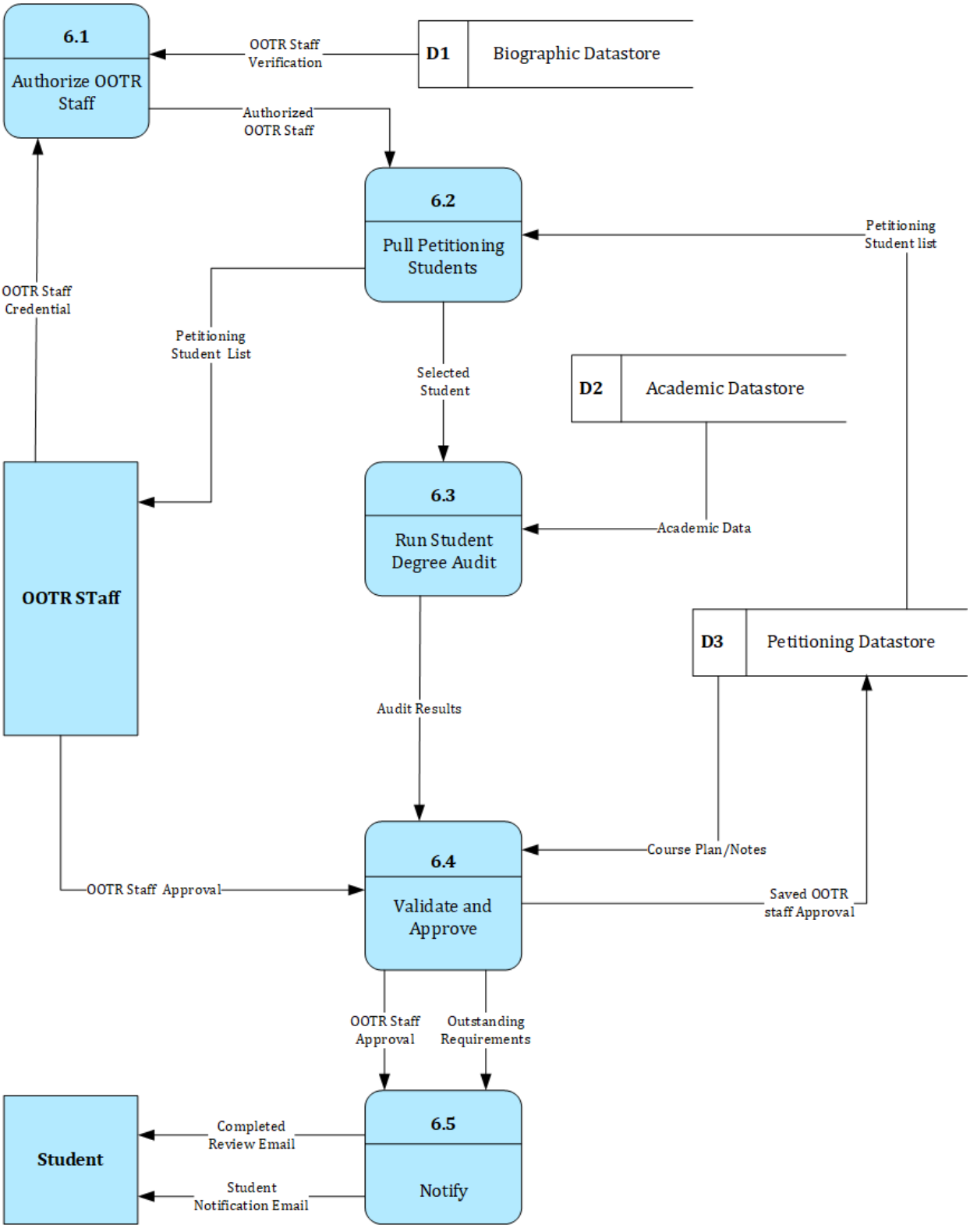
Level 1 Diagram for Process 4:



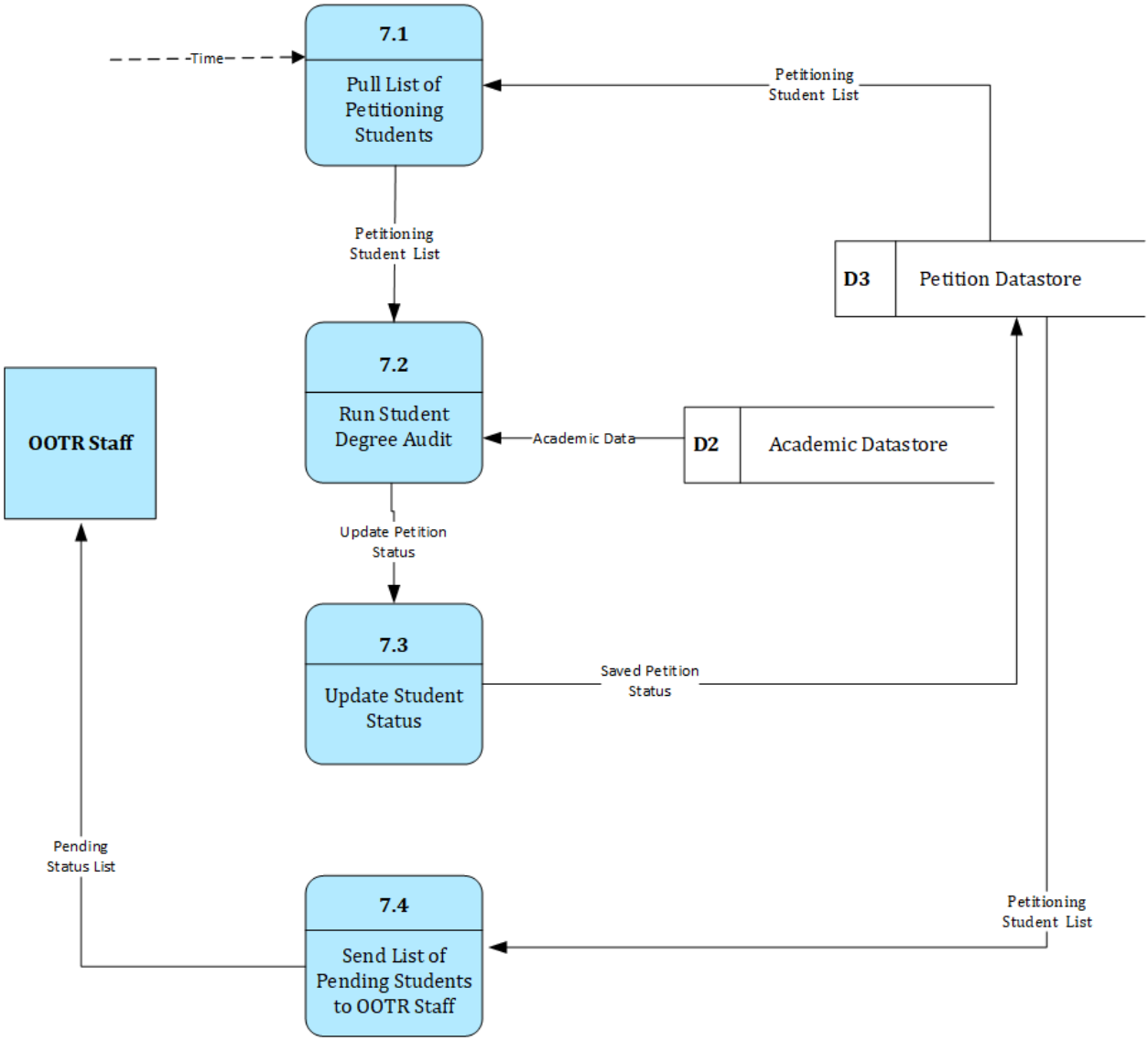
Level 1 Diagram for Process 5:



