IS 436 Structured Systems Analysis and Design (02.1207) SP2023

Final Integrated Capstone Project Deliverable

Team 4

Immama Asif

Muhammad Baig

Dexter Kong

Liliana Majano Santos

Mohammed Ullah

Jason Zicha

Kameron Smith

Due Date: 5/16/2023

Contents

Executive Summary

Our team of IS 436 Systems Analysts decided to partner with the UMBC Division of Information Technology for our capstone project at UMBC. More specifically, we engaged the Department of Audio-Visual Services, which provides classroom support to a variety of administrative and academic units on campus. The primary function of the AV services team is to manage the audiovisual technology in all registrar-controlled classrooms at UMBC. The current business problem AV services faces is that the ticketing system needs to be updated to serve better faculty, staff, and students who require on-site assistance in classroom settings. The ticketing system needs to be updated due to the reorganization of the RT system. It takes time to communicate and know which ticket to prioritize due to the urgency of ensuring that its technology and design are far more efficient and practical to utilize and save its data on. The RT ticketing system hasn't been updated since it was last updated five years ago. In light of the pandemic, the rise in dependence on virtual problem-solving, and technological advancements, the department needs to enhance this system and give resources for a better way to manage the RT ticketing system. Our proposed solution is to create a far more convenient workflow that can be improved by creating a system that determines the types of tickets they will receive before creating a ticketing system and identifying the types of tickets they will receive. These changes will help designate additional categories and workflows. It is also defining ticket categories. Once identified the types of tickets that will be received, create specific categories for each type. For example, if they have a malfunction, they could have categories such as "Audio, Touch panel, Camera, Mic, Projector" Also, create a workflow for each ticket category that outlines the steps involved in addressing each ticket. For example, for a "Create a ticket" form, the workflow might include steps such as "Subject," "Determine the cause of the issue," and "Provide a fix."

and they can establish priority levels to determine the priority levels for each ticket category based on the severity of the issue and the impact on their customers. These changes will help prioritize tickets and promptly address critical issues. Develop a ticket assignment system for assigning tickets to specific team members based on their expertise and workload. This will help ensure that the appropriate team members handle tickets and that workload is distributed evenly. An important aspect of our student-documented interpretation of this information system is to allow AV specialists to monitor and analyze ticket data continuously to identify trends, common issues, and areas for process improvement. Use this data to refine their ticketing system and improve customer satisfaction. This method will improve UMBC staff efficiency and data organization. A better and more improved ticketing system can help enhance a more convenient workflow. It will also improve time management regarding addressing and solving issues with RT tickets. It will also help UMBC staff to assist better and organize their data. The workers will be more organized to have their data in one place and can access it conveniently with the help of this system. Deliverable 1 includes the Systems Request, which summarizes the business need and motivations to initiate the project. It also assigned deliverables to each team member. Deliverable 2 includes a formalized requirements specification for the information system, like Functional Requirements and Non-Functional Requirements. It has the Interview Guide with UMBC audio visual managers and the artifact analysis report with use cases. Deliverable 3 includes a process model which consists of a context diagram, level 0 diagram, level 1 diagrams, and the Data Dictionary Definitions. Deliverable 4 includes the architectural design of the system with Data Modeling, Entity Relationship Diagrams. Developed an alternative matrix portraying two design alternatives, Alternative Matrix 1 and 2. Developed a Hardware and Software

Specification for the chosen architecture. Deliverable 5 includes user interface design principles, physical process & data models, and five interface design prototypes.

1.0 Deliverable 1

1.1 Purpose

Business:

UMBC Division of Information Technology
Engineering ENGR020

Project Sponsor

Steve Anderson

Manager Audiovisual Specialist.

UMBC Division of Information Technology

Steve is the manager of the department of Audio-Visual Services. This department of UMBC provides audiovisual support to administrative and academic units throughout the University. The following services manage the audiovisual technology present in all registrar-controlled classrooms. Managing the operation, including software installation, of all registrar-controlled computer classrooms. Manage the software operation for all staff and faculty computers supporting day-to-day duties. Provide support for virtual desktop software to remotely access our computer lab software from anywhere you have an internet connection. Provide consulting support for the development of active learning, hyflex, and synchronous lecture capture spaces on campus; and oversee support for video conferencing.

Email: Steven.Anderson@umbc.edu

Problem Summary

The department at the University of Maryland Baltimore County (UMBC) and Audio-Visual services which are under DoIT (Division of Information Technology) has been facing challenges when it comes to the RT ticketing system.

UMBC DoIT (Division of Information Technology) needs to provide the Audiovisual services with a better and more convent update. This will help our staff solve issues and keep their data neater and more organized.

Business Needs

UMBC Audio Visual services and DoIT (Division of Information Technology) needs to update their ticketing system to be able to better assist the staff, professors, and students when it comes to on-site help in classrooms.

Business Requirement

Using a better and more improved ticketing system can help improve a much more convenient workflow. It will also improve time management when it comes to addressing and solving issues with RT tickets. It will also help UMBC staff to assist better and organize their data.

Business Value

UMBC Department of DoIT and engineering needs to update its technology and ticketing system. By making sure its system and technology are much more organized and convenient to use and store its data. The RT ticketing system has not been updated in the past 5 years. However, now since the pandemic and relying more on solving issues virtually and advancements in technology, the department needs to update this system and fund a better way to keep the RT ticketing system up to date.

