**Towson Academic Pathway (TAP)**

**Fall 2024**

**TAP Dev**

**Nyle Clements, Romerico David, Tim DeLloyd, Mitchell Griff, Zachary Hall**

**10/08/2024**

**TAP Dev Resumes**

**NYLE CLEMENTS**

Baltimore, MD 21206 • 443-681-8629 • [cl1092331@email.ccbcmd.edu](mailto:cl1092331@email.ccbcmd.edu)

**OBJECTIVE**

Computer Science major seeking position to apply programming and problem-solving skills

**EDUCATION**

Community College of Baltimore County - **Associate of Science, Magna Cum Laude**

Towson University – expected graduation May 2026

**Computer Science Major, 3.75 GPA**

**Relevant Coursework:** Data Structures and Algorithms, Advanced Programming, C++ Programming, Introduction to Programming, Data Communications, Calculus II

**TECHNICAL SKILLS**

* **Proficient in Java programming**
* **Implementation of OOP concepts: ADTs, inheritance, polymorphism, recursion, algorithm efficiency, etc**
* **Implementation of collections: stacks, queues, lists**
* **Implementation of data structures: arrays, linear nodes, LinkedLists**
* **Understanding of Big O notation (time complexity of algorithms)**
* **IDEs (Microsoft Visual Studio, Eclipse, NetBeans)**
* **Microsoft Office Suite: Word, PowerPoint, Excel, and Outlook**

**PROFESSIONAL EXPERIENCE**

FX Studios & Under Armour Performance Center, Baltimore MD

Guest Relations Teammate February 2022-present

* Provide front desk service including check in, membership activation, and scheduling.
* Training lead for staff on Mindbody software and leading conversion to ABC software.
* Equipment maintenance, billing, and processing records for personal training sessions.

Hacked & Loaded Research Group at CCBC- Baltimore, MD (Hybrid)

Cybersecurity Intern Summer 2022

* Designed and secured a network infrastructure for three remote sites.
* Implemented security solutions for anticipated security threats.
* Presented diagram and plan to stakeholders.

AutoZone- Baltimore, Maryland

Service Representative February 2020-August 2021

* Provided automotive maintenance, education, and customer service to patrons.
* Reset debit pads and reconfigured Znet computer software during errors.
* Trained new service representatives on automotive maintenance and company software.

**ACADEMIC PROJECTS AND COURSE WORK**

Working with Queues Project— Java

* Completed an implementation of the LinkedQueue class within Java Foundations. Specifically, the first, isEmpty, size and toString methods. Also created a driver class (main) to test the functionality of the methods of the program. ToString printed all the elements in the Queue.
* Completed the implementation of the CircularArrayQueue class within Java Foundations, including all methods. Also created a driver class (main) to test the functionality of the methods of the program.

Time Complexity and Recursion Lab— Java

* Analyzed given algorithms with nested loops and calculated the Big O run time.
* Wrote a program that used recursion to convert a base 10 number into a base b number, where b < 10.

Stacks Lab— Java

* Implemented a class called ReverseWords that used a stack to output a set of elements input by the user in reverse order.
* Wrote a program that examined a string of characters and determined if all opening bracket characters such as (, [, or { were matched with their correct closing characters using a stack.

Bank Accounts Project— Java

* Designed and wrote 3 classes – Account, CheckingAccount, and SavingsAccount.
* Had CheckingAccount and SavingsAccount inherit from Account.
* Prevented deposit method overriding in child classes by using the final keyword.
* Overrode the toString method to display everything the Account toString displays and the minimum balance for SavingsAccount. Took advantage of the parent class toString by using the super keyword to invoke the parent toString method.

**AWARDS AND MEMBERSHIPS**

* Dean’s List, Summer 2021, Fall 2021, 2022
* Technology Club Member, Spring 2021-Present
* Oracle Academy Java Foundations Certificate, Fall 2021
* Male Student Success Initiative Member, Fall 2021-Present

**STEM Core Scholar**  Fall 2021-present

* Completed accelerated mathematics courses through Calculus 1.
* Receiving tuition funding sponsored by the National Science Foundation.
* Student ambassador for the Maryland Blue Collar STEM Festival.
* Selected for a student spotlight interview for the newsletter, November 2021 edition.