Level 1 Diagram for Process 6:



Level 1 Diagram for Process 7:



Data Dictionary

The majority of data captured in our Data Dictionary is common and typical as almost every piece of data is either an integer or character with a specified maximum value. The Data Dictionary is alphabetical by data flow and includes information regarding which Data Flow Diagram has a corresponding data flow and the datastore to which it belongs.

Data Flow: Academic Data	**Datastore: Academic**	**DFD 1, 3, 4, 5, 6, 7**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Program	varchar(15)		
Degree	varchar(3)		
First Major	varchar(40)		
First Major Subplan	varchar(40)		
Second Major	varchar(40)		
Second Major Subplan	varchar(40)		
Third Major	varchar(40)		
Third Major Subplan	varchar(40)		
Frist Minor	varchar(40)		
Second Minor	varchar(40)		
Third Minor	varchar(40)		
Subject	varchar(40)		Student will have as many rows as classes needed
Catalog Number	varchar(40)		Student will have as many rows as classes needed
Quarter	varchar(40)		Student will have as many rows as classes needed
Grade	char(3)		Student will have as many rows as classes needed

Data Flow: All Classes	**Datastore: Class Schedule**	**DFD 3**	**Notes**
Fields	*Format*	*Default Value*	
Subject	varchar(40)		Multiple rows as needed
Catalog Number	varchar(40)		Multiple rows as needed
Quarter	varchar(40)		Multiple rows as needed, shows quarter offered

Data Flow: Contact Info	**Datastore: Biographic**	**DFD 1**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
First Name	varchar(30)		
Last Name	varchar(30)		
Email Address	varchar(50)		

Data Flow: Course Plan/Notes	**Datastore: Petition**	**DFD 3, 4, 5, 6**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Subject	varchar(40)		Student will have as many rows as classes needed
Catalog Number	varchar(40)		Student will have as many rows as classes needed
Quarter	varchar(40)		Student will have as many rows as classes needed
Notes	varchar(200)		
Course Plan/Notes Completion Flag	char(1)	'N'	

Data Flow: Department Chair Approval	**Datastore: Petition**	**DFD 5**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Final Department Chair Approval Flag	char(1)	'N'	
Notes	varchar(200)		

Data Flow: OOTR Staff Approval	**Datastore: Petition**	**DFD 6**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Final OOTR Staff Approval Flag	char(1)	'N'	
Notes	varchar(200)		

Data Flow: Petition to Graduate Approval	**Datastore: Petition**	**DFD 1, 2, 7**	**Notes**

Fields	*Format*	*Default Value*	
Student ID	integer		
Petition to Graduate Approval	char(1)	'N'	

Data Flow: Preferred / Validated Diploma Address	**Datastore: Biographic**	**DFD 2**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Address Line 1	varchar(40)		
Address Line 2	varchar(40)		
City	varchar(20)		
State	char(2)		
Zip Code	char(5)		
Country	varchar(40)		
Address Type	char(10)	Diploma	

Data Flow: Primary / Validated Diploma Name	**Datastore: Biographic**	**DFD 2**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
First Name	varchar(30)		
Middle Name	varchar(30)		
Last Name	varchar(30)		
Name Type	varchar(10)	Diploma	

Data Flow: Saved Advisor Approval	**Datastore: Petition**	**DFD 4**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Final Advisor Approval Flag	char(1)	'N'	
Notes	varchar(200)		

Data Flow: Student Account Trigger	**Datastore: Petition**	**DFD 2**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Account Trigger Flag	char(1)	'N'	

Data Flow: Student Petitioning List	**Datastore: Petition**	**DFD 4, 5, 6, 7**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Graduation Term	varchar(12)		
Course Plan/Notes Completion Flag	char(1)	'N'	
Final Student Approval Date	date		

Data Flow: System Generated Approval	**Datastore: Petition**	**DFD 7**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
System Generated Approval	char(1)	'N'	

Data Flow: Time of Email	**Datastore: Petition**	**DFD 1**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Time of Email	date		

Data Flow: Validated Acknowledgement	**Datastore: Petition**	**DFD 2**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Acknowledgement Flag	char(1)	'N'	

Data Flow: Validated Final Approval	**Datastore: Petition**	**DFD 2**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Final Student Approval Flag	char(1)	'N'	
Final Student Approval Date	date		

