



SmithTA Project Plan

Client: The University of Maryland, Robert H Smith School Of Business, College Park (UMD)

BUDT723 Business Process Analysis
Professor Sindhu Rajan
By: Anwesha Gupta

Table of Contents

Executive Summary	3
Introduction	3
System Planning Phase Report	3
1.0 Client and Industry Background	3
1.1 Opportunity	4
1.2 Project Objective	4
1.3 Scope of Work	4
2.0 Project And System Requirements	4
2.1 Business Requirements	5
2.2 User Requirements	5
2.3 Functional	6
2.4 Nonfunctional	6
3.0 Project Methodology: Agile Methodology	7
3.1 Justification	7
4.0 Project Plan	8
5.0 Data Flow Diagrams	9
5.1 Context Diagram	9
5.2 Level 0 Diagram	10
6.0 Physical ERD	11
7.0 Architecture SmithTA	12
8.0 Swimlane Diagram	12
9.0 Test Scripts	14
10.0 Structure Chart	19
11.0 User Interface	20
11.1 Professor Login Screens, Desktop Site	20
11.2 Student Screens, Mobile View	23
12.0 Conclusion	25

Executive Summary

Smith TA is the proposed system described in this document. This application management system will be the one-stop solution for all the assistantship applications applicable throughout the Robert H Smith School of Business for all business school students. The main functions of the portal will be to automate TA hiring/application process, reduce professor monitoring time, improve the TA training process, and increase the number of students hired/professors assisted

Agile methodology will be used by the team due to changing requirements of the application process and administrative processes. The entire system will be developed by DataSTX Systems, an outside vendor, and will be successfully handed over to the Business School after a short run period of about 30 days from the date of delivery. All sensitive data and personal information will be strictly protected and DataSTX Systems will be provided with realistic mock data by the Business School. Since the project will be customized to the Business School's needs, all system features will be rigorously tested and a thorough demo of the complete system will be documented for all of the stakeholder's and user's benefit. This project is expected to cost a total of \$80,000.00 with a duration of 93 days from planning to implementation. Smith TA stands to be an integral part of the application processing systems at the University of Maryland (UMD) at no cost to students and professors. This portal will be viable for the long run in the future and will be a strong foundation to build upon for future application systems.

Introduction

This project aims to create a business school-wide career TA application system for the University of Maryland (UMD) Robert H Smith School of Business, College Park. This system will be called Smith TA and will serve as a portal for students, alumni, faculty, and administrative bodies like (Payroll) interested in and hiring Smith School students for positions of TA(Teaching Assistant), RA(Research Assistant), GA(Graduate Assistant).

System Planning Phase Report

1.0 Client and Industry Background

Robert Smith Business School is a well-respected institution with a proven track record of producing successful graduates. The faculty is composed of experienced professionals with advanced degrees in their fields, providing students with valuable real-world knowledge. The school's career services team is highly effective, with a large percentage of graduates finding employment shortly after graduation with high starting salaries.

DataSTX Systems is all about creating information systems that make businesses run smoother and better than ever before. We've got a talented team of developers and engineers who are all about making custom solutions that fit our client's needs like a glove. Whether we're crunching data, building software, or web

applications, DataSTX Systems offers a wide range of services to help businesses stay ahead of the curve in the ever-evolving world of technology.

1.1 Opportunity

The main objective of a portal that manages applications is to ensure hassle-free form filling and processing for all the stakeholders involved in professional employment. As of 2023, the Smith School has no formal channel or portal for managing applications for their TA, RA, and GA positions. The process is largely manual for all the stakeholders involved, especially for students and professors who encounter a large number of cold emails. To minimize chaos and the flurry of emails being sent around there is an urgent need for an application management portal that will be centralized and be connected to Testudo and Canvas for all Smith School students with a separate application dashboard showing vacant and filled positions.

1.2 Project Objective

The main objective of Smith TA is to connect students and professors to facilitate the job search process. Once implemented, Smith TA will act as a central database for assistantship job postings, payroll processes, scheduling working hours, and a monitoring application.

1.3 Scope of Work

SmithTA will be used throughout the Smith School and will be always accessible as a web portal/application. All enrolled students, even the incoming accepted students and professors will be able to use this portal as per their needs. Maintenance and upgradation of a rich UI/UX will occur quarterly and Smith school personnel will be allowed to manipulate data from the backend. As a measure to keep the system robust, all applications dated 4 years from the date of submission will be transferred to a separate master database to create an employment history record, ensuring that all the CRUD operations of the database associated with the system remain efficient.

Furthermore, this system can be scaled and used as a centralized portal to access all student-based hiring needs of the University of Maryland (UMD), instead of having individual schools within the university having their separate portals.

2.0 Project And System Requirements

To determine the necessary requirements for the system, the team held an interview with Dr. Sujin Kim, who coordinates the TA application process. Currently, she is emailed directly by students interested in becoming TAs and professors seeking TAs for their upcoming classes. Dr. Kim sends the prospective TAs a Google form to fill out. The input of the form is linked to a Google sheet which she sends to the professors seeking TAs. However, this list is not updated as TA positions are filled or student availability changes, so professors need to inquire with Dr. Kim about students' availability. The process of determining eligibility is non-standard: the professor may email the career centers of each college in order to assess current capabilities and possible improvements. During these interview days, members of the team took the time to speak to incoming students seeking career consultation at the career centers as a brief informal interview.

2.1 Business Requirements

1. Automate TA hiring/application process.
2. Reduce professor monitoring time.
3. Improve the TA training process.
4. Increase the number of students hired/professors assisted.

2.2 User Requirements

1. Prospective TAs, RAs, GAs
 - Login with SSO - To ensure that the user has to access multiple applications(Testudo, Canvas, HR System) or systems using a single set of login credentials. Only UMD credentials are accepted.
 - Submit applications - Capturing basic personal information (First Name, Last Name, Date of Birth, Tenure Semester, Tenure Year, Age, Gender, Profile Summary, Interest Tag Words, Visa Type, Enrollment Category), uploading transcripts, resume, and contact information(number, umd.edu mail, UID) and resume submission along with preference and interest on the choice of subjects.
 - See available positions - A dashboard with a search that will show no. applications open, no. of applications filled, no. of applications in process, no. of vacant positions, no. of positions filled. Users will be able to view all metrics/volumes of applications in the queue and their separate personal metrics.
 - Update availability/resumes/interest - Able to edit forms and able to visualize admission results from the dashboard.
 - Accept positions/interviews - Electronically accept positions, sign documents regarding the hiring process, grant consent for viewing time schedule.
 - Submit payroll info and Consent forms - Direct integration with the UMD payroll department for filing up of TAX forms, SSN requirements, and getting the Letter of Endorsement from ISSS and other University Departments.
 - Indicate available hours - Direct integration with Testudo to schedule working hours. This will be available to be viewed upon consent from the student to the hiring professor, to reduce time conflicts.
2. Professors and Academic Staff
 - Login with SSO - To ensure that the user has to access multiple applications(Testudo, Canvas, HR System) or systems using a single set of login credentials. Only UMD credentials are accepted.
 - Post vacancies - Allow employers to post vacancies with job descriptions, minimum qualifications required, eligibility criteria, and interest.
 - Visualize ideal candidates - employers will be able to search with multiple filters to see their best fit/ ideal candidates' applications. They will also be able to sort and rank based on various criteria.
 - Employment work history - Allow employers to see if the student has already worked for other professors/ jobs inside the university system. This will only be limited to the work done by the student after enrolling in the university.
 - Dashboard for submitted applications- Employers should be able to view the job they have posted, and applications submitted to them against the particular job. No. of positions that have been filled and which students are working under their supervision.