**Romerico David Jr.**

[**romericodavidjr.site**](http://www.romericodavidjr.site) **• romedavid2@outlook.com • 443-768-8722 •** [**linkedin.com/in/romerico-david**](https://www.linkedin.com/in/romerico-david)

**EDUCATION**

**Towson University** Towson, MD

*Bachelor of Science in Computer Science (3.94 GPA), Minor in Mathematics* May 2026

* **Coursework:** Object-Oriented Design and Programming, Data Structures and Algorithms, Software Engineering, Web App Development, IOS App Development, Calculus III, Ordinary Differential Equations, Linear Algebra, Discrete Math, Statistical Methods

**TECHNICAL SKILLS**

**Programming Languages:** Java, Python, C++, HTML, CSS, JavaScript (Node.js), TypeScript, JSX/TSX, LaTex, Swift

**Frameworks/Libraries:** Express.js, React.js, Angular, MongoDB, Bootstrap, NumPy, Matplotlib, pandas

**Methodologies**: Agile (Scrum, Kanban, Extreme Programming)

**Databases:** MongoDB

**Developer Tools:** Visual Studio Code, Anaconda, Jupyter Notebook, Git, GitHub

**EXPERIENCES­­**

**SecurEd Inc.** Towson, MD

*Junior Software Developer* Aug 2024 to Present

* Redesigning CLARK and Cyber Competencies products, boosting user engagement and platform usability for 14,000+ active users
* Building and maintaining scalable RESTful APIs using MEAN stack, supporting a web application with 56,000+ downloads
* Developing unit and end-to-end tests for HTTP requests, improving platform stability and reducing bugs
* Apply and utilize Agile methodologies in sprints, optimizing project timelines and code quality
* Leveraging Git and GitHub for version control, ensuring efficient and seamless team collaboration

**Towson University** Towson, MD

*Computer Science Peer Tutor* Feb 2024 to Present

* Provide drop-in tutoring up to 250 students every semester in Java, Python, and C++
* Assist students with understanding the concepts and principles in data structures, algorithms, structured, procedural and object-oriented programming

**Towson University**  Towson, MD

*Undergraduate Researcher in Federated Learning*  Aug 2023 to Jan 2024

* Conducted research on model poisoning in Federated Learning under Dr. Weixian Liao, contributing to the understanding of security vulnerabilities in FL systems
* Utilized the Flower Federated Learning (FL) framework (TensorFlow) to conduct experiments of vulnerabilities to model poisoning attacks in federated learning
* Compared FL aggregation methods FedAvg, FedProx, and QffedAvg across varying types of model poisoning attacks during data processing and model training

**Towson University**  Towson, MD

*Research Intern*  June 2023 to July 2023

* 1 of 12 students chosen for the TIGURS summer undergraduate research program
* Utilized PyTorch, NumPy, pandas, Matplotlib, and scikit-learn to simulate feed-forward, convolutional, and recurrent neural networks using the MNIST and CIFAR-10 datasets
* Evaluated experiments based on Accuracy, Confusion Matrix, Precision, and Recall

**PROJECTS**

[TU Campus Inquiry Project](https://github.com/Romerico234/TU-Campus-Inquiry-Project) June 2024 to Present

* Developing a full-stack web application featuring a REST API to facilitate student requests and connect them with Towson University counselors
* Utilizing the MEAN stack, Bootstrap, and Nodemailer to enhance email functionality
* Implementing secure authorization using Auth0

[Nonlinear ODEs and Linear PDEs Equivalence Project](https://github.com/Romerico234/Nonlinear-ODEs-and-Linear-PDEs-Equivalence-in-Fluid-Dynamics) March 2024 to May 2024

* Researched the equivalence between nonlinear ordinary differential equations and linear partial differential equations in fluid dynamics
* Utilized Python and frameworks such as NumPy, SciPy, and Matplotlib for simulation and visualizations
* Developed papers and presentations using LaTex and Microsoft PowerPoint

**TIMOTHY DELLOYD**

tpdelloyd@comcast.net – tdello3@students.towson.edu – (301) 676-1656

**SUMMARY**

A hard-working, patient mathematics major, computer science minor, with excellent problem-solving skills. Able to communicate complex concepts well and get along well with others (team player). Good at learning new things and following written directions and paying close attention to detail.