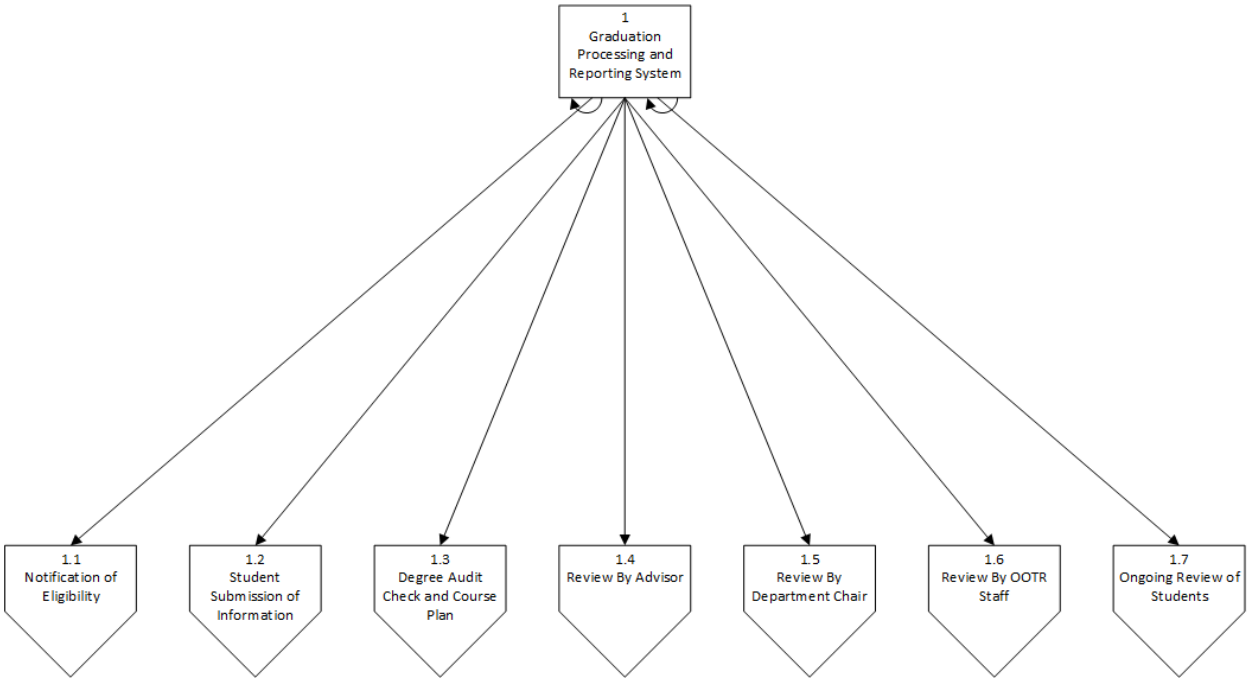
Data Flow: Validated Graduation Term	**Datastore: Academic**	**DFD 2**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Graduation Term	varchar(12)		

Data Flow: Validated Majors/Minors	**Datastore: Academic**	**DFD 2**	**Notes**
Fields	*Format*	*Default Value*	
Student ID	integer		
Program	varchar(15)		
Degree	varchar(3)		
First Major	varchar(40)		
First Major Subplan	varchar(40)		
Second Major	varchar(40)		
Second Major Subplan	varchar(40)		
Third Major	varchar(40)		
Third Major Subplan	varchar(40)		
Frist Minor	varchar(40)		
Second Minor	varchar(40)		
Third Minor	varchar(40)		

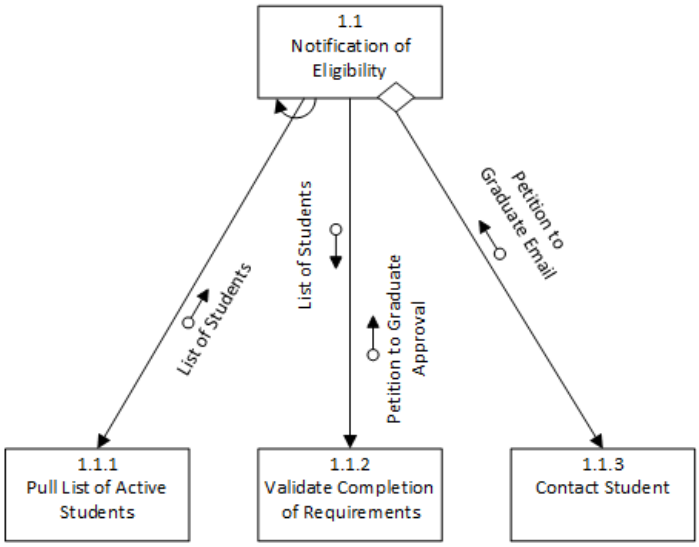
Program Structure

The program structure has a number of library modules that are repeated throughout, include verifying the constituent, running degree audits and pulling lists of students. This will reduce the number of modules that need to be created and reduce the programming labor needed. The program structure is well balanced with only no structure including more than 7 modules in the fan out. Most of the structures are transform structures as data is passed back and forth to change the input to the appropriate output.

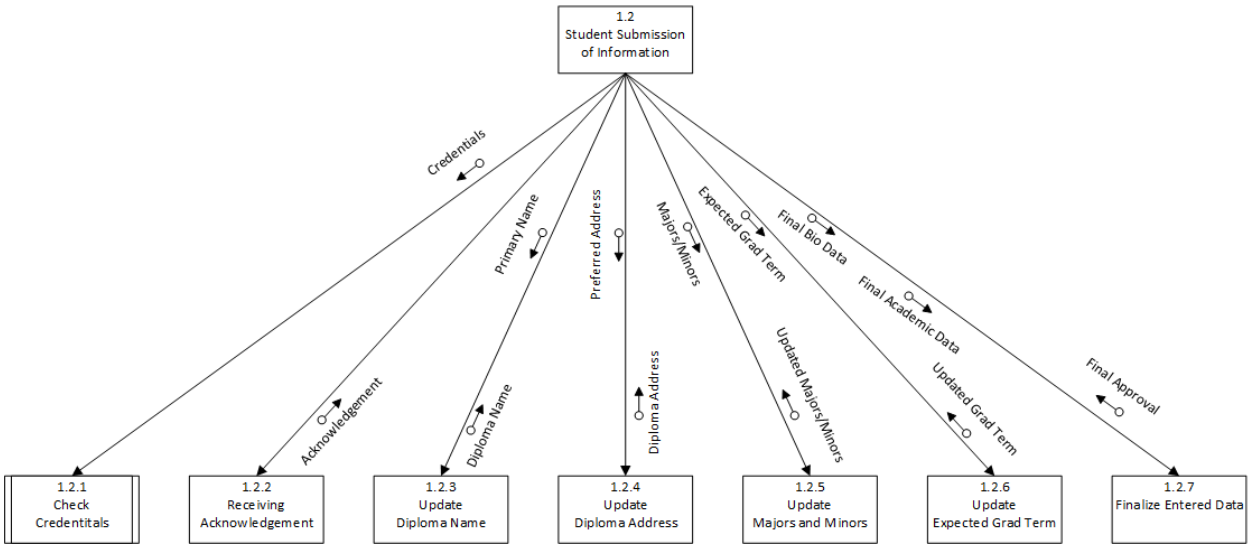
Structure Chart 1:



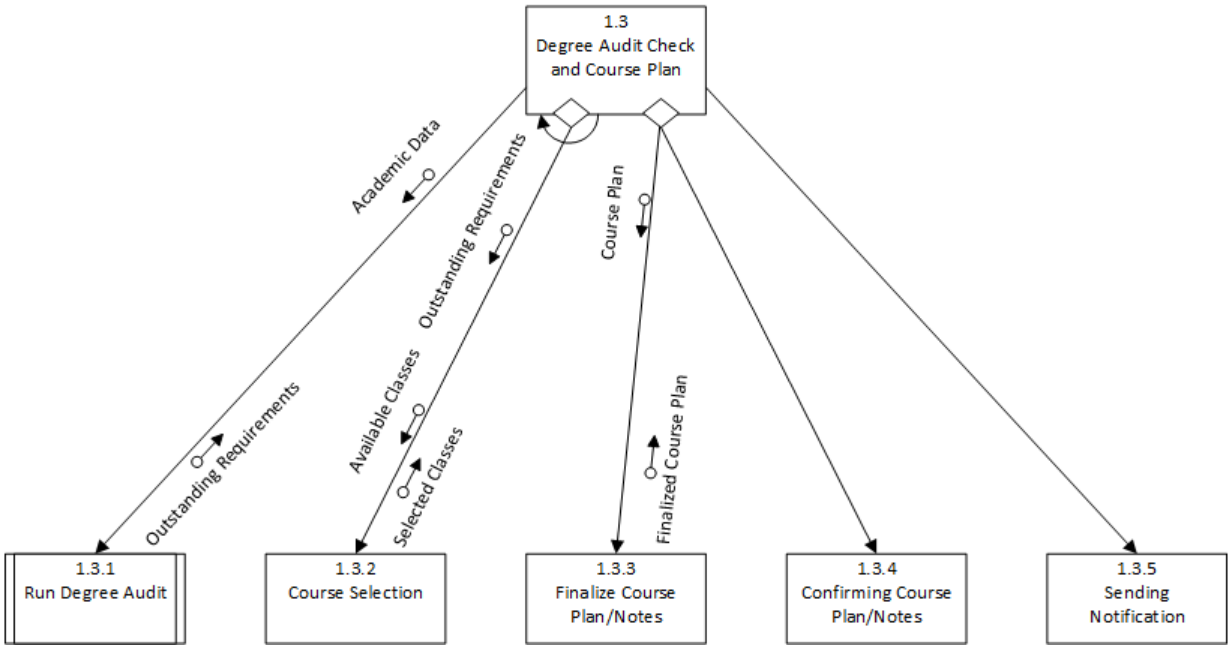
Structure Chart 1.1:



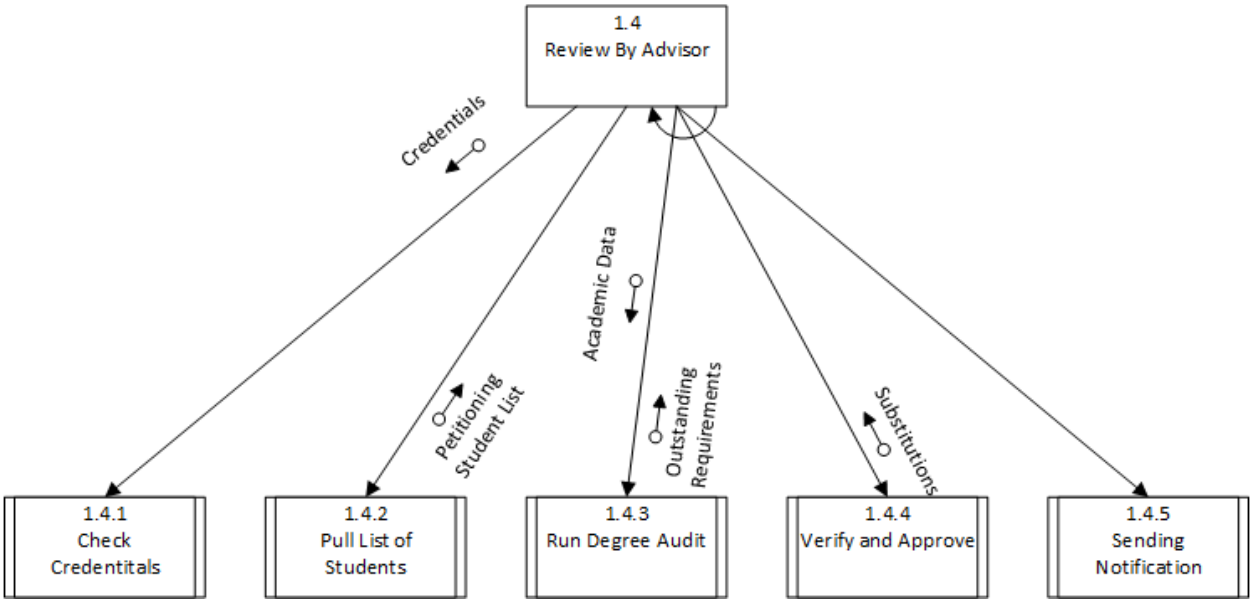
Structure Chart 1.2:



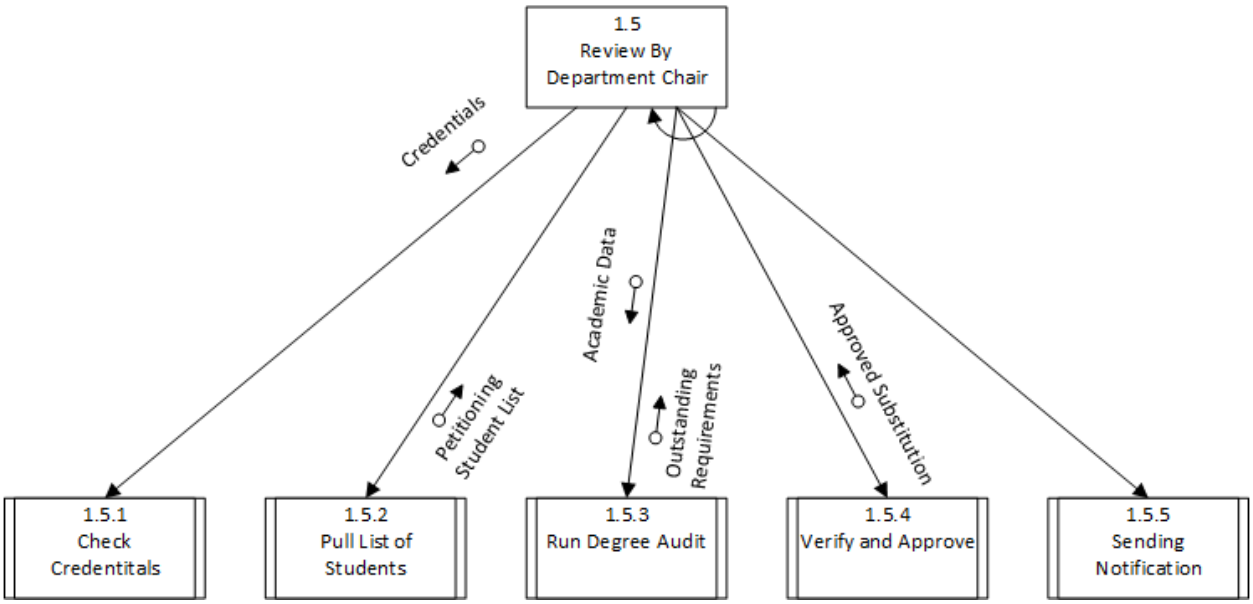
Structure Chart 1.3:



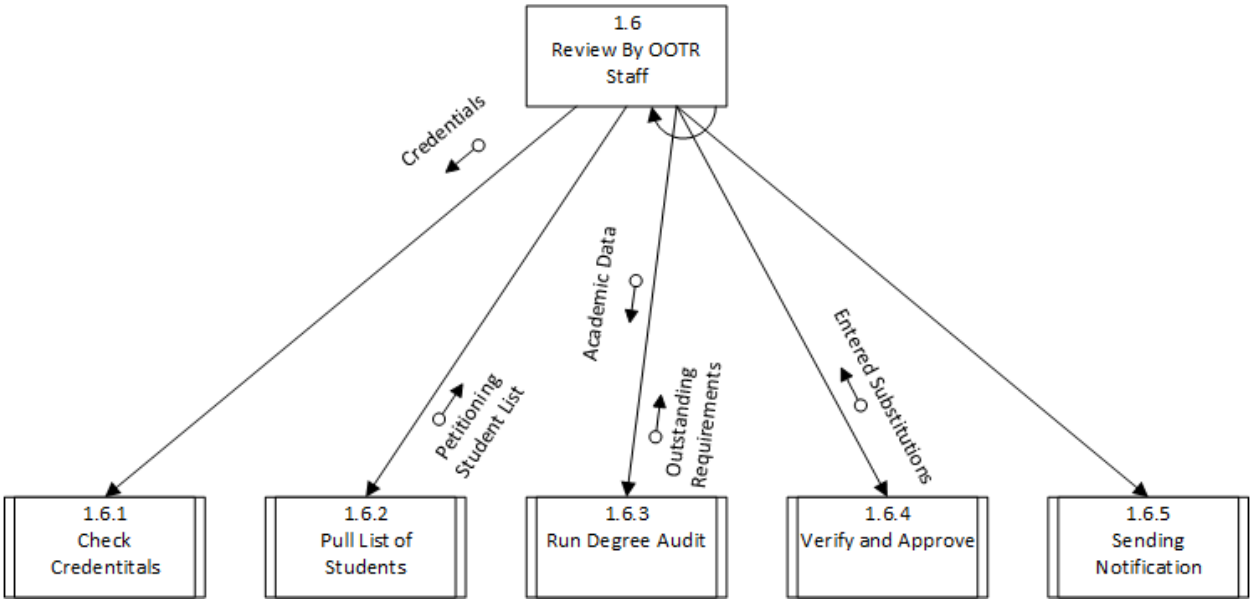
Structure Chart 1.4:



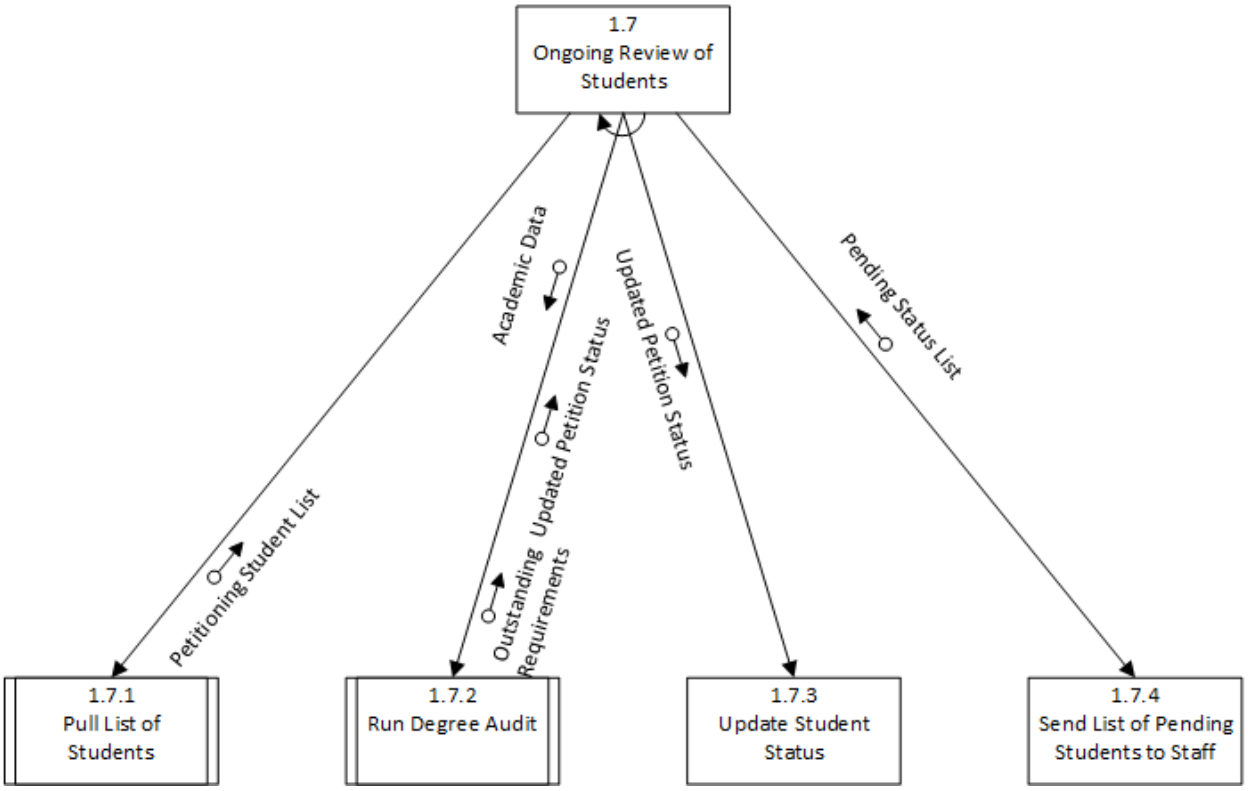
Structure Chart 1.5:



Structure Chart 1.6:



Structure Chart 1.7

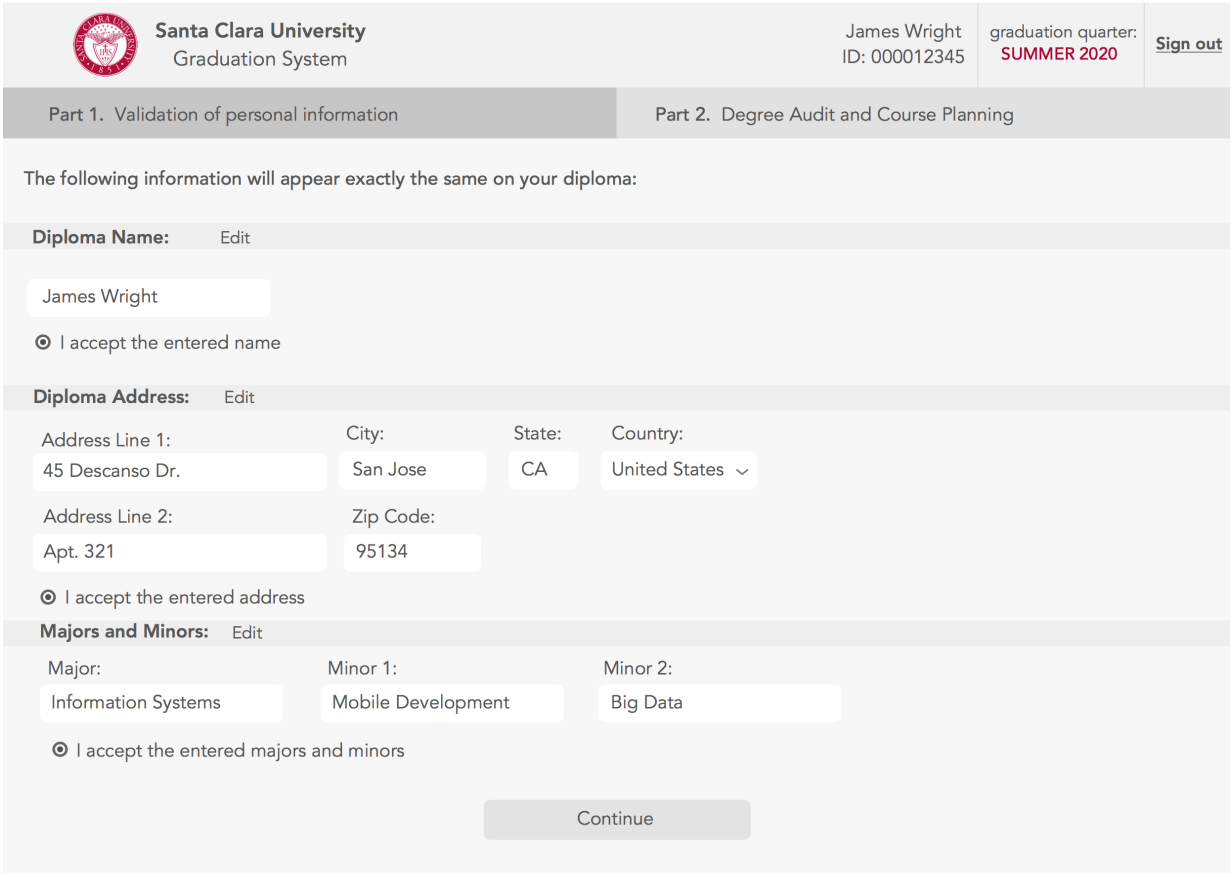


User Interfaces, Reports and Outputs

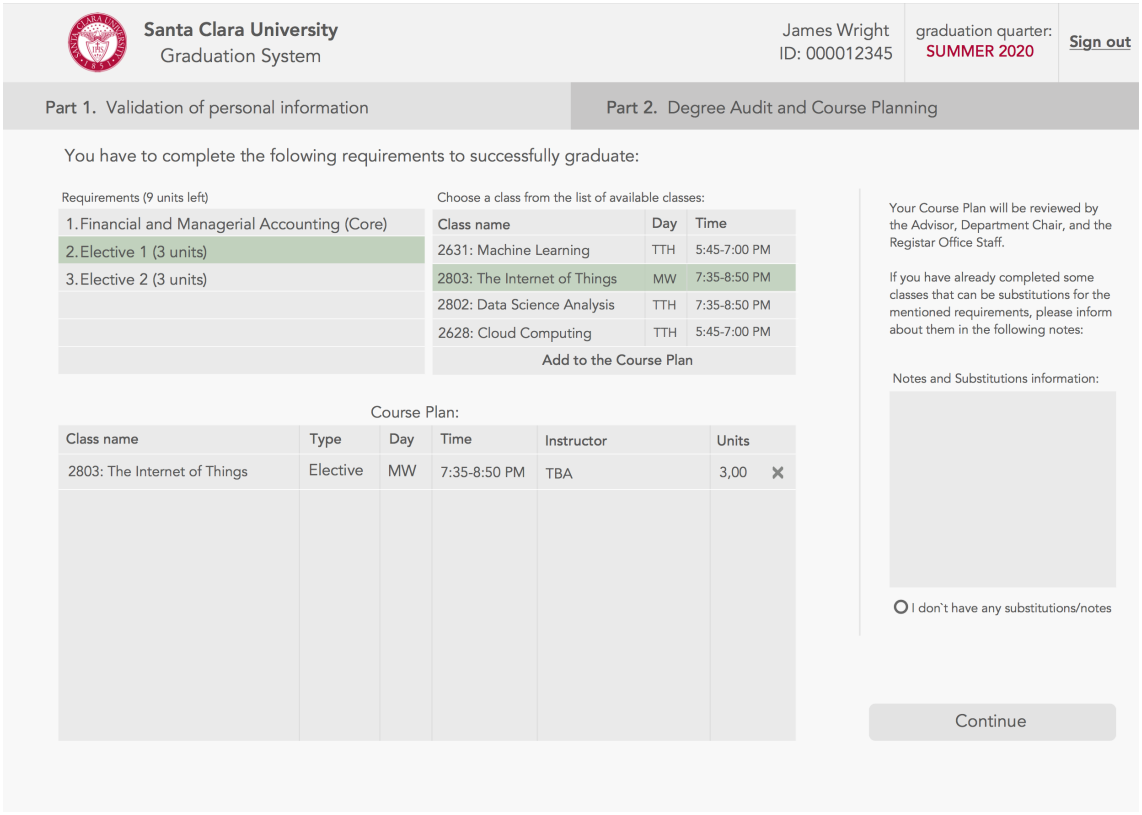
The user interfaces provide a sleek, simple design that allows users to clearly understanding the next steps they need to take in the process. The goal is to make sure that student's enter information at every step and don't skip over any information that may be of importance. A skipped field could mean that a student does not enter the correct diploma name or diploma address. Clarity is also necessary when dealing with the approval lists. Advisors, department chairs, and the OOTR staff need to be able to quickly see which students need to be reviewed and which are pending.