- Request Interview/Hire - After the employer has decided on a candidate, they will be able to request an online interview along with a request to hire if the interview is satisfactory.
 - Send info to payroll - Employers will be able to send the offer letter directly to the student and other departments like the payroll department and ISSS(International Student Services) for other administrative work.
 - View available hours - After Permission is granted by the student to view their time schedule, the employer can see the student's available hours and timings.
3. Department Heads
- Approve /Deny Hiring Requests - According to the administrative process, after finalizing the ideal candidate the employer needs to seek approval from the Department heads for a go-ahead. After the approval, the payroll process and HR process can begin.
 - Receive notifications of new requests - For every approval request the Department heads will receive a notification from the employer.
 - See summary/analytical reports of past hires- Will be able to view a dashboard with metrics of the past and present heires and how many TAs, RAs, and GAs are hired, annually.
4. HR (Human Resources)
- Process payroll info - Process student financial information as per the UMD administrative process.
 - Securely manage TAs, RAs, and GA's financial information.

2.3 Functional

1. Process
 - Filter students by availability/qualifications.
 - Notify students of selection/interviews.
 - Notify the dept head of hiring requests.
 - Notify professors of new applicants.
 - Request training module added to students' Canvas upon hire.
 - Create payroll report and send it to HR upon hire approval.
 - Request time schedule approval of testudo.
 - Display the hours/duration the student is hired for.
 - Search bar with filters for candidates and employers.
2. Informational
 - Class schedule: get from testudo/MPO- scheduler.
 - Past TA information to be displayed for a certain time - until after graduation.
-UID, Date Of Birth, Course Name for the Positions Held, Job Title, and Supervisors Name
 - Prerequisites of the hiring process for domestic and international students.
 - Payment information on an hourly basis by different departments.

2.4 Nonfunctional

1. Languages: English, French, Spanish, Chinese
2. Allow non-smith teachers to view Smith school candidates to be hired for positions outside Smith School like the School of Engineering, School of Computer Science, etc.
3. Monitor and upgrade system response speed limit.
4. Personal data security of candidates and employers.
5. Compatible with Windows & Mac, mobile OS, and platforms

3.0 Project Methodology: Agile Methodology



We intend to use agile methodology to develop our web portal.


3.1 Justification

Agile methodology is the preferred choice for developing this application, given its ability to adapt to changing circumstances, the high degree of collaboration involved, the incorporation of customer feedback, and quickly producing results. Comparison to other development models is as follows:

Ability to develop systems with...	Waterfall	Agile Development	System Prototyping
Short Time Constraint	Poor	Excellent	Excellent
Team Communication and Collaboration	Average	Excellent	Good
Flexibility/Adaptability	Poor	Excellent	Good
Unclear/ Changing User Requirements	Poor	Excellent	Excellent