The Tangible benefits would be:

- Upgraded ticketing systems for UMBC DoIT staff.
- Adjustments to organizing data for all the rooms on campus
- Time management when solving and organizing issues.
- More tangible benefits will include post-interviews.

The intangible benefits would be…

- Updated and training for both faculty and staff to send requests or use the system.
- Less confusion in communication when addressing the issues.
- The ability for staff to easily communicate issues in a more organized manner.
- DoIT is on campus, meaning that communication is quick and easy in case of any questions, repairs, concerns, or upgrades.
- Less stress for professors to communicate their issues and also for staff to understand the issue.
- Students lose less lecture/lab time due to better time management and solving issues in a timely manner when issues arise.

Special Issues or Constraints

No resources are needed outside of DoIT UMBC Staff.

Project Assumptions

We will have a better understanding of the assumptions of the RT ticketing system as we continue the project and the interview process. As of right now, we assume that we need to create a better-organized ticketing system.

- We can add a location quo outdated field.
- Need to know which one is a major or less important ticket.
- add # (tags) to relate the issues.

- We need to create a timer that shows the exact timing of when AV receives the most alerts and tickers. This can help determine how many staff should be on site to help assist in the correct, timely manner.

Preliminary Project Recommendation

By using this system, DoIT employees and UMBC will be able to access it easily, and the staff will be more organized to keep their data in one place, especially when it comes to time management. The audio and visual team will identify and develop a solution to the RT ticketing system (more about this once we complete our interviews). As of right now, our team will conduct two interviews with two different people to talk about the issue that arises in the department. Afterward, we will set times/days to discuss the possible solutions. This capstone project will be done by the end of the semester with a completely analyzed solution for the RT ticketing system.

1.2 Deliverable Roles/Bio

1.2.1 Deliverable 1

Role:

The lead of deliverable 1 is Immama Asif, and the Admin is Muhammad Baig. We have contacted the supervisor and got two interview dates confirmed and posted to everyone's Google calendar with added Webex links. The dates confirmed are 02/22/23 and 02/23/23. I also created a deliverable 1 report stating the contents of the project.

Lead: Immama Asif

Bio: I'm a senior majoring in information systems. I enjoy hikes and spending time with family and friends. I am the primary lead for this project. I will be doing deliverable 1 and admin for deliverable 2.

Email: Immamaa1@umbc.edu

Admin - Muhammad Baig

Bio: I am an IS major and in my last year. My hobbies are gaming, working out, and watching anime. I am the lead for deliverable 2 and an admin for deliverable 1.

Email: Mbaig1@gmail.com

1.2.2 Deliverable 2

Role: Muhammad will be the lead for deliverable D2. He will draft for deliverable D2 inputs. Immama is the admin for deliverable D2.

Lead: Muhammad Baig

Bio: I am an IS major and in my last year. My hobbies are gaming, working out, and watching anime. I am the lead for deliverable 2 and an admin for deliverable 1.

Email: Mbaig1@gmail.com

Admin - Immama Asif

Bio: I'm a senior majoring in information systems. I enjoy hikes and spending time with family and friends. I am the primary lead for this project. I will be doing deliverable 1 and admin for deliverable 2.

Email: Immamaa1@umbc.edu

1.2.3 Deliverable 3

Role: Mohammed Ullah will be the lead, and Liliana will be the co-lead. Kameron will be the admin, and Dexter will be the co-admin.

Lead: Mohammed Ullah

Bio: I'm an information system major graduating after this semester. My hobbies include watching sports, playing video games, and working out. I am the lead for deliverable 3 and an administrator for deliverable 5.

Email: mullah2@umbc.edu

Co-Lead: Liliana Majano Santos

Bio: I am an IS major that will be graduating this semester as long as I pass all of my classes. My hobbies include streaming, playing video games, and reading manga. I am the team co-lead for deliverable 3 and the co-admin for deliverable 5.

Email: lmajano1@umbc.edu

Admin - Kameron Smith

Bio: I am a senior information system major My hobbies include reading, walking, and chess. I am the administrator for this deliverable and the Team Lead for deliverable 4.

Email: ksmith18@umbc.edu

Co-Admin: Dexter Kong

Bio: Hi, My name is Dexter Kong. I am a senior Information Systems major at UMBC. In my free time, I enjoy playing and developing video games. I am a team lead for deliverable 5 and an admin for deliverable 3.

Email: dkong1@umbc.edu

1.2.4 Deliverable 4

Role: Kameron is the Lead, and Jason is the Admin

Lead: Kameron Smith

Bio: Hello, my name is Kameron Smith. I am a senior information systems major. My hobbies include writing, reading, and playing chess. I am the Team Lead for this deliverable as well as the administrator for deliverable 3.

Email: ksmith18@umbc.edu

Admin - Jason Zicha

Bio: Hello! My name is Jason Zicha. I'm a senior Information Systems major at UMBC. I work in the commercial, automotive industry and enjoy being in the garage with my friends in my free time. I am the administrator for deliverable 4.

Email: jzicha1@umbc.edu

1.2.5 Deliverable 5

Role: Jason will be the lead, and Dexter will be the co-lead. Mohammed Ullah will be the admin, and Liliana will be the co-admin.

Lead: Jason Zicha

Bio: Hello! My name is Jason Zicha. I'm a senior Information Systems major at UMBC. I work in the commercial, automotive industry and enjoy being in the garage with my friends in my free time. I am the team lead for deliverable 5.