**TECHNICAL SKILLS** 

* Coding in C, C++, Java, Python, LaTeX, XML, HTML, NetBeans, VS Code, Wolfram Mathematica, MatLab, Microsoft Office, Word, PowerPoint, Excel, Publisher, NumPy, MatPlotLib, Pandas, Geopandas, IBM SPSS, Jupyter, Slack, GitHub, Linux, Ubuntu

**RELATED EXPERIENCE**

Tutoring: Saint John’s Catholic Prep, Buckeystown, MD

* Tutored students weekly on math and physics for two years as a National Honor Society member; assisted students with their homework

Leadership: Frederick County Knucklers (Marbles Club), Middletown, MD

* Taught younger kids how to play the game of ringer (competitive marbles); Was previously a member until aged out of competition (2013-2016); Assisted with Marble Club promotions at local events, including Brunswick Railroad Days

Volunteering: United Way Summer Serve, Frederick County, MD

* Taught disabled kids how to ride bicycles with iCanShine; Helped weed a field for a farmer
* Helped clean up trash from a creek bed; Helped gather school supplies for low-income families

Volunteering: Sound Crew (High School AV Club), Saint John’s Catholic Prep, Buckeystown, MD

* Helped set up and take down sound equipment for the monthly masses; Helped with sound equipment during assemblies

Experience: ACE Mentor Program (acementor.org)

* Worked in a group under the mentorship of professionals in the architectural, construction, and engineering fields to create a mock apartment building project following specified criteria

**WORK EXPERIENCE**

* Amazon warehouse associate – 20 hours/week – Summer 2023 @ DBA7 in Williamsport, MD
* Research Assistant – 40 hours, Summer 2024, Towson University Mathematics Department
* Software Assurance Intern – 40 hours/week, Summer 2024, US Government

**EDUCATION**

*High School Diploma* – Saint John’s Catholic Prep, Buckeystown, MD – June 2020

*Bachelor of Science, Mathematics (Applied Concentration)* – Towson University, Towson, MD – May 2025

* **Minor**: *Computer Science*; **GPA**: 3.53/4.0; was accepted into the Honors College; Dean’s List Fall 2021

**RELATED COURSES**

* ***Elementary Linear Algebra***– learned the basics of linear algebra; eigenvalues/vectors, nullspace, cross product, etc.
* ***Calculus III*** – learned multivariate calculus
* ***Introduction to Abstract Mathematics***– learned the basics of advanced mathematical branches
* ***Probability***– learned how to comprehensively read probability-based problems and determine the formula(s) to solve
* ***Computer Science II*** – learned more advanced coding techniques with Java, like objects
* ***Mathematical Statistics***– learned how to test statistical hypotheses and calculate the likelihood of error in such tests
* ***Data Structures and Algorithm Analysis*** – learned data search algorithms such as breadth first search, hash tables, random sort, etc. and their speeds in Big O Notation; involved some creative problem solving on various assignments
* ***Artificial Intelligence***– learned about the basics of AI, including concepts like MinMax and AlphaBeta pruning
* ***Differential Equations***– learned how to solve systems of ordinary differential equations
* ***Real Analysis***– learned the rigorous mathematical proofs behind Calculus, such as the basis of Taylor’s Theorem
* ***Object Oriented Programming and Design***– learned advanced programming techniques in OOP
* ***Complex Analysis***– learned about mathematics on the complex plane (imaginary numbers)
* ***Abstract Algebra***– learned group theory, the definition of a field, modulo, etc.
* ***Numerical Analysis*** – learned about data approximation algorithms; estimating error; included coding algorithms in Python
* ***Mathematical Models*** – learned about applying math in physics/chemistry
  + Completed a group project with 2 classmates that involved analyzing a paper on macroscopic traffic models and presenting it to the class; we received an A
  + Completed a group project with 2 classmates that involved collecting and analyzing data with a Lotka-Volterra model; required us to figure out a program we had never used before (MatLab) to get a least squares regression line 2 nonlinear differential equations; I used Python with my own implementation of Heun’s method to graph experiments that involved changing the coefficients in the original model; we received an A
* ***Senior Seminar in Mathematics*** – learned about the mathematical proofs behind machine learning; did group and individual projects
  + Completed a large individual project by singlehandedly developing an algorithm to estimate a non-axis parallel rectangle on a closed plane when only knowing whether single points were inside or outside the rectangle, think battleship; figured out the concept of convex hulls with no prior knowledge of them
* ***Applied Regression*** – learned about various regression models and how to analyze them
* ***Software Engineering*** – learning about the software development process (agile); will complete a group project
* ***Cryptography*** – learning about various encryption and decryption methods
* ***Combinatorics*** – learning about the Pigeonhole Principle and various counting methods
* ***Applied Math Lab*** – learning about mathematical research by measuring crime mobility in Baltimore City