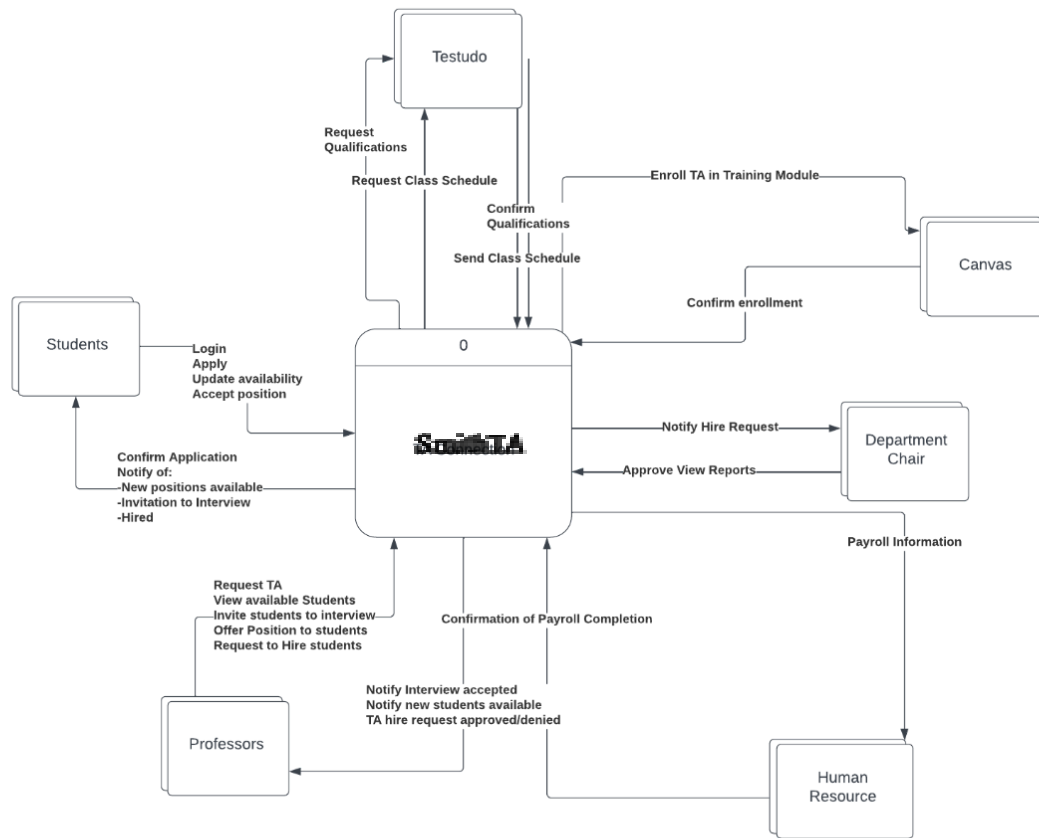
4.0 Project Plan

		WBS	Name	Duration	Start	Finish
1		1	SmithTA	93 days	4/5/23 8:00 AM	8/11/23 5:00 PM
2		1.1	Planning	4 days	4/5/23 8:00 AM	4/10/23 5:00 PM
3		1.1.1	Project Initiation Meeting	2 days	4/5/23 8:00 AM	4/6/23 5:00 PM
4		1.1.2	Communication Plans	2 days	4/7/23 8:00 AM	4/10/23 5:00 PM
5		1.1.3	Team Appointment and Delegation	2 days	4/7/23 8:00 AM	4/10/23 5:00 PM
6		1.1.4	Identify Stakeholders	2 days	4/7/23 8:00 AM	4/10/23 5:00 PM
7		1.2	Collect and analyze client requirements	15 days	4/11/23 8:00 AM	5/1/23 5:00 PM
8		1.2.1	Identify Requirements	7 days	4/11/23 8:00 AM	4/19/23 5:00 PM
9		1.2.1.1	Department Requirements	7 days	4/11/23 8:00 AM	4/19/23 5:00 PM
10		1.2.1.2	Faculty Requirements	7 days	4/11/23 8:00 AM	4/19/23 5:00 PM
11		1.2.1.3	Student Requirements	7 days	4/11/23 8:00 AM	4/19/23 5:00 PM
12		1.2.2	Department annd Payroll Integration Analysis	8 days	4/20/23 8:00 AM	5/1/23 5:00 PM
13		1.2.2.1	Develop and Approve Use Cases	8 days	4/20/23 8:00 AM	5/1/23 5:00 PM
14		1.3	Design user interface and application layout	55 days	4/20/23 8:00 AM	7/5/23 5:00 PM
15		1.3.1	Design website layout	5 days	4/20/23 8:00 AM	4/26/23 5:00 PM
16		1.3.2	Develop front-end design	10 days	4/27/23 8:00 AM	5/10/23 5:00 PM
17		1.3.3	Develop back-end design	15 days	5/11/23 8:00 AM	5/31/23 5:00 PM
18		1.3.4	Develop database for applicant information	15 days	6/1/23 8:00 AM	6/21/23 5:00 PM
19		1.3.5	Develop master database for storing applicant history	10 days	6/22/23 8:00 AM	7/5/23 5:00 PM
20		1.4	Testing of application system	7 days	7/6/23 8:00 AM	7/14/23 5:00 PM
21		1.4.1	Perform unit testing	7 days	7/6/23 8:00 AM	7/14/23 5:00 PM
22		1.4.2	Perform system testing	7 days	7/6/23 8:00 AM	7/14/23 5:00 PM
23		1.4.3	Perform performance testing	7 days	7/6/23 8:00 AM	7/14/23 5:00 PM
24		1.4.4	Perform user acceptance testing	7 days	7/6/23 8:00 AM	7/14/23 5:00 PM
25		1.4.5	Fix bugs & performance issues	7 days	7/6/23 8:00 AM	7/14/23 5:00 PM
26		1.5	Deployment and Implementation	10 days	7/17/23 8:00 AM	7/28/23 5:00 PM
27		1.5.1	Deploy application system to production	3 days	7/17/23 8:00 AM	7/19/23 5:00 PM
28		1.5.2	Train stakeholders on the system	3 days	7/20/23 8:00 AM	7/24/23 5:00 PM
29		1.5.3	Track Application Performance	2 days	7/25/23 8:00 AM	7/26/23 5:00 PM
30		1.5.4	Gather Feedback	2 days	7/27/23 8:00 AM	7/28/23 5:00 PM
31		1.6	Maintenance and Support	7 days	7/31/23 8:00 AM	8/8/23 5:00 PM
32		1.6.1	Provide technical support for the system	7 days	7/31/23 8:00 AM	8/8/23 5:00 PM
33		1.6.2	Perform regular maintenance checks	7 days	7/31/23 8:00 AM	8/8/23 5:00 PM
34		1.6.3	Perform any software updates	7 days	7/31/23 8:00 AM	8/8/23 5:00 PM
35		1.7	Project Close-out	3 days	8/9/23 8:00 AM	8/11/23 5:00 PM
SmithTA - page1						

		WBS	Name	Duration	Start	Finish
36		1.7.1	Scheduled Emergency Change Window	3 days	8/9/23 8:00 AM	8/11/23 5:00 PM
37		1.7.2	Documentation	1 day	8/9/23 8:00 AM	8/9/23 5:00 PM
38		1.7.3	Inspection & Acceptance	1 day	8/9/23 8:00 AM	8/9/23 5:00 PM
39		1.7.4	Disposition of resources	1 day	8/9/23 8:00 AM	8/9/23 5:00 PM
40		1.7.5	Project Handover	1 day	8/9/23 8:00 AM	8/9/23 5:00 PM

5.0 Data Flow Diagrams

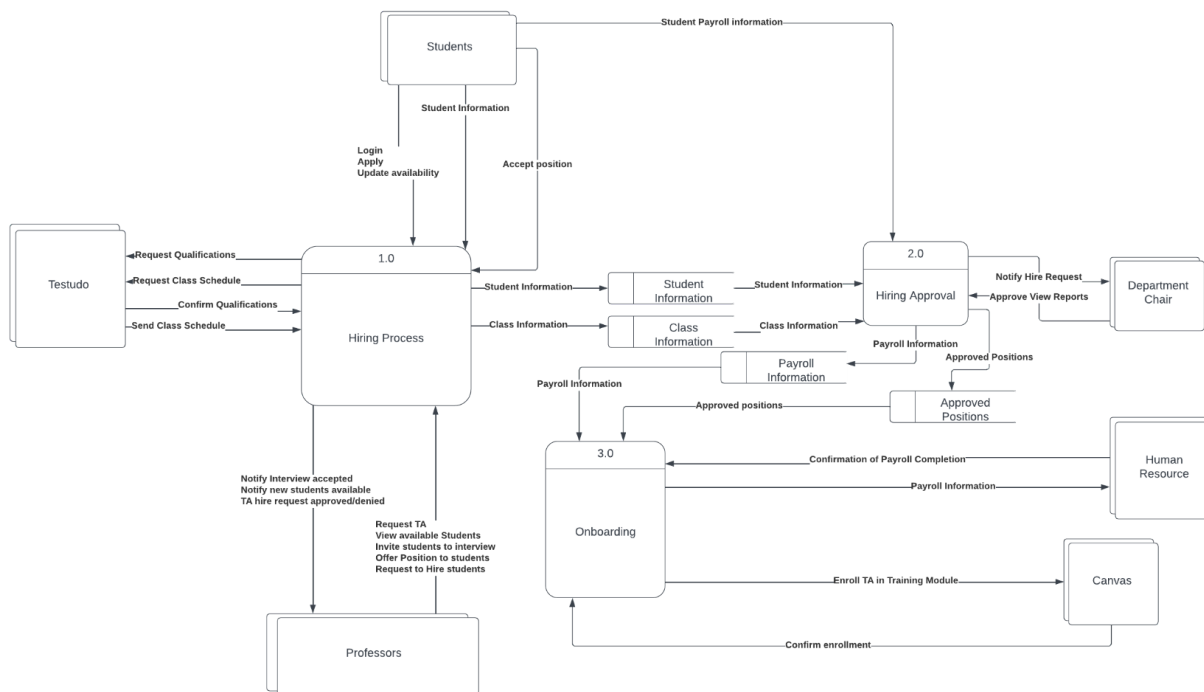
5.1 Context Diagram



External Entities:

- **Students:** Candidates for TAs, RAs, and GAs positions
- **Professors:** Represents a current professor who is looking for TAs, RAs, and GAs
- **Testudo:** Course Registration and Grade Report System
- **Canvas:** A web-based learning management system used to train TAs, RAs, and GAs
- **Department Chair:** Must approve all TAs, RAs, and GAs hiring decisions
- **Human Resource:** External Department that enrolls TAs, RAs, and GAs in the payment system