There are two different types of outputs used in the system. The first is a report of students that can be run by key stakeholders to find out more information regarding the a certain set of students. The second are sample emails sent during the process to correspond with different actors regarding their status.

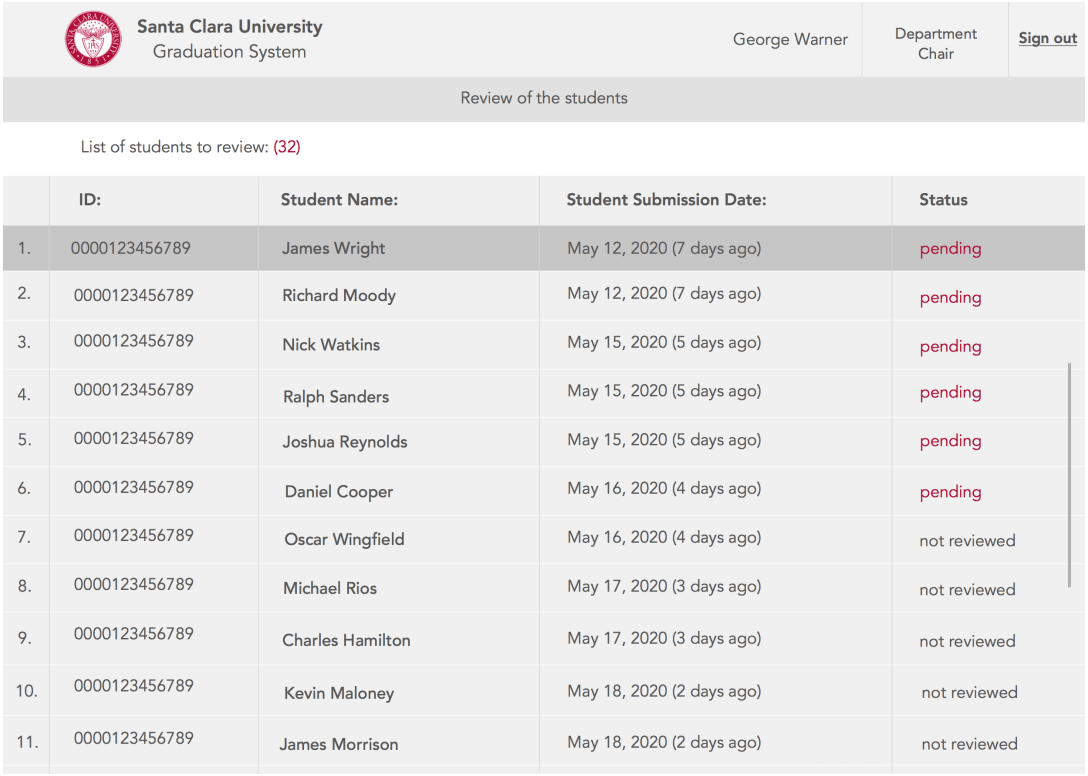
Input of Personal Information Interface:



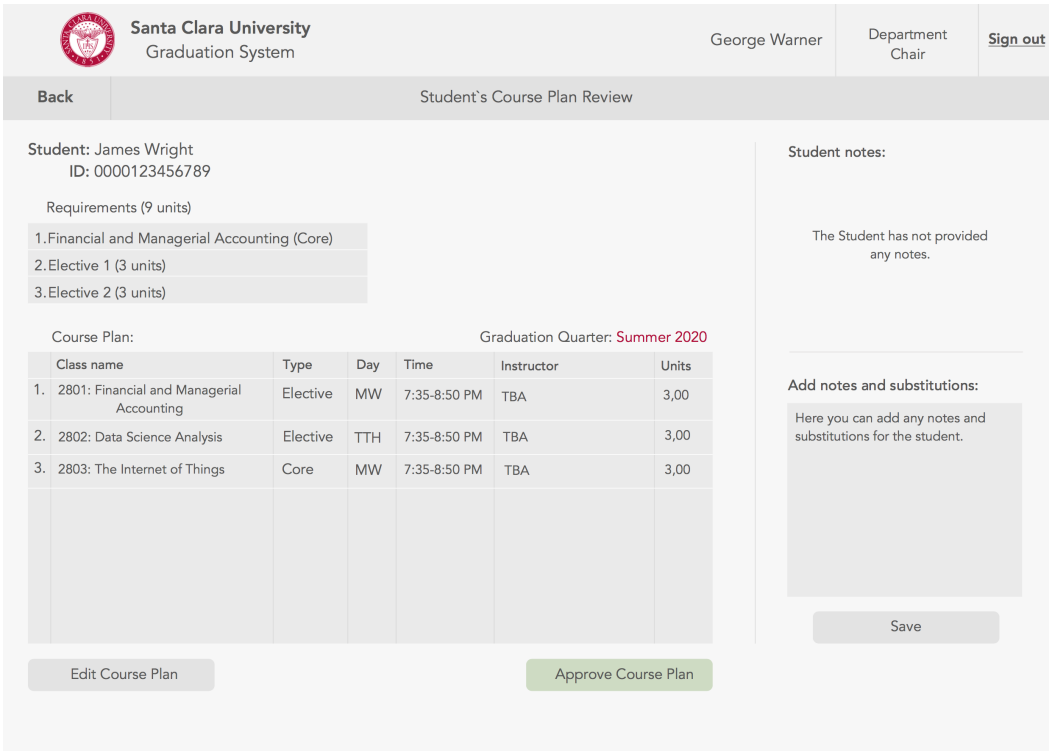
Degree Audit and Course Planning Interface:



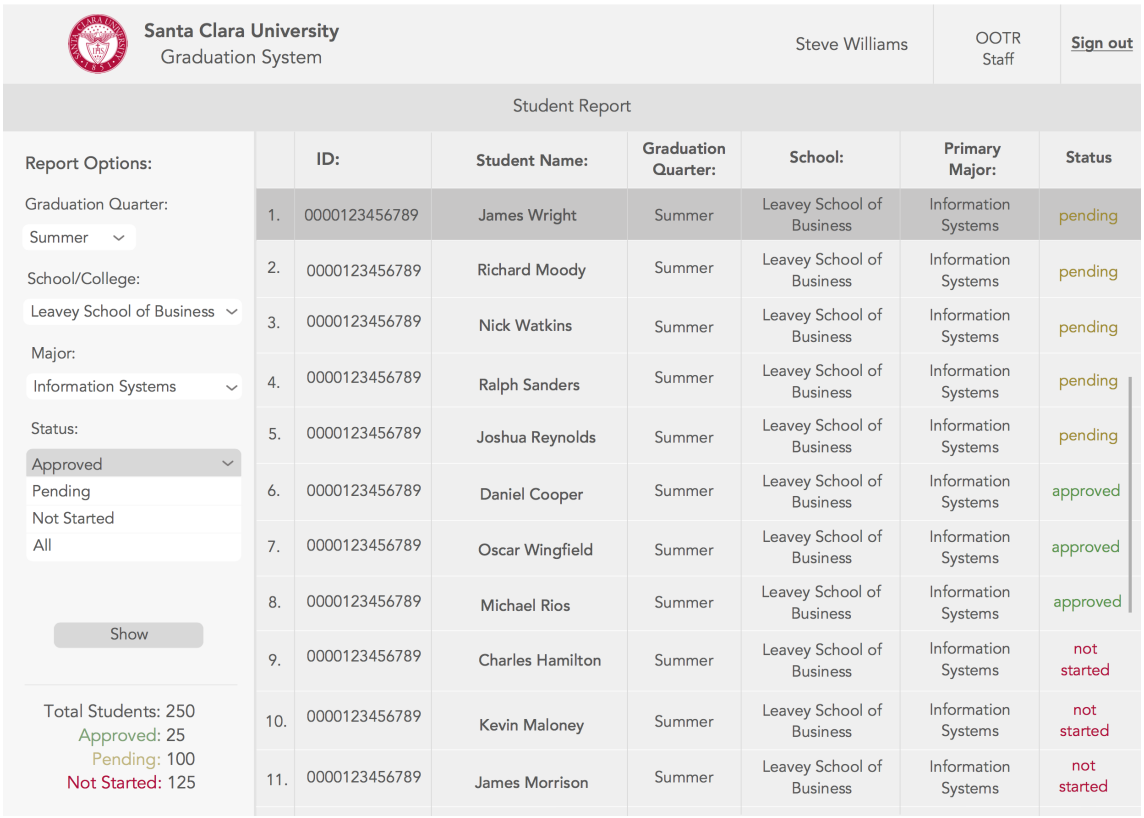
Review of the Students Interface:



Student`s Course Plan Review Interface:



Report Interface:



Sample Email Output - Student Eligible to Graduate:

Dear <STUDENT NAME>,

Congratulations! You have met to qualifications to be eligible to graduate! There are few more steps that need to be taken in order to complete your petition to graduate.

1) Log on to the Graduation Processing System at www.scu.edu/PetitionToGraduate and complete the necessary forms.
2) Keep an eye on your petition status as it will need to be approved by your Advisor, Department Chair and the Office of the Registrar Staff.
3) Start looking at information regarding the June Commencement Ceremony at www.scu.edu/Commencement.

As always, please keep an eye on your degree audit and make sure you have a full understanding of all academic requirements. Please feel free to reach out to your Record Analyst if you have any questions regarding your degree audit.

You are nearing the end of your academic career here at Santa Clara University and we are so excited for you to become an Alumni of Santa Clara University!

Sincerely,
Office of the Registrar Staff

Sample Email Output - Advisor/Department Chair Email:

Dear <ADVISOR NAME>.

One of your advisees has submitted a petition to graduate. Please log on to the Graduation Processing System at www.scu.edu/PetitionToGraduate and complete the necessary forms. Students are not able to continue on to the next step until you have completed your portion.

Petition to Graduate Statistics for <ADVISOR>
Total Outstanding Students: 32
Pending Students: 11
Not Started: 21

Please feel free to reach out to your Record Analyst if you have any questions regarding your degree audit.

Sincerely,
Office of the Registrar Staff

Future Scope of the System

There are two areas in which this system can be expanded in the future: 1) Inclusion of the Graduate Programs and 2) Inclusion of department specific information.

The graduation programs process their students in a slightly different fashion for each program. Minor changes in the system would be necessary if the system were to be expanded to those programs. Changes may include a slightly different workflow or possible changes to the review process. Also, not all graduate programs have a degree audit.

Inclusion of department specific information would be another addition that could be add in a future version of the system. Many, but not all, department conduct “senior survey” to collect feedback from their students. The addition of this information would greatly support the individual departments who help manage their student through the graduation process. Change may include the ability for departments to add a set of questions that a student must answer before the review done by the department chair. Additional reporting would be needed so that each department can review the information submitted by their students.

Conclusion

Santa Clara University prides itself as the Jesuit University of Silicon Valley. Use of technology should be the centerpiece of how the University functions. The current system, using paper, is antiquated and needs to be overhauled to show the University cares about its place in higher education. The Graduation Processing & Reporting System has the opportunity to enhance the last step a student must take before graduating. Streamlining the process through automation and the collection of information electronically will show Santa Clara University’s continued commitment to technology and proves it is truly the Jesuit University of Silicon Valley.