Mitchell Griff

301-514-5273 | mgriff30@students.towson.edu | https://github.com/Graymarch

### Education

Towson University, Towson MD - B.S Computer Science (Software Engineering):

* September 2022 - Current
* Coursework: Object Oriented Design and Programming, Software Engineering, Operating Systems, Data Structures and Algorithm Analysis, Calculus II
* GPA: 3.84

### Experiences

N/A

### Relevant Skills

Languages: Java, C++, Visual Basic, HTML,

Database Tools: SQL

Platforms: Visual Studio, Visual Studio Code, Github, Slack, Ubuntu

**Zachary Hall**

Baltimore, MD | LinkedIn | 443-473-3387 | zacharyhallofficial@gmail.com

**EDUCATION**

**Towson University Towson, MD** *Bachelors of Science, Computer Science 2026*

● Coursework: Gen Computer Science (C++)

**Boys’ Latin School of Maryland (High School) Baltimore, MD** *Diploma, GPA: 3.48/4.0 2022*

● Awards: Dean’s List, Honor Roll

● Related Courses: Cybersecurity, Computer Science 1 (Python), AP Computer Science Principles (Exam), AP Calculus AB

**LEADERSHIP EXPERIENCE**

**Boys Latin School of Maryland (Club) Baltimore, MD** *Co-Founder & Co-Leader 2021-2022*

● Taught a group of 7 how to earn money online through freelance work and entertainment.

● Used my experience as a freelance graphic designer to educate my students on how to design thumbnails that grabbed a potential viewer's attention.

● Learned how to organize, create and give helpful presentations that were tailored to all the club members’ needs.

**PROJECTS**

**Portfolio Review -** [**Website**](https://portfolioreview.me/) **,** [**Github**](https://github.com/darkzelli/portfolioreview) **2024-Present**

● Designed and created a platform for software engineers to receive feedback on their portfolios (**JavaScript, HTML, CSS, Node.js, React.js).**

● Created a system to self host the application on a VPS through SSH using Coolify

● Implemented an efficient caching system using a combination of React Query and the NextJs API to reliable and fast user interface

**Programming Skills:** JavaScript, Python, C++

**Other:** HTML, CSS, ReactJS, Git, NodeJS, Supabase, Nextjs, React Query, AWS SES, Coolify

**Interests:** Freelance Graphic Design, Uploading and streaming gaming content to youtube and twitch, Tradition Art

**Scheduling and Planning Tables**

| Assignee | Email | Task | Duration (hours) | Dependency | Due Date |
| --- | --- | --- | --- | --- | --- |
| Nyle Clements | nclemen1@students.towson.edu | Collaborate with Timothy on writing our problem statement per Task 4. | N/A | Google Drive | 10/8/24 |
| Romerico David | rdavid6@students.towson.edu | Setting up the GitHub repository and kanban project per Task 2.  Describe the system’s architecture per Task 5.  Discuss project details with Dr. Sai per Task 3.  Formatting report per Task 6. | 5 | GitHub  Outlook  Google Drive | 10/8/24 |
| Timothy DeLloyd | tdello3@students.towson.edu | Wrote the problem statement per Task 4.  Helped Romerico with Task 5. | 2.5 | Google Drive | 10/8/24 |
| Mitchell Griff | mgriff30@students.towson.edu | Build the assignment table as in Task 1.  Partial contribution to Task 3, communicating with Dr. Sai about the project topic.  Minor contribution to Task 2, pasting screenshots of relevant materials in the report. | 1.5 | Google Drive  Outlook | 10/8/24 |
| Zachary Hall | zhall6@students.towson.edu | N/A | N/A | Google Drive | 10/8/24 |