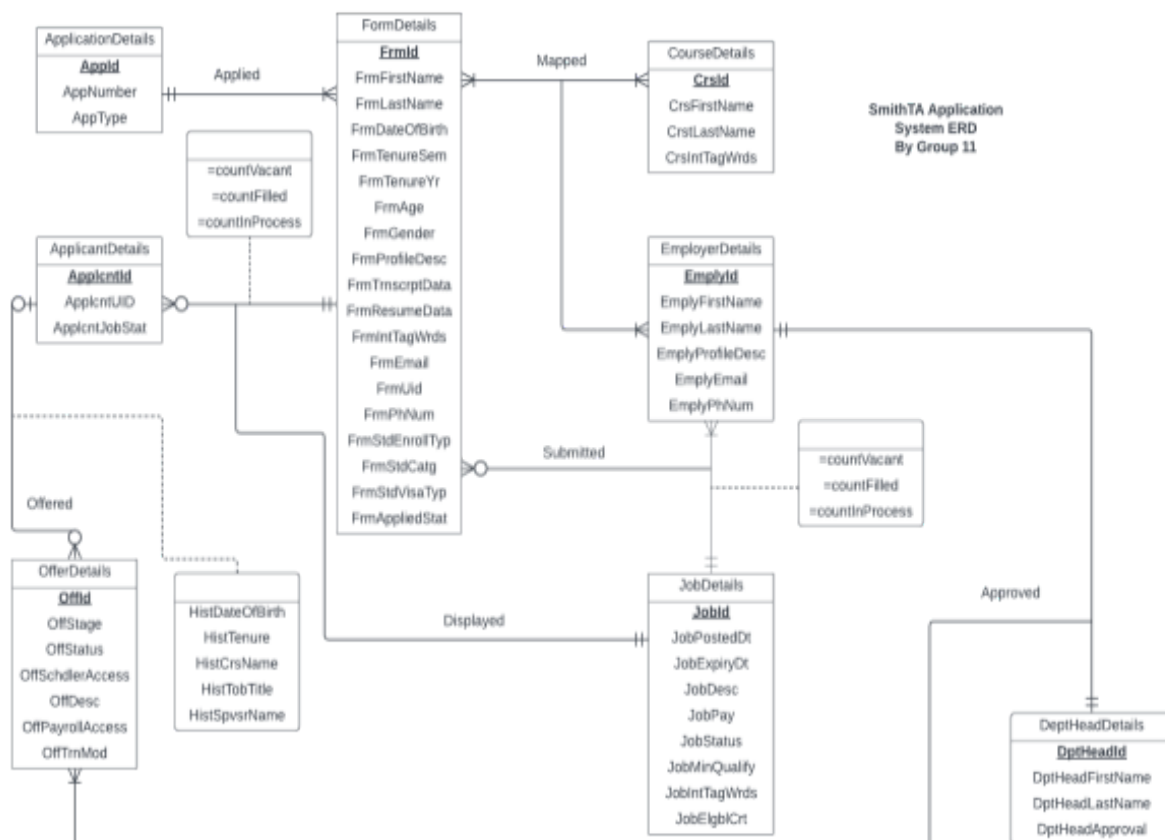
5.2 Level 0 Diagram



6.0 Physical ERD

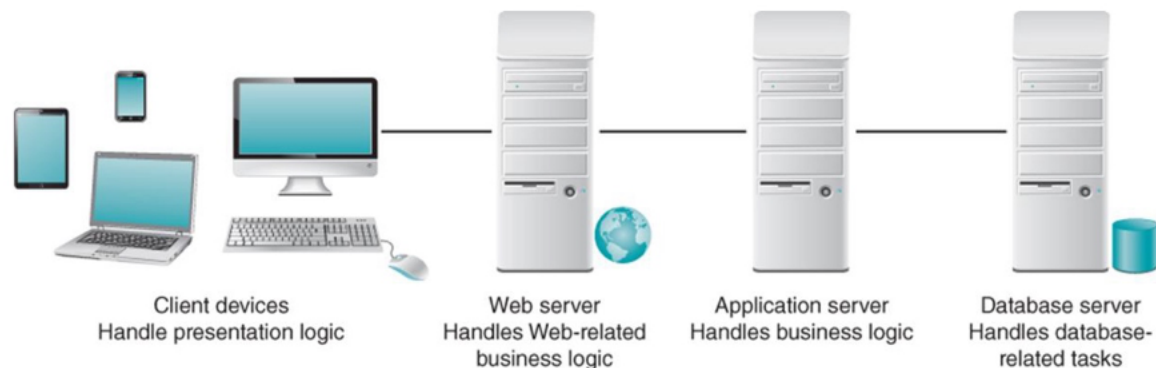
The Physical ERD of this application has 8 entities and 3 Associative or Relationship entities that will help track dashboard metrics and history of employment. There are 4 ternary relationships and 2 binary relationships present in the diagram. Each entity is described by a special ID of its own to help track the uniqueness of the record without compromising its integrity. There are several 'status' attributes present for entities and their descriptions are listed below:

- ApplicationDetails [AppType] → Will hold values 'TA', 'RA', and 'GA' for different assistantships.
- FormDetails[FrmAppliedStat] → Will hold values 'Submitted', 'In Review', 'Interview Requested', 'Pending Approval', 'Not Selected And 'Offered' for application submission.
- DeptHeadDetails[DptHeadApproval] → Will hold 'Approved', and 'Rejected' for offer approvals.
- JobDetails[JobStatus] → Will hold values 'Vacant', 'InProgress', and 'Filled' for jobs on the system.
- OfferDetails[OffStatus] → Will hold values 'Accepted', 'Declined', and 'Withdrawn' for every offer.
- ApplicantDetails[ApplcntJobStatus] → Will hold values 'Hired' and 'Available' for every aspiring applicant.



7.0 Architecture SmithTA

Design architecture: n-tiered client-server architecture



Justification:

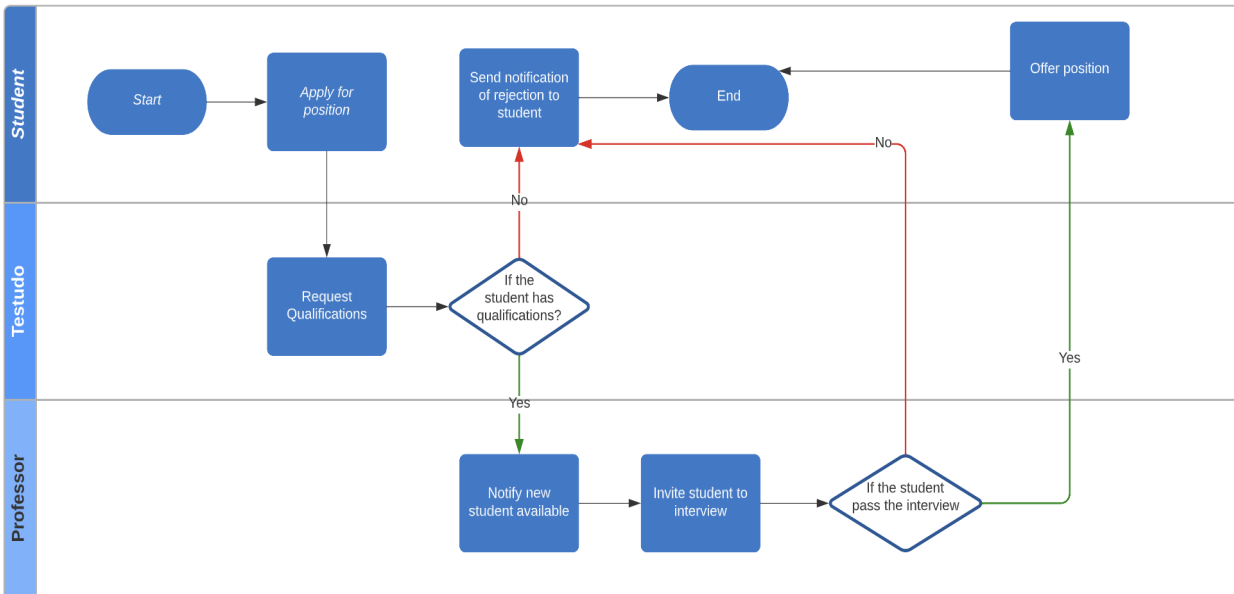
- We have different users for the system, the client-server architecture can support different types of clients and servers through middleware.
- The presentation logic, the application logic, and the data processing logic can be independent, which helps to balance the server loads efficiently.
- Client-server solutions tend to be less expensive when compared to cloud-server solutions since there is no server or electricity cost.
- The client-server architecture is suitable for systems that have separate functional requirements, which also suit our application system. It also implies that different clients and servers have their own responsibilities. Clients would be the side to manage the validation and the server side is used to execute the users' requests and return results.

8.0 Swimlane Diagrams

These are the 2 transactional swimlane diagrams.

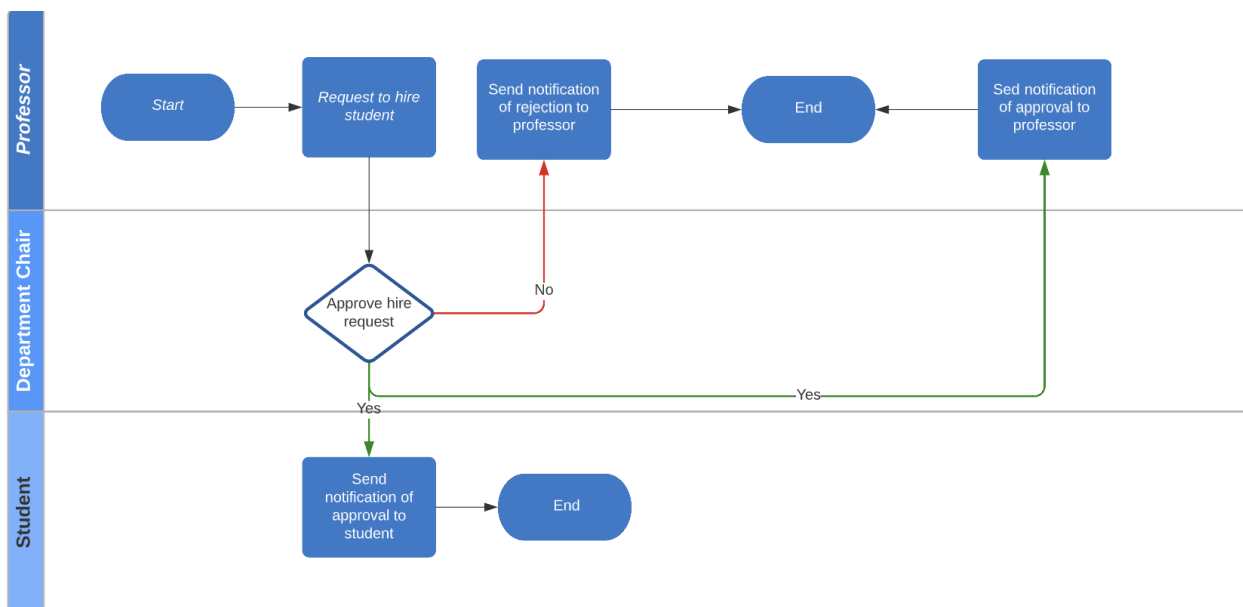
Student Application Process

- The student, Testudo, and Professor are involved in this transactional process
- It starts from student submits the application for the TA/RA/GA position and ends with either sending a notification of rejection to the student or offering the position to the student
- There are two conditional processes involved and decided by Testudo and Professor



Department Approval process

- Professor, Department Head, and Student are involved in this transactional process
- It starts from professor sending a request to hire student, and ends with either sending notification of rejection to a professor or sending notification of approval to professor
- There is only one conditional process, which will be decided by the department chair



9.0 Test Scripts

Type of Test	Scenario	User Input	Expected Result
Stub Test	Testing functionality for retrieving candidate information from the database	Candidate UID or name	The function should return the candidate information corresponding to the provided UID, such as name, contact information, and qualifications
Stub Test	Testing functionality for retrieving qualified candidates by course	Course ID or name	The function should return the candidate details corresponding to the provided course ID
Stub Test	Testing functionality for sending email notifications to users	Email address of user	The function should simulate sending an email notification to the provided recipient email address and record the action

Stub Test	Testing functionality for checking application status	Application ID	The function should return the status of the application corresponding to the provided UID, such as 'Submitted', 'In Review', 'Interview Requested', 'Pending Approval' and 'Offered'
Unit Test	Testing the function to validate user login credentials	Username and password	The function should validate the credentials and return a success status if they are correct
Unit Test	Testing the function to submit a new application	Application form data	The function should save and submit the application data and return a success status
Unit Test	Testing the function to update availability/resumes/interest	Updated availability, resume, and interest information	The function should update the corresponding data and return a success status