Email: jzicha1@umbc.edu

Co-Lead: Dexter Kong

Bio: Hi, My name is Dexter Kong. I am a senior Information Systems major at UMBC. In my free time, I enjoy playing and developing video games. I am a team lead for deliverable 5 and admin for deliverable 3.

Email: dkong1@umbc.edu

Admin - Mohammed Ullah

Bio: I'm an information system major graduating after this semester. My hobbies include watching sports, playing video games, and working out. I am the lead for deliverable 3 and an administrator for deliverable 5.

Email: mullah2@umbc.edu

Co-Admin: Liliana Majano Santos

Bio: I am an IS major that will be graduating this semester as long as I pass all of my classes. My hobbies include streaming, playing video games, and reading manga. I am the team co-lead for deliverable 3 and the co-admin for deliverable 5.

Email: lmajano1@umbc.edu

2.0 Deliverable 2

2.1 Functional Requirements/Non-Functional Requirements:

Functional requirements refer to the specific actions or tasks that a system or software must perform to meet the needs of its users. They are typically related to the core functionality of the system, such as processing data, performing calculations, or generating reports. Non-functional requirements, on the other hand, are characteristics or qualities that describe how the system should perform, rather than what it should do. These can include factors such as reliability, performance, security, and usability. In software development, it is important to clearly define and prioritize both functional and non-functional requirements to ensure that the final product meets the needs of its users and stakeholders.

2.1.1 Functional Requirements:

Process-Oriented:

1. The RT ticketing system shall have a boolean search engine that can quickly locate tickets based on building codes and room numbers. This feature is important because it enables the AV department to locate and resolve issues in specific locations on campus quickly and efficiently.

2. The system shall have clear and concise communication capabilities to keep customers updated on the status of their tickets and the steps taken to resolve the issue. This requirement is essential to provide excellent customer service and ensure customers are informed of progress throughout the issue resolution process.

3. The system shall prioritize tickets based on the urgency and importance of the problem so that Management can allocate resources and manage workload effectively. This requirement is crucial for efficient issue management, as it ensures that the most urgent and important issues receive prompt attention.

4. The system shall have a mechanism for identifying recurring issues and proactively addressing them. This requirement is vital as it enables the AV department to reduce the number of recurring issues and improve service delivery.

Information-Oriented:

5. The system shall be designed specifically for IT environments, not for general campus use. This requirement is essential because the IT department needs a system tailored to its needs and requirements.

6. The system shall integrate with other tools and software, such as R25 and Web Admin, to help manage workloads and resolve issues. This requirement is necessary to improve efficiency and streamline processes across multiple systems.

7. The system shall allow for collaboration and communication with other departments and teams within the organization, such as HR and Finance, to ensure efficient issue resolution. This requirement is crucial for promoting collaboration and ensuring a streamlined and efficient service delivery process.

2.1.2 Non-Functional Requirements:

Operational:

1. The system shall be easy to use and navigate.

2. The system shall have high availability and reliability to ensure tickets are processed and resolved in a timely manner.

3. The system shall be scalable and able to handle a large volume of tickets.

Performance:

4. The system shall have fast response times to ensure efficient ticket resolution.

Security:

5. The system shall have secure login and access controls to prevent unauthorized access to sensitive information.

Cultural/Political:

6. The system shall be sensitive to the culture and politics of the organization, such as adhering to relevant policies and regulations.

2.2 Interview Purpose/Objectives:

The purpose of this interview is to gather requirements for a new mobile app that we are planning to develop. The objectives of the interview are to understand the user's needs, preferences, and pain points, as well as to identify the features and functionality that would be most valuable to them.

2.3 Interview guide:

2.3.1 Interview 1 - Date, Time, and Location of Interview:

Date: 2/22/23

Time: 10:00am

Location: Webex

Name of Interviewer:

Immama Asif

Name and Title of Interviewee:

Steve Anderson Manager

Interview Questions:

I. Introduction

Can you tell me a bit about yourself and your background? (Open-ended)

Steven Anderson, Visual Arts ‘87, joined the former Instructional Technology department in 1985 as a student employee in AV Services. He also worked as a student employee for UMCP Facilities Management during semester breaks. Following graduation, he worked as a carpenter’s assistant, spent a month traveling and photographing in Europe, and worked as a portrait photographer. From 1988-1990, he served as a photographer for Md Institute for Emergency Medical Services (Shock Trauma Center). During 1990-1992, Steven worked for a pioneering electronic imaging company to help create the first computerized image databases at National Geographic and the Library of Congress. Tired of the DC commute and wishing to work on a master’s degree, Steven returned to UMBC in 1992. He served as an AV Technician, Supervisor, and now Manager. As part of the Division of Information Technology, Steven has worked across departments to shift to installed AV systems in all classrooms and served on many new building and renovation teams. He and the DoIT AV Services team are focused on ensuring campus AV systems are intuitive, reliable, and relevant to users’ needs. Steven is proud that he, his wife Juli, his son Ian, and his daughter-in-law Rosie are all UMBC alums. Interests include photography, music, outdoor activities, and horseback riding.

II. User Needs

What are some of the challenges you face in your work that you think our team/group could help with? (Open-ended)

Getting useful information from the service ticket request system to inform planning and decision making. For example, it helps to know how often and why we visit particular rooms. Also, it would help to understand how service requests are trending for similar topics so we can get ahead of problems.