**Problem Statement**

Towson Academic Pathway (TAP) is a web application designed to assist Towson University students in planning and organizing their academic path. TAP generates personalized course recommendations and adjusts plans dynamically based on students' preferences, progress, and Towson's academic policies.

TAP is for incoming and current students of all majors at Towson University, regardless of whether they are full or part-time. Auditors and non-students who are merely considering a select few classes may not find it as useful.

For a student, the process of planning out what courses one will take over the next few years can be very difficult and overwhelming. TAP eases out that process using artificial intelligence. Additionally, plans may need to change if a student changes their major or is not able to take a course when planned because it was not offered or because the course did not fit their schedule. TAP can also help in these instances as well, making the process of replanning less stressful on the student.

TAP helps students schedule classes for an upcoming semester. Using information from a students’ career path, other obligations a student has, and allowing for various preferences, TAP can provide the student with scheduling options for the upcoming semester(s) from Towson’s list of offered courses. This is often a stressful obligation students often go through at least twice a year while tackling their current classes and whatever other obligations they might have; TAP would help alleviate this stress and let students focus on the rest of their lives by handling the logistics of their next semester for them.

The only current alternative is to work manually, for example with an Excel spreadsheet. This is time consuming and can distract a student from other obligations they may have in their lives, such as their current classes, a job, or parenting.

Students have many responsibilities and obligations during a semester; they have their current classes, may have jobs, and may even be parents. The logistical nightmare of determining what classes one should take in the upcoming semester is always stressful, time-consuming, and distracting. TAP helps busy Towson students by taking the scheduling process off their backs, making their lives just a bit easier.

The top-level objective is to provide the target customer, a Towson student, with a list of potential schedules for the upcoming semester for them to choose from. The scheduler is limited to upcoming semesters (1 minor semester and 1 major semester) since the availability of classes beyond those semesters is unknown. For example, one cannot schedule classes for Fall 2026 during Fall 2025, because the Fall 2026 catalog has not been released.

TAP does not have any known competitors, which makes TAP new and innovative.

AI-based scheduling software has already been built for other purposes, such as Data317, which has software to help school districts create class schedules across a whole school; it only makes sense, then, that our scheduling system would be buildable.

By simply inputting preferences and requirements, students can use TAP to generate several potential schedules, making the process of planning their academic plan simpler, efficient, and less stressful.

**System Requirements**

The Towson Academic Pathway (TAP) is a web application designed to help students efficiently plan and schedule their courses. TAP integrates a backend with the frontend through API calls to provide personalized academic planning based on individual student needs.

### **Frontend:**

The TAP frontend will be built using **React.js** and will feature the following components:

* **HomeComponent:** Homepage to display a little bit about the team, product, and mission statement
* **FormComponent**: Gathers user input regarding preferences, academic progress, and Towson University policies. This data will be processed using a script that leverages OpenAI to generate a personalized four-year academic plan. User inputs will be stored in the database and referenced in the SettingsComponent.
* **SettingsComponent**: Manages user preferences, academic interests, and personal information. Users can modify their preferences and update their academic schedule as needed.
* **CatalogComponent**: Displays all available courses with filtering options to help users browse and select courses.
* **DegreeCompletionPlannerComponent**: Provides a detailed academic plan for four years (8 semesters by default). Users can add or remove semesters and courses to customize their plan, with dynamic updates based on prerequisite requirements, course selection, and interests.
* **EnrollmentPlannerComponent**: Focuses on the upcoming semester’s schedule.
* **AboutComponent**: Introduces the development team behind TAP.
* **FAQComponent**: Provides a list of frequently asked questions and allows users to submit email inquiries which will be accomplished through **Nodemailer**.
* **AdminComponent**: Offers administrative functions for managing student data and other components, such as handling user-submitted FAQs.
* **AuthenticationComponent\***: Handles user authentication, including login and logout functionality. This may require additional supporting components.