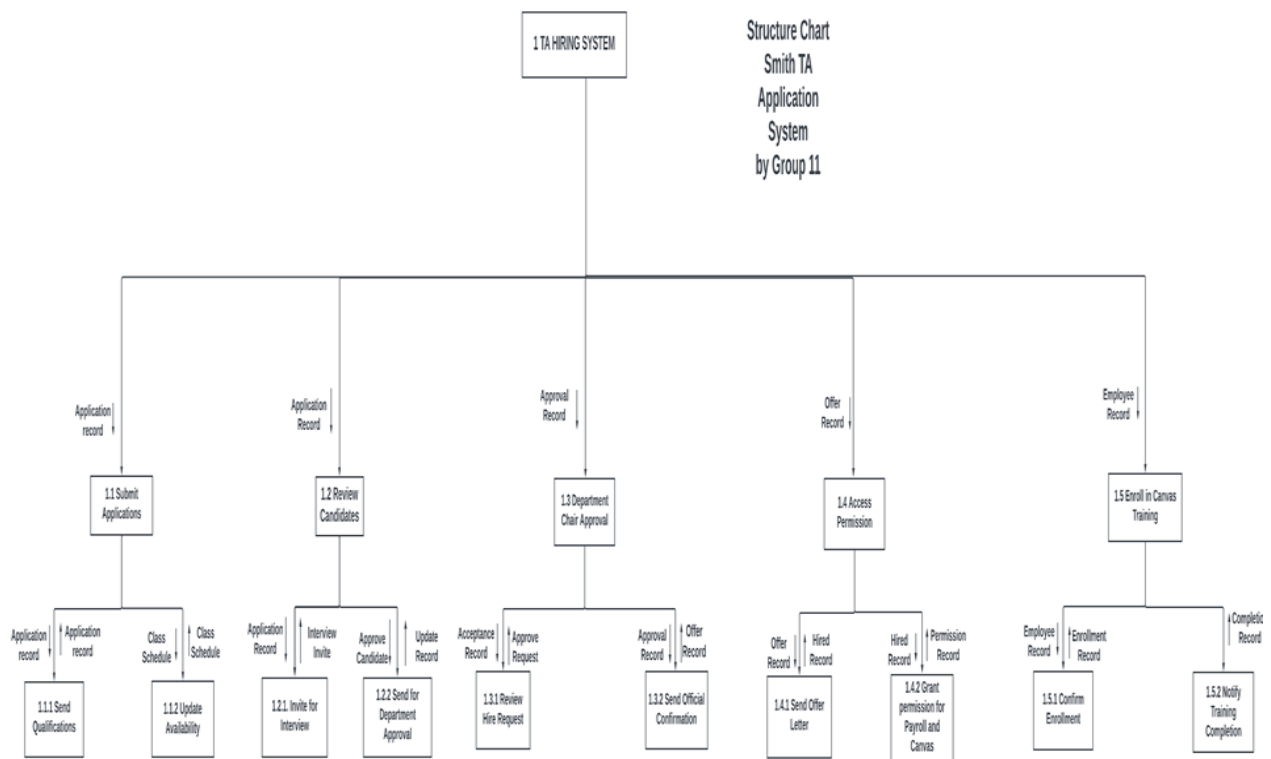
Unit Test	Testing the function to accept a position/interview	Acceptance of position/interview	The function should update the acceptance and return a success status
Integration Test	Testing the integration between the payroll processing and application approval	Status check of candidate application	Upon approval of candidate application, it should move ahead to payroll processing with updated status
Integration Test	Testing the integration between the candidate dashboard and job posts functionality	User searches for open jobs	When user searches for open positions, their dashboard should display all relevant jobs posted by employers
Integration Test	Testing the integration between the employer dashboard and candidate selection	Employer accepts candidate for the position	When employer accepts and hires the candidate, the employer dashboard should update with the selected candidate information for the position

Integration Test	Testing the integration between the application submission and candidate search functionality	Candidate searches for their submitted applications	When candidate searches for only their submitted applications, the function should retrieve all the relevant applications
System Test	Testing the end-to-end hiring process for a prospective TA	Complete application data, position acceptance, and payroll information	The entire process should flow smoothly, and the TA should be successfully hired
System Test	Testing the end-to-end hiring process for a professor posting a job and selecting a candidate	Complete job posting, initiate hiring process, invite for interviews, and approve selections	The entire process should be seamless, and result in the professor hiring the candidate for the job
System Test	Testing the end-to-end hiring process for a department head approving a hiring request	Hiring request details, and approval decision	The department head should be able to review the hiring request, provide approval or denial, and the system should update the status accordingly

System Test	Testing the end-to-end hiring process for the HR department processing payroll information	Payroll data, and payroll processing	The HR department should be able to process the payroll information, generate payroll reports, and ensure accurate payment to hired candidates
Stress Test	Simulating high concurrent user traffic	Multiple concurrent user requests to access the system	The system should handle a high volume of concurrent user requests without significant performance degradation or crashing.
Stress Test	Testing the system's scalability	Increasing the number of users, applications, and job postings beyond the normal capacity	The system should be able to handle a significant increase in user load, applications, and job postings. It should scale dynamically and efficiently allocate resources to maintain optimal performance and response times.

10.0 Structure Chart

In the proposed SmithTA Application System, the professor posts an open job application for the position to invite candidates to apply. The candidates, on the other hand, can log into the system and submit the application for the open positions. Additionally, they can also send further details like their transcript, resume, or class schedule to adjust availability. The professor then reviews all candidates and sends an interview request to the qualified candidates to initiate the final screening process. If the candidate is selected, he/she is then offered the position through an official mail after approval from the Department Head. However, if the candidate is not selected or rejected for hiring by the Department Chair then, they are notified of the same through mail. On the other hand, if the candidate accepts the offer, the application is further forwarded to the payroll team for payroll approval and processing. However, if the candidate fails to accept the position, the offer is rescinded and the position remains open for hiring. Also, he/she is further enrolled in any required training requested by the professor for the course. On completion of the training, the professor is notified of the same, and the TA is now officially hired. See the [link](#) for the structure chart.



11.0 User Interface

11.1 Professor Login Screens, Desktop Site ([Drive Link](#))

- Showing Login Screen, Search Page, HomePage and Dashboard in order

150 Assistants Placed

75 Open Positions

82 Available Students

Sign In

UNIVERSITY OF MARYLAND

Central Authentication Service (CAS)

e

DIRECTORY ID

PASSPHRASE

0

DON'T REMEMBER LOGIN

LOG IN

For security reasons, please log out and exit your Web browser when you finish accessing services that require authentication.

The Division of IT will never ask you to put your passphrase into an email message, but scammers will.

Do not share your passphrase with others!

[Forgot Your Passphrase?](#) / [Forgot Your ID?](#) / [Need help?](#)


[Home](#)
[Past Courses](#)
[Applicants](#)
[New Listing](#)


Search for TAs


Filter By


☐ Courses
☐ Availability
☐ Major


☐ View Details
☐ Email
☐ Submit for approval
☐ Decline



Hudson Gibson
MS Information Systems



Andrea Shultz
PhD Accounting



Willow Jones
MS Supply Chain Management



Hector Norris
MS Business Analytics



Raegan Brooks
MS Business Analytics



Christian Martin
MS Finance



Julius Foster
MS Marketing Analytics


Rylie Thompson
MS Information Systems


Gabriela Johnson
MS Supply Chain


Ben Douglas
MS Quantitative


Nova Hoffman
MS Business


Samuel Waters
PhD Economics

[Home](#)
[Past Courses](#)
[Applicants](#)
[New Listing](#)

Search for TAs

Welcome Back Sindhu!