How do you currently manage [Insert problem that our group is solving]? (Probing)

We have to run reports based on keywords and/or requestors.

What would you say is the most important thing that our team could do to help you with [Insert problem that our group is solving]? (Open-ended)

It would help to have a guided process to encourage the requesters to put in useful information in a standard form, while also suggesting various self-help (FAQs) along the way where appropriate. The information entered should facilitate easy analysis, spotting trends, and have the potential to set up custom dashboards depending on the staff person's role (tier I, II, III, trends, time open, etc.)

III. System Features and Functionality

In your opinion, what are the most important features that a Ticket system should have? (Open-ended)

Ease of use for user and staff, guided request entry, smart routing of requests, easy search, and reporting.

Can you walk through the stages of the process that they follow when creating a ticket or searching?

When creating a ticket, you go to the RT website and login by your credentials then to your top right you click create new ticket. There are three boxes one is "Create a new ticket in DoIT-INFORMATION TECHNOLOGY" this is where you put the requestor email and subject line there is also a "include article" which we don't use. Then in the description you add the issues. The second box is "Basics" where there is queue which is Doit then there is status with the options of "new, open or resolved" and then owner where the creator can add or assign this ticket by email. The last box is custom fields depending on the Queue it changes.

How do you prioritize the requests that come in? Also, what factors do you consider when determining which requests to address first? (Open-ended)

Tickets get routed based on assigned staff roles and specialties when possible. Due to the small size of the unit, advanced staff may at times work on Tier I tickets. In general, support of in-progress instruction takes top priority. Room and/or requestor availability determines when other service requests can be served.

Are there any features or functionality that you would like to see in this system that currently doesn't exist in any other similar apps? (Open-ended)

Interview for the user to give more info about the ticket. User to notice the ticket was looked at. Easy way to find out trend either through hashtags or word clouds. Easy way to generate statics with each ticket reports.

Can you describe any security or privacy concerns you may have regarding a mobile app for request tracking? (Open-ended)

Not for my unit, since DoIT has service queue access based on assigned roles.

How would you want those concerns to be addressed? (Follow-up)

Replacement of the ticket system, create a more user friendly to make easier for the user and support staff/managers, Also have meeting people in a specific time (others should be notified through calendar).

IV. User Experience

How important is ease of use to you when it comes to the ticketing system? (Closed-ended)

Very important. If it is not easy to use, it will be bypassed.

What is some design or user experience elements that you really like in other ticket systems? (Open-ended)

Guided data entry, required fields, suggestion of self-help resources related to current issue.

Is there anything that you find frustrating or confusing when it comes to submitting a ticket request? (Open-ended)

It is too easy to enter incomplete or incorrect information, or to get it routed to the wrong unit.

How do you see a RT fitting into your long-term goals for improving your work? (Open-ended)

RT should make it easier for staff to gather useful information to respond to clients, recognize trends, and be proactive. Ticket systems tend to be reactive products, focused on what went wrong.

How do you typically collaborate with other team members or departments when addressing requests? (Closed-ended)

Within DoIT we typically use Webex direct messaging as well as Spaces to get additional assistance as needed.

How could a RT facilitate that collaboration? (Follow up - Open-ended)

Flagging related departments in a ticket would be helpful - some tickets require input from more than one unit but must now sit in only one queue at a time.

V. Wrap-Up

Is there anything else that you would like to share with us that we haven't covered yet? (Open-ended)

He suggested that the American University had a really good ticketing systems which UMBC is looking into.

Is there another person you recommend us to interview regarding this issue that will help give us their opinion and perspective?

Dave Souder worked on *remedy/part of doit. Email:* souder@umbc.edu

2.3.2 Interview 2 - Date, Time, and Location of Interview:

Date: 3/03/23

Time: 10:00am

Location: Webex

Name of Interviewer:

Immama Asif

Name and Title of Interviewee:

Dave Souder

Interview Questions:

I. Introduction

Can you tell me a bit about yourself and your background? (Open-ended)

Dave Souder, Coordinator of AV service. Managed a big AV company in the Baltimore area.

Majored in theater. In the industry since 76.'

Introduce the Rt ticketing to us from your perspective.

The RT was started out as an open-source free software package that Joe Kirby the DO IT Director of Business Systems was able to modify to replace the old RT system.

Have you ever encountered any complications while using the RT ticketing system? If so, how did you work around the problem?

Difficult to find rooms that consistently has issues. Created a subject naming scheme. Customers can see ticket information too. Different support employees can't have a chat/forum inside of a ticket. A lot has to do with boolean search engine. Building number and code. A lot of it had to do with a boolean search engine on the back end. Trying to look up a ticket and coming up with shorthand so they could quickly look for rooms that were causing problem. So, we came up with a shorthand for the building codes, and then putting in the room number without a space. So as long as everybody follows that naming picture they can find where the issue arises.

Can you walk me through a specific example of a particularly challenging issue that you resolved using the RT ticketing system?

RT was for the customer not for the staff. Customer does not know the details of the status and communication of what is happening with the steps taken to resolve the ticket. The ticketing system was never for the staff or for the university. This ticketing system is for the customer that's having the problem. The customer doesn't know the problem has been taken care of yet or that was taken care of when it was taken care of and what was done a lot of people don't know if it's their fault or something to do with the equipment or a combination of the two.