All components will be styled using **Tailwind CSS**

**HTTP requests** between the frontend and backend will be handled using **Axios**.

### **Backend:**

The backend will be developed using **Express.js**, responsible for handling user requests and managing database interactions. The database will be hosted using either **MongoDB** or **Firebase**, depending on the final decision. It will contain the following clusters:

* **Student Data Cluster**: Stores student-related data, including preferences, academic progress, and schedules.
* **Course Data Cluster**: Holds course data from Towson University.. Will accomplish this by creating a web scraping program to scrape data from TU website.
* **FAQ Cluster**: Store user-submitted questions which will be displayed in the AdminComponent for review and response.

### **APIs:**

### Integrating a **REST API** between the frontend and backend for seamless communication and data exchange.

### **OpenAI API**: Used to generate personalized academic recommendations based on user input.

**Interactions**

### **1. User Authentication**

#### *User Interactions:*

* Users will login and log out using the **AuthenticationComponent**, providing their credentials.

#### *System Interactions:*

* The frontend sends the login request to the backend, where user credentials are validated against stored data. The backend then provides access tokens if authentication is successful or returns an error if the login fails.

### **2. User Preferences and Academic Planning**

#### *User Interactions:*

* Through the **FormComponent**, users will input their personal information, course preferences, academic progress, goals, etc.
* Users can adjust or modify their personal information or preferences through the **SettingsComponent** with the option to update to their academic plan.

#### *System Interactions:*

* Upon form submission, the frontend sends the input data to the backend databases. The data then communicates with an **OpenAI API** to generate a personalized four-year academic plan based on the user’s preferences and university policies.
* The backend stores the user inputs and generated academic plan in the **Student Data Cluster** and returns the results to the frontend. These results are then displayed in the **DegreeCompletionPlannerComponent** for the user to view or modify.
* When preferences are updated in the **SettingsComponent**, the changes are sent to the backend database, which updates the stored data and recalculates the academic plan if requested.

### **3. Course Selection & Scheduling**

#### *User Interactions:*

* Users will browse and select courses using the **CatalogComponent**, which includes filters to help narrow down available courses. Selected courses are added to their degree completion plan or upcoming semester schedule.
* In the **DegreeCompletionPlannerComponent**, users can add or remove semesters and courses to customize their plan. These changes can also be made to reflect prerequisite requirements.
* The **EnrollmentPlannerComponent** allows users to focus on scheduling courses for the upcoming semester. Users can add or drop courses here based on their immediate needs, such as course availability and time slots.

#### *System Interactions:*

* The **CatalogComponent** sends a request to the backend to retrieve available courses from the **Course Data Cluster**, populated from TU courses. Course data is displayed on the frontend, and when a course is selected, this information is sent to the backend to update the user’s academic plan.
* Changes made in the **DegreeCompletionPlannerComponent** are communicated to the backend databases, where course selections are validated.
* Changes made in the **EnrollmentPlannerComponent** are sent to the backend, which updates the current semester schedule in the databases.

### **4. Admin and FAQ Management**

#### *User Interactions:*

* Users can browse common inquiries in the **FAQComponent**. If they have additional questions, they can submit them via an email form.
* Admins will manage the application through the **AdminComponent**, where they can handle user-submitted FAQs, student data, and course management tasks.