☐ Interview Scheduled
BUDT 724
Omsri Yarram


☒ Interview Scheduled
BUDT 724
Afia Simeen

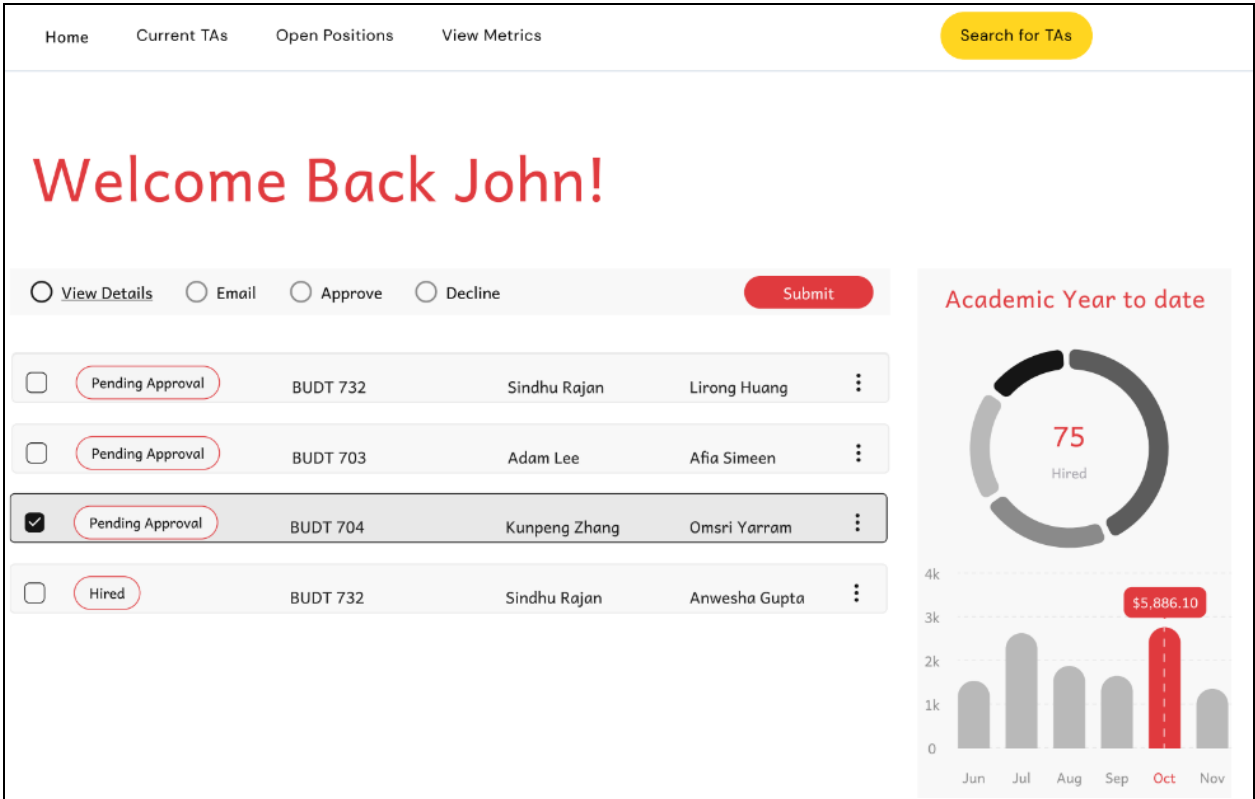
☐ Pending Approval
BUDT 732
Lirong Huang

☐ Hired
BUDT 732
Anwesha Gupta

New Applicants


Johannah Ryan
MS Information Systems


Jackson Davis
MS Information Systems



11.2 Student Screens, Mobile View ([Drive Link](#))

- Showing Login, Application Form, Search, Notification, Home Screens in order

150 Placed


75 Open Positions

82 Seeking

Sign In

UNIVERSITY OF MARYLAND

Central Authentication Service (CAS)



DIRECTORY ID

PASSPHRASE

☐ **DON'T REMEMBER LOGIN**

LOG IN

For security reasons, please log out and exit your Web browser when you finish accessing services that require authentication.