How do you prioritize and manage your workload when dealing with a large number of open tickets in the RT system?

What the boss asks to do first. Then do the problem that exists first. Then problems that exist that aren't really affecting anything. Lastly problems that exist that I can fix.

Have you ever encountered a situation where you were unsure how to proceed with a ticket or needed to seek guidance from a colleague or supervisor? How did you handle this situation?

Try to communicate, a lot of times we need to consider that the customer can see our tickets of trying to communicate with say, mechanical engineers and explain to them that they don't know how to press a button gets a little dicey. So, you've got to really use soft cat's paws to make sure that you're not insulting a professor by saying hey, you don't know what you're doing kind of thing. So, trying to work within the RT phrasing I guess, to be able to put it down in such a way that you're being polite and respectful at all times.

What steps do you typically take to ensure that tickets are resolved in a timely and efficient manner?

Look at it with the effect of class or if the problem is occurring on the same day. Look at it in terms of whether it's affecting immediate class right now.

In addition to the RT system, are there any other tools or software that you use to help manage your workload and resolve issues?

On the back end they use R25. See the details if they only have the room number to see who is using it or if it is open. Web admin (UMBC username lookup) we use R 25 Campus scheduling. The other thing I use is web admin.

Have you ever identified a pattern of recurring issues or problems? How did you address this and work to prevent future occurrences?

See the problem that comes with the same customer or same room. We'll see several tickets come in for the same room but not the same customer. So that's probably a problem with the room. When tickets coming in from the same customer but not in the same room that leads us to believe that the problem is more with the customer.

Can you tell me about a time when you had to work with a particularly difficult customer or client? How did you handle the situation?

Client several years ago who would not bring his own laptop around. He would not come to the office to pick up a laptop so he was asking for a laptop to be brought to his classroom every day. We reached a point where we couldn't wait. It's just physically impossible. The professor reached a point where he started just leaving the laptop and walking away and eventually, we had to stop.

How do you balance the need for thoroughness and attention to detail with the need to resolve issues quickly and efficiently?

You have to be able to do both. Clear and concise. Put in detail what happened with the ticket for possible future use. You've got to be able to do both. I need to be clear and concise. When I'm using RT, I need to make sure that I'm clear to the customer. And also, what I do is clear to anybody that's going to come behind me and look at this ticket and go you know what you were

doing in lecture. If I don't go into detail about what the problem was and what my steps were in troubleshooting and solving the problem. It's not going to help. So, I've got to be concise, but I need to be quick as well and thorough.

Have you ever identified an issue that required input or collaboration from other departments or teams within the organization? How did you go about securing the necessary assistance?

Sister department desktop support. They can’t administrate computer but use desktop support if they need someone there to fix the bult in pc. We do this quite often with our sister department head which is desktop support. desktop support, for instance, manages the open computers that we have in all the lecture halls and several large teaching spaces as well. We can administrate these computers all we can do is look at them and know, something's wrong. So, then we need to reach out to desktop support and we have opened a chat channel within WebEx messaging to communicate directly with desktop support and to say, hey, we need to borrow one of you people to go out because we're having a problem with a computer. That's out in a lecture hall. It's not necessarily an audio-visual problem. But we need somebody who has admin privileges to be able to fix the problem on the built in PC.

What are features you recommend RT tickets should have?

More designed for IT environment and not for the general campus. Dave mentioned how It's clunky and slow. Wish it was a little more a little faster and a little more streamlined. It's sort of a catch all for any problem whether it was you know, somebody trying to pay a bill with tuition or register a COVID test or a million other things and RTX is like this big catch all for everything. So, if it was more designed for the IT environment and it questions as opposed to say the campus as a whole like you know, a curb is broken or something like that, or there's a pothole in the street that becomes an RT ticket, as well as a software bug.

2.4 Artifact Analysis Report: Request Tracker (RT)

2.4.1 Introduction:

This report provides an analysis of Request Tracker (RT), a Help Desk ticketing system used by the University of Maryland, Baltimore County (UMBC). The analysis was conducted through document and artifact analysis, which involved examining relevant documents and artifacts to understand the use and effectiveness of RT. The purpose of this analysis is to provide insights into the adoption and implementation of RT at UMBC, and to identify any areas for improvement.

2.4.2 Background:

Before the implementation of RT, UMBC used a client-based product called Remedy, which required a vendor license for access. Due to the cost of Remedy, it was not feasible to use the program widespread across the university. In order to find a cost-effective and campus-wide collaborative solution, DoIT staff began to review other Help Desk options. After an extensive search, RT was discovered and fully launched in February 2009.

