# TA Management Suite — System Design

TA Management Suite has the following distinct subsystems that can be identified based on functionalities and the user roles they cater to.

Here’s a breakdown:

**1. User Authentication and Management Subsystem**

**Components:**

* Azure Active Directory

**Responsibilities:**

* Handle user registration, login, and authentication.
* Role-based access control to assign appropriate permissions to users.

**2. TA Application Management Subsystem**

**Components:**

* Frontend forms for application
* Python backend for application processing
* Azure SQL Database for application data

**Responsibilities:**

* Allow TA applicants to submit, view, and update applications.
* Store application data and attached CVs.

>Sequence Diagram

A screenshot of a computer

Description automatically generated

In this sequence diagram:

* The actors and components involved are identified as TA Applicants, Presentation Layer, Business Logic Layer, Data Access Layer, Data Storage, User Management, and Notification System.
* The steps or methods in the sequence include registering and logging in (Register/Login), opening and filling out the TA application form (Open Application Form, Fill and Submit Form), storing the application in the Azure SQL Database (Store Application), and finally, checking the application's status (Check Application Status).

>Use Case

A diagram of a application

Description automatically generatedA diagram of instruction

Description automatically generatedA diagram of a department staff

Description automatically generatedDiagram of a company member

Description automatically generated

**3. Course Management Subsystem**

**Components:**

* Admin Interface for department staff
* Azure SQL Database for course data

**Responsibilities:**

* Enable department staff to add, update, or delete courses that require TAs.
* Provide a mechanism to specify the qualifications and requirements for each course.

**4. Matching and Recommendation Subsystem**

**Components:**

* Python-based Matching Algorithm
* Azure Logic Apps or Azure Functions for automation

**Responsibilities:**

* Suggest appropriate TA-to-course matches based on various criteria.
* Generate preliminary recommendations for the TA Committee.

>State Chart

**States**

New Application: This is the initial state where the TA applicant starts the application process.

FillingForm: In this state, the applicant fills out the application form with personal and academic details.

CVUpload: After completing the form, the applicant uploads their CV.

CourseSelection: The applicant selects the courses they are qualified for and would like to assist with.

ApplicationReview: Before submitting, the applicant reviews the entire application.

Submitted: The application is successfully submitted and is now in the review process.

StatusCheck: This is where the applicant can check the current status of their application. The sub-states here are:

* Pending: The default sub-state, where the application is in the review process but no decision has been made yet.
* Accepted: The application has been approved.
* Rejected: The application has been denied.

**Transitions**

* Initiate Application: Transitions from the start point to the NewApplication state.
* Edit: Allows the applicant to edit the form while in the FillingForm state.
* Submit: Transitions from FillingForm to CVUpload.
* Next: Transitions from CVUpload to CourseSelection and from CourseSelection to ApplicationReview.
* Confirm: Confirms the application, transitioning from ApplicationReview to Submitted.
* Edit Application: Goes back to the FillingForm state for editing.
* View Status: Transitions from Submitted to StatusCheck for checking the application status.
* Application Approved: Transitions from Pending to Accepted within StatusCheck.
* Application Denied: Transitions from Pending to Rejected within StatusCheck.
* Exit: This transition occurs from Submitted and StatusCheck states to the end state, indicating the end of interaction with the system.

A diagram of a software process

Description automatically generated

**5. Decision-making and Approval Subsystem**

**Components:**

* TA Committee Dashboard
* Azure SQL Database to record decisions

**Responsibilities:**

* Enable TA Committee members to review applications and matches.
* Record and store decisions on TA assignments.

>Component Diagram

Web Application (Frontend)

* Built using HTML, CSS, JavaScript
* Interacts with Business Logic via REST APIs

Business Logic (Backend)

* Developed in Python
* Handles CRUD operations, matching algorithms, and notifications
* Interacts with the Data Access Layer

Data Access Layer

* Azure SQL Database for structured data
* Azure Blob Storage for unstructured data like CVs
* Responsible for all data storage tasks

User Management

* Azure Active Directory for authentication and RBAC
* Interacts with Business Logic and Data Access Layer

Performance Evaluation

* Processes evaluations submitted by instructors
* Interacts with Business Logic and Data Access Layer

Automated Workflows

* Azure Logic Apps or Azure Functions
* Interacts with Business Logic and Data Access Layer

Monitoring and Maintenance

* Uses Azure Monitoring tools
* Interacts with all layers for logging and monitoring

>Component Diagram

A diagram of a software system

Description automatically generated

**6. Notification Subsystem**

**Components:**

* Azure Notification Hubs

**Responsibilities:**

* Notify TA applicants about the status of their applications.
* Notify successful candidates for offer acceptance or declination.

**7. Evaluation and Feedback Subsystem**

**Components:**

* Instructor’s Interface for feedback
* Azure SQL Database for storing evaluation data

**Responsibilities:**

* Allow instructors to submit performance evaluations for TAs.
* Store performance evaluation data for future reference and decision-making.

**8. Monitoring and Audit Subsystem**

**Components:**

* Azure Monitoring Tools
* Azure SQL Database for audit logs

**Responsibilities:**

* Monitor system health and user activity.
* Regular audits for data integrity and compliance.

**9. Backup and Recovery Subsystem**

**Components:**

* Azure SQL Database Backup features
* Azure Blob Storage redundancy features

**Responsibilities:**

* Perform scheduled backups for databases and critical files.
* Enable quick data recovery in case of system failures or data loss.