Home Search Profile

Name

Anwesha Gupta

Email

an*****@umd.edu

Gender

UID

Select Program >

Select Start Term >

Home Search Profile

Interest Areas

Information Systems ⊗ SQL ⊗

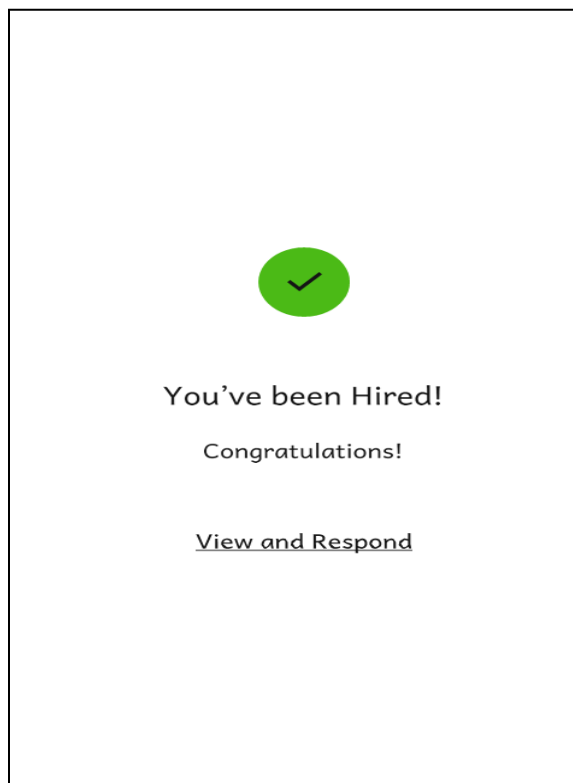
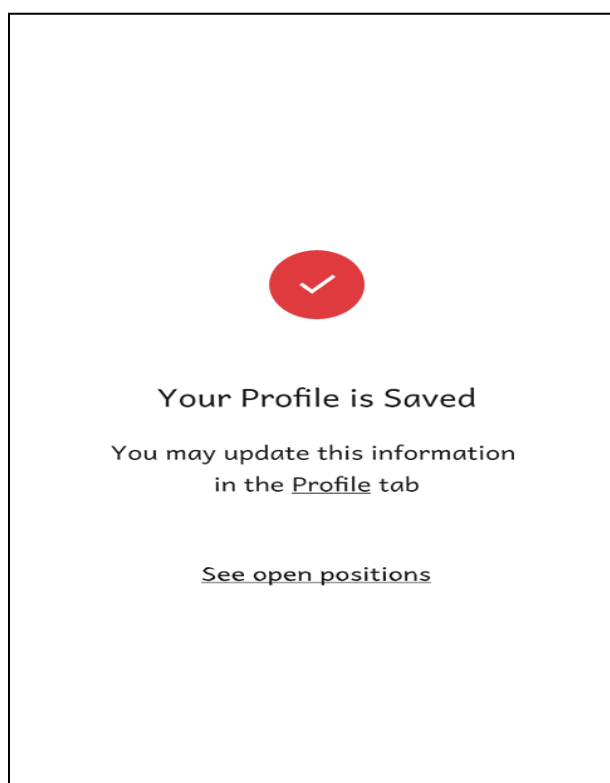
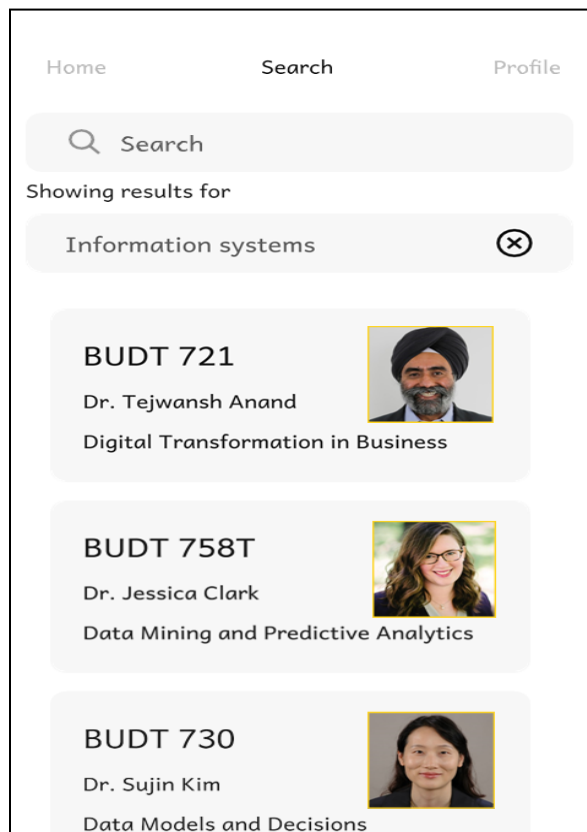
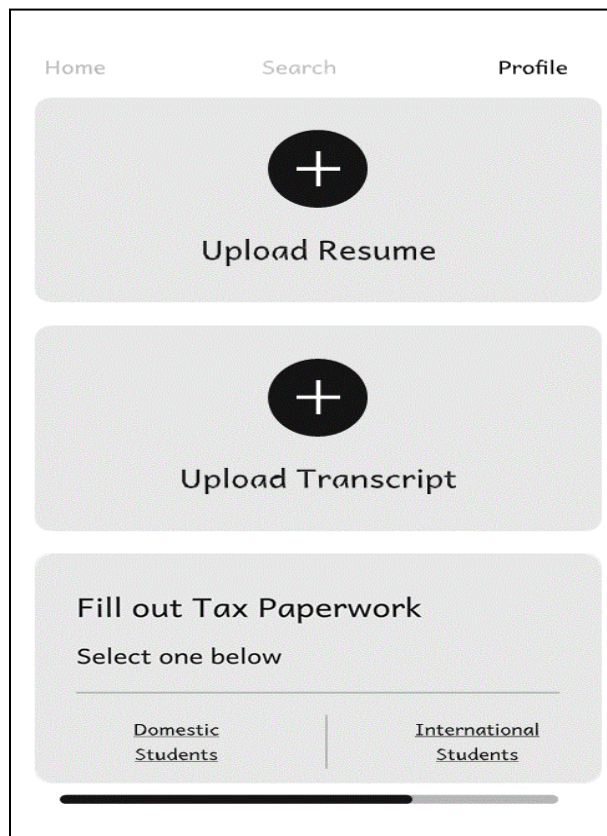
Python ⊗

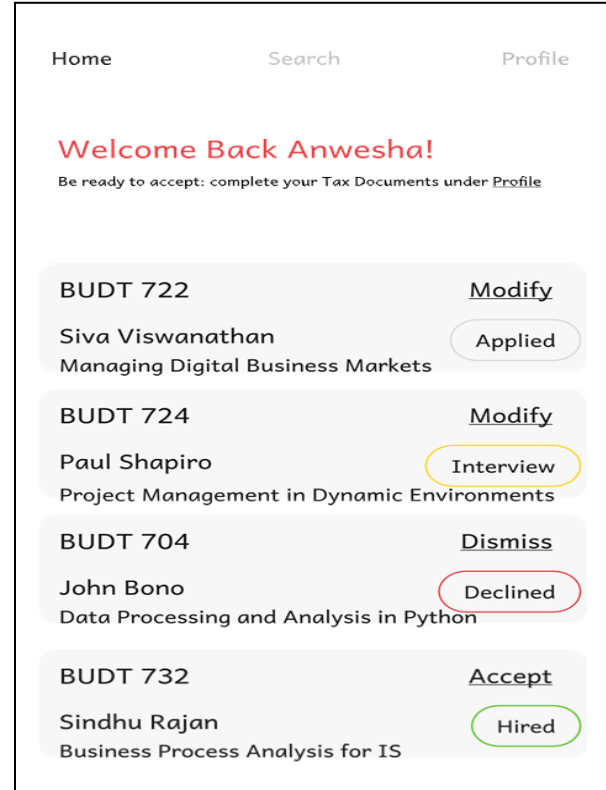
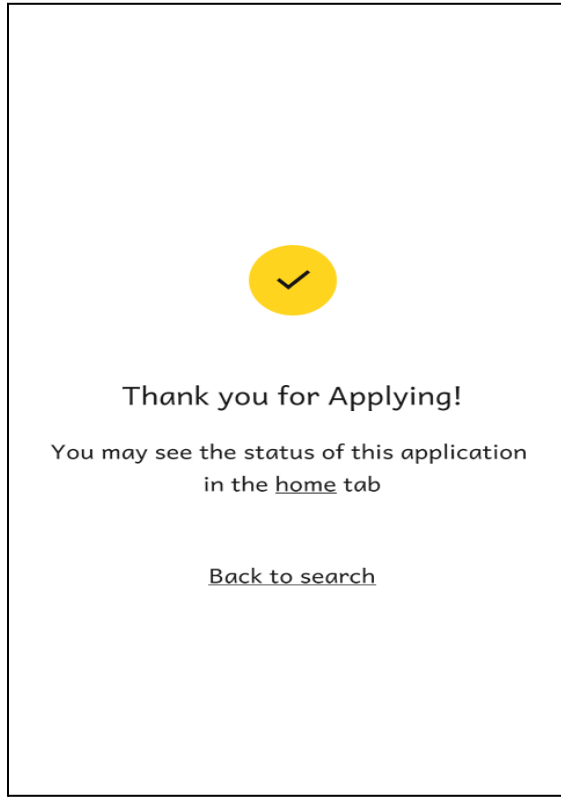
Consider me for All Positions ☐

Availability

January 2022 > < >

Mon	tue	wed	thu	fri	sat	Sun
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					





12.0 Conclusion

The Smith TA system is an essential step towards creating a centralized, streamlined process for all assistantship applications in the Robert H Smith School of Business. With its automated hiring/application process, reduced monitoring time, and improved training process, the Smith TA portal will serve as a valuable resource to students and faculty. Smith TA will be an integral part of the application processing systems at UMD, and all enrolled students and professors will be able to use it. DataSTX Systems will develop the portal using the Agile methodology, ensuring flexibility and adaptability to changing requirements. The system's scope of work includes maintenance and upgradation of a rich UI/UX, which will occur quarterly. Overall, the Smith TA system is a strong foundation for future application systems at the University of Maryland (UMD).