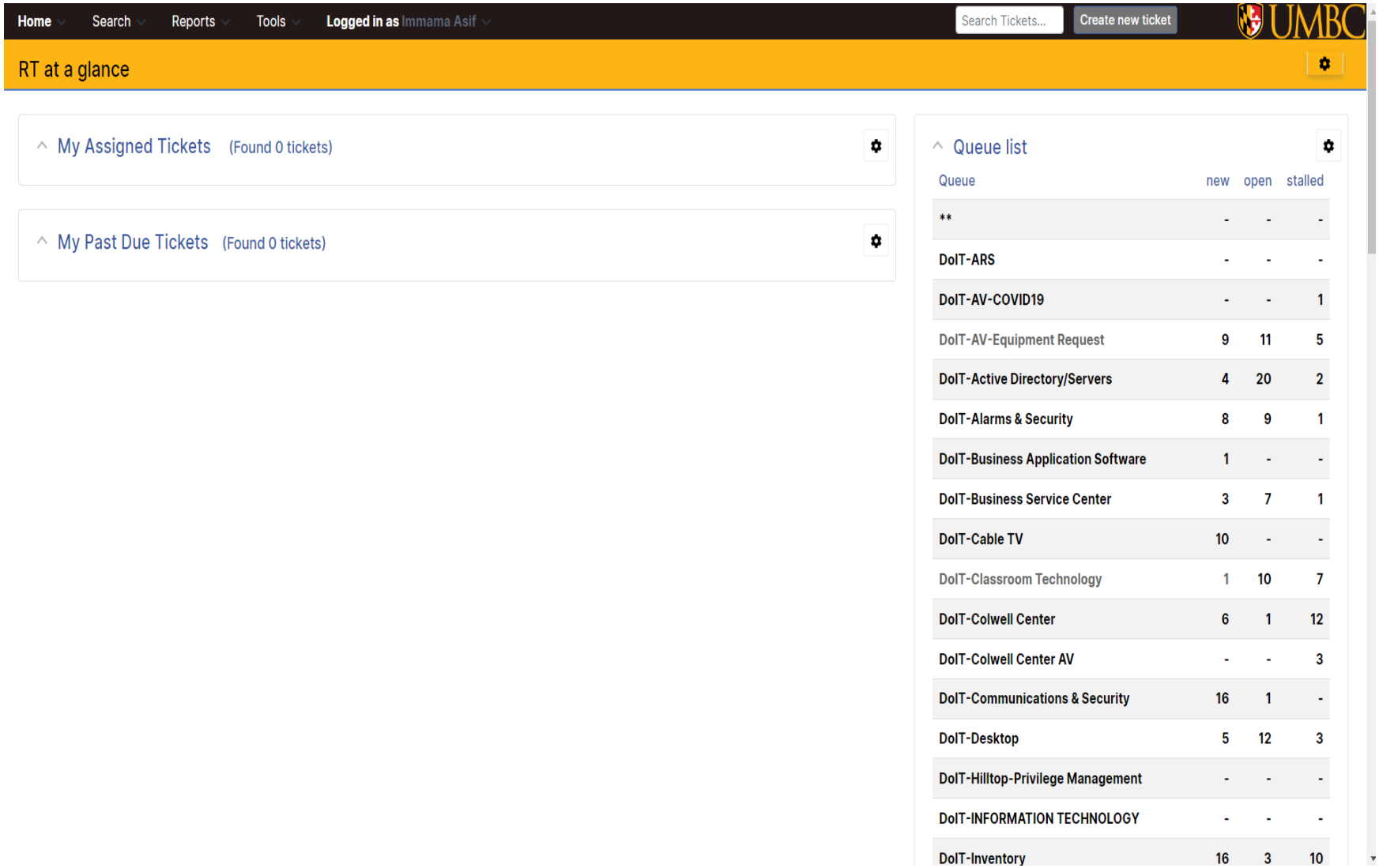
2.4.3 Analysis:

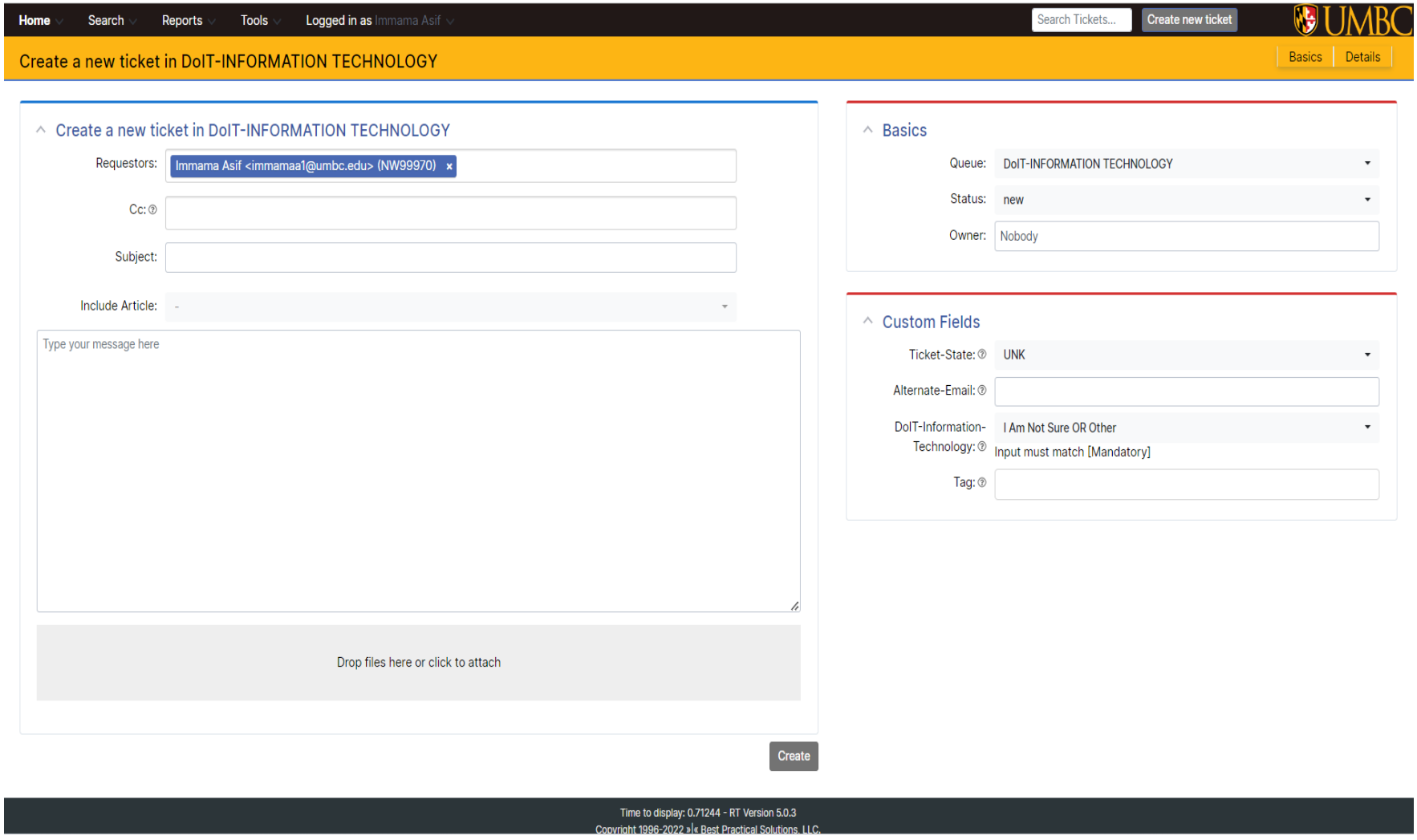
The analysis was conducted by reviewing relevant documents and artifacts related to the adoption and use of RT at UMBC. The following observations were made:

- RT is a free and open-source Help Desk ticketing system that allows for cost-effective and large-scale deployment across the university.
- RT is web-based, which means that users can access the program from any computer or web-enabled device.
- DoIT frequently holds meetings to discuss any problems or updates to the system.
- UMBC has not exercised the option for a higher level of support from Best Practical Solutions, as the configuration and management of RT has been absorbed into existing DoIT staff responsibilities.
- Since the initial roll-out of RT in DoIT, Finance, and HR, many other departments such as the Registrar's Office, Graduate and Undergraduate Admissions, Parking Services, and Residential Life have also incorporated the ticketing system into their support processes.
- Report Exchange (REX) has also been integrated for use with RT, allowing departments to examine data from submitted help tickets to observe trends and better serve the UMBC community.

2.4.4 Conclusion:

The analysis of RT at UMBC suggests that it is a cost-effective and scalable Help Desk ticketing system that has been widely adopted across various departments. The use of RT has allowed for collaboration and data analysis across departments, leading to better service for the UMBC community. However, it is recommended that UMBC continue to hold regular meetings to discuss any problems or updates to the system and consider exploring the option for a higher level of support from Best Practical Solutions if necessary.


Home
Search
Reports
Tools
Logged in as Immama Asif
Search Tickets...
Create new ticket
UMBC
RT at a glance
My Assigned Tickets (Found 0 tickets)
My Past Due Tickets (Found 0 tickets)
Queue list
Queue new open stalled
** - - -
DoIT-ARS - - -
DoIT-AV-COVID19 - - 1
DoIT-AV-Equipment Request 9 11 5
DoIT-Active Directory/Servers 4 20 2
DoIT-Alarms & Security 8 9 1
DoIT-Business Application Software 1 - -
DoIT-Business Service Center 3 7 1
DoIT-Cable TV 10 - -
DoIT-Classroom Technology 1 10 7
DoIT-Colwell Center 6 1 12
DoIT-Colwell Center AV - - 3
DoIT-Communications & Security 16 1 -
DoIT-Desktop 5 12 3
DoIT-Hilltop-Privilege Management - - -
DoIT-INFORMATION TECHNOLOGY - - -
DoIT-Inventory 16 3 10


Home
Search
Reports
Tools
Logged in as Immama Asif
Search Tickets...
Create new ticket
UMBC
Create a new ticket in DoIT-INFORMATION TECHNOLOGY
Basics
Details
Create a new ticket in DoIT-INFORMATION TECHNOLOGY
Requestors:
Immama Asif <immamaa1@umbc.edu> (NW99970)
Cc:
Subject:
Include Article:
Type your message here
Drop files here or click to attach
Create
Basics
Queue:
DoIT-INFORMATION TECHNOLOGY
Status:
new
Owner:
Nobody
Custom Fields
Ticket-State:
UNK
Alternate-Email:
DoIT-Information-Technology:
I Am Not Sure OR Other
Input must match [Mandatory]
Tag:
Time to display: 0.71244 - RT Version 5.0.3
Copyright 1996-2022 »|« Best Practical Solutions, LLC.