#### *System Interactions:*

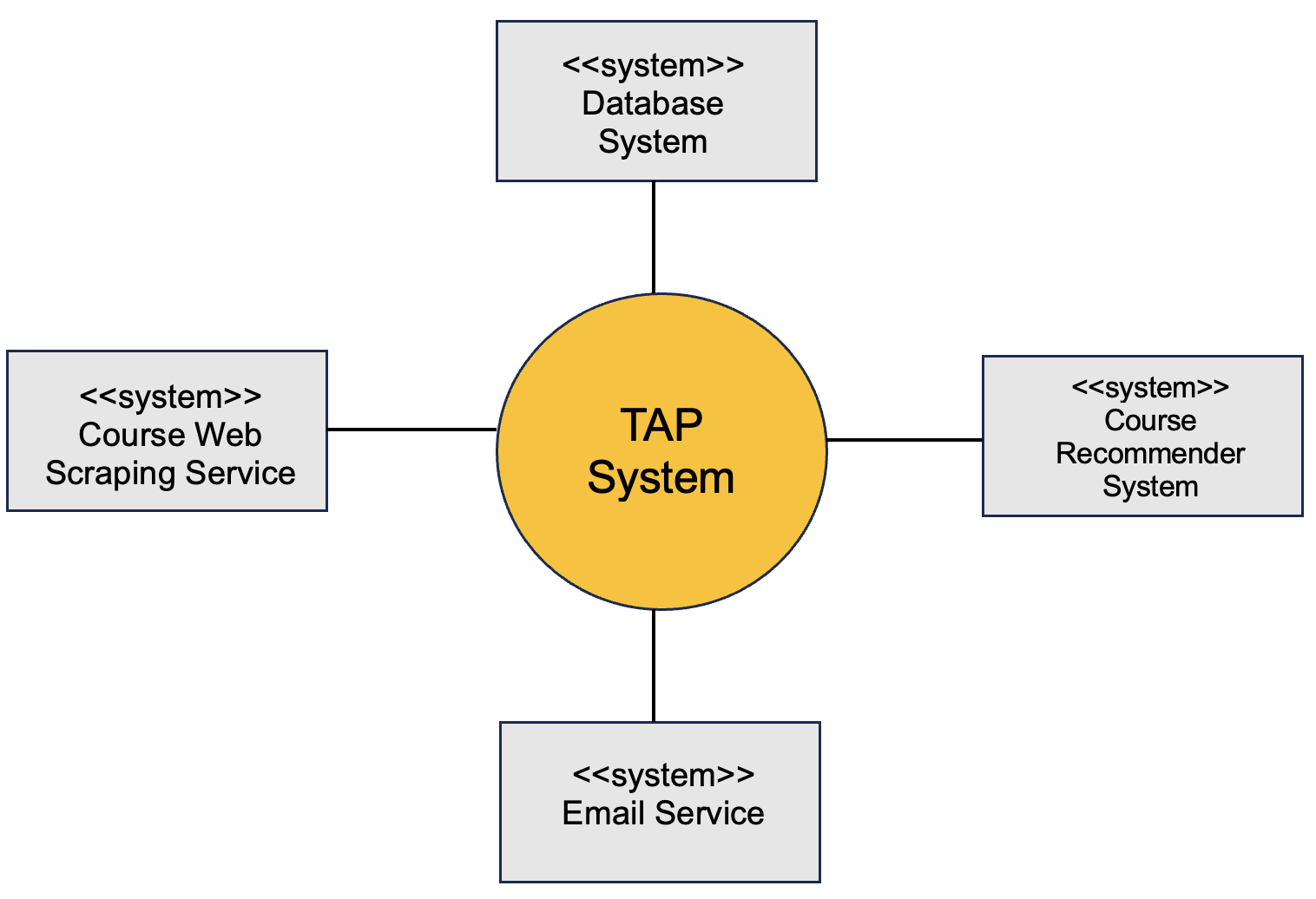
* The FAQ submissions are sent to the backend, where they are stored in the **FAQ Cluster**. Admins, using the **AdminComponent**, retrieve these questions from the backend and manage responses.
* The backend ensures admin privileges for accessing sensitive data related to student interactions, preferences, and course scheduling. Admin updates are saved in the appropriate clusters, such as student or course data, and reflected in the respective user components.

### **5. Data Management and API Interactions**

#### *System Interactions:*

* The backend, built with **Express.js**, serves as the mediator between the frontend and the database. It handles requests from the frontend (e.g., updating course schedules, retrieving course data) and processes them before updating the **Student Data Cluster** and **Course Data Cluster**.
* **API Integration**: The backend interacts with the **OpenAI API** to generate academic plans based on user inputs.

**Context Model**

****

**Appendix**

**README Screenshot:**

****

**Kanban Screenshot:**