UML Diagrams:

>Class Diagram

A screenshot of a computer

Description automatically generated

this diagram, the classes represent different layers, roles, and components in the system:

* PresentationLayer, BusinessLogicLayer, and DataAccessLayer form the architectural backbone.
* TAApplicants, DepartmentStaff, TACommitteeMembers, and Instructors are roles that interact with the system.
* UserManagement, DataStorage, BackendLogic, and PerformanceEvaluation are specific functional components of the system.

>ER Diagram:

A screenshot of a computer

Description automatically generated

>Component Diagram

A screenshot of a computer flowchart

Description automatically generated

**Flow:**

**1. Authentication and Management (AuthenticationAndManagement):**

This is the entry point for users into the system.

* Utilizes Azure Active Directory for functionalities.
* Responsible for handling user registration, login, and authentication.
* Implements role-based access control to ensure users have permissions appropriate for their role (e.g., TA applicant, TA committee member, admin, etc.).

**2. TA Application Management (TAApplicationManagement):**

After authentication, TA applicants interact with this subsystem.

* Contains frontend forms for TA application submission.
* Uses a Python backend for processing these applications.
* Stores application data, including attached CVs, in the Azure SQL Database.

**3. Course Management (CourseManagement):**

This subsystem is mainly interacted with by the department staff.

* Provides an admin interface for managing courses that need TAs.
* Uses Azure SQL Database to store details of each course, including their requirements and qualifications.

**4. Matching and Recommendation (MatchingRecommendation):**

This subsystem is vital for intelligently suggesting TA-to-course matches.

* Incorporates a Python-based algorithm that matches TAs to courses based on various criteria, such as qualifications, past experiences, etc.
* Uses Azure Logic Apps or Azure Functions to automate some of these matching processes, like generating preliminary recommendations.

**5. Decision-making and Approval (DecisionApproval):**

This is the subsystem where the TA Committee operates.

* Provides a dashboard for the TA Committee to review and decide upon TA applications.
* Records and stores these decisions in the Azure SQL Database for future reference.

**6. Notification (Notification):**

After decisions are made, notifications are sent out to TA applicants.

* Uses Azure Notification Hubs to inform TA applicants about the status of their applications.
* Sends out notifications to successful candidates, offering them TA positions.

**7. Evaluation and Feedback (EvaluationFeedback):**

After TAs have assisted with courses, instructors provide feedback through this subsystem.

* Provides an interface for instructors to submit performance evaluations for TAs.
* Uses Azure SQL Database to store this feedback for future decision-making.

**8. Monitoring and Audit (MonitoringAudit):**

Ensures the system operates smoothly and complies with necessary regulations.

* Uses Azure Monitoring Tools to keep an eye on system health and user activities.
* Carries out regular audits to ensure data integrity, security, and regulatory compliance.

**9. Backup and Recovery (BackupRecovery):**

Ensures data is safe and can be recovered if anything goes wrong.

* Employs Azure SQL Database Backup features for database backups.
* Uses Azure Blob Storage’s redundancy features for backups of unstructured data, like CVs.

**Flow and Interactions:**

* A user first interacts with the Authentication and Management system to gain access.
* TA applicants then proceed to the TA Application Management subsystem to submit their applications.
* The department staff populates course data via the Course Management system.
* Applications are matched to courses in the Matching and Recommendation subsystem.
* These matches and applications are then reviewed in the Decision-making and Approval subsystem.
* Applicants are notified of the committee’s decision through the Notification system.
* After the TA tenure, instructors provide feedback using the Evaluation and Feedback system.
* System activities are continuously monitored by the Monitoring and Audit system.
* All data is backed up and kept ready for recovery by the Backup and Recovery subsystem.

**Software Tools and Integration**

**1. User Authentication and Management Subsystem**

* Microsoft Identity Platform: For user registration, login, and authentication.
* Azure Role-Based Access Control (RBAC): To assign appropriate permissions to users.

**2. TA Application Management Subsystem**

* React or Angular: For frontend form development.
* Flask or Django: Python frameworks for backend application processing.
* Azure SQL Database: To store application data.

**3. Course Management Subsystem**

* React or Angular: For the admin interface development.
* Azure SQL Database: To store course data.

**4. Matching and Recommendation Subsystem**

* Python (Scikit-learn or similar libraries): For implementing the matching algorithm.
* Azure Logic Apps or Azure Functions: For backend automation.

**5. Decision-making and Approval Subsystem**

* React or Angular: For TA Committee Dashboard.
* Azure SQL Database: To record decisions.

**6. Notification Subsystem**

* Azure Notification Hubs: To manage and send notifications.

**7. Evaluation and Feedback Subsystem**

* React or Angular: For the instructor’s feedback interface.
* Azure SQL Database: To store evaluation data.

**8. Monitoring and Audit Subsystem**

* Azure Monitor: For system health and monitoring.
* Azure SQL Database: For storing audit logs.

**9. Backup and Recovery Subsystem**

* Azure SQL Database Backup: For scheduled database backups.
* Azure Blob Storage: For data redundancy and backups.

**Integration, Testing, and Troubleshooting Tools:**

* Git/GitHub: Version control and source code management.
* PyTest: For Python-based unit testing.
* Selenium: For automated web browser testing.
* Azure DevOps: For comprehensive CI/CD pipelines, especially when heavily using Azure services.