2.5 Use Cases

2.5.1 Use Case 1:

<table>
<tr><td colspan="2">Use Case Name: Create Trouble Ticket</td><td>ID: 1</td><td>Priority: High</td></tr>
<tr><td colspan="4">Actor: RT (Request Tracker) Ticketing System</td></tr>
<tr><td colspan="4">Description: This use case describes the process for creating a new trouble ticket in the RT ticketing system</td></tr>
<tr><td colspan="4">Trigger: Client has an issue that needs to be resolved</td></tr>
<tr><td colspan="4">Type: External</td></tr>
<tr><td colspan="4">Preconditions: User has access to the RT ticketing system
User has knowledge of the issue that needs to be resolved</td></tr>
<tr><td colspan="2">Normal Course:
1 Create Trouble Ticket
1. User logs into the RT ticketing system through their MyUmbc login.
2. User selects the option to create a new trouble ticket.
3. User enters a self-made title of their issue.
4. User provides tags that are built in the system to best describe the issue they are having.
5. User enters a room number under the format BUILDING_NUMBER.
6. User selects a system-built category for the issue.
7. User selects a priority level for the issue between the number of 10 and 1. One being low priority and ten being high.
8. User submits the new trouble ticket</td><td colspan="2">User_id (input)

Problem User(output)

Title(Input)

Problem Name (Output)

BUILDING_NUMBER(Input)
Problem Location (Output)

Priority level (Input)
Problem Priority Level (Output)</td></tr>
<tr><td colspan="4">Postconditions:

Trouble Ticket has been created and stored in the system.
Notification has been given to the user that their request is being handled.</td></tr>
</table>

Notification given to A/V that a new trouble ticket has been created.
Exceptions 1.) Tickets from the same person/group describing the same exact problem. a. System generates an error message to users trying to create another ticket that their ticket is already in the system and their problem is being handled. 2.) User tries to submit ticket with incomplete information (field for name, student Id, or even the problem itself are missing a. System generates an error message and highlights the missing fields of information. They cannot generate a ticket that has important information missing. 3.) User creates a ticket, but this problem is not within the purview of the department. a. System sends a message to the user that their problem is not something that this particular department can assist with. They may need to contact someone else.

Summary Input:	**Source:**	**Summary Output**	**Destination:**
User_id Title Building Number Priority Level	User User User User	Problem User Problem Name Problem Location Problem Priority Level	RT Ticketing System RT Ticketing System RT Ticketing System RT Ticketing System

2.5.2 Use Case 2:

Use Case Name: Evaluate Ticket Performance	**ID: 2**	**Priority: Medium**
Actor: User - Support Staff and Management		
Description: This particular use case outlines the steps involved in assessing the effectiveness of a previously created trouble ticket within the RT ticketing system. With this use case, the organization can identify areas of improvement in their support processes and enhance their user experience and customer satisfaction.		
Trigger: The user wants to evaluate progress on tickets.		
Type: External		

Preconditions: The user must have access to the RT ticketing system.	
Normal Course: 1. The user signs into the RT ticketing system. 2. The user selects the assigned trouble ticket. 3. The user reviews the ticket details and any related communications or notes. 4. The user updates the ticket status to reflect the current progress and actions taken. 5. The user adds any relevant comments or updates to the ticket. 6. The user closes the ticket if the issue has been resolved.	Interface log in (Input) User interaction with software (Input) Ticketing system displays status and updates on tickets (Output) User input (Input) Customer receives feedback on ticket (Output) Customer receives conclusion notes on ticket (Output)
Postconditions: The performance of the trouble ticket is evaluated.	
Exceptions: 5a. If the ticket requires further investigation or action, user assigns the ticket to another staff member. 6a. If the issue cannot be resolved, the user updates the ticket status to reflect this and adds relevant comments.	

Summary Input:	**Source:**	**Summary Output**	**Destination:**
1. Details of the assigned trouble ticket	Create trouble ticket	Details on trouble ticket	Staff member or Management
2. Current progress and actions taken	Request board	Notification of action Notification of comments	Staff/Management/Customer
3. Comments or updates to the ticket	Ticket ID	Staff member ID	Customer
4. Assigned staff member (optional)	Management		Management

2.5.3 Use Case 3:

<table>
<tr><td colspan="2">Use Case Name: Retrieve Ticket Records</td><td>ID: 3</td><td>Priority: Medium</td></tr>
<tr><td colspan="4">Actor: Staff and Management</td></tr>
<tr><td colspan="4">Description: This use case describes the user requesting access to previous ticket records. The system retrieves a list of past tickets which the user can scroll or search through to find the chosen ticket. The chosen ticket can then be opened up to display in depth information.</td></tr>
<tr><td colspan="4">Trigger: User wants details of past completed tickets</td></tr>
<tr><td colspan="4">Type: External</td></tr>
<tr><td colspan="4">Preconditions:
1. The user has knowledge on how to navigate the system.
2. The ticket records are up to date</td></tr>
<tr><td colspan="2">Normal Course:
1. The user signs into the RT ticketing system.
2. The user requests to look at ticket records.
3. The system shows a list of completed tickets.
a. The list is ordered by completion date, new to old.
4. The user then scrolls through a list of tickets to choose from.
a. The user can use a search function to look for the chosen ticket.
b. Tickets can be searched based on the ticket name, ticket id, and ticket tags.
5. The user then opens up the chosen ticket once located.
6. The system then gives in depth details about the chosen ticket.</td><td colspan="2">User logs in (Input)

Request is sent to system (Input)

System displays log of tickets (Output)

User navigates the records (Input)

User selects ticket (Input)

System displays ticket information (Output)</td></tr>
<tr><td colspan="4">Postconditions:
1. User gains access to information from selected ticket
2. System logs request to access records</td></tr>
<tr><td colspan="4">Exceptions:
E1: The user is not logged in to authenticated account (Occurs during step 2)
1. The system displays an error message “This account is not allowed access to ticket</td></tr>
</table>

records”

E2: The queried ticket cannot be found (Occurs during step 4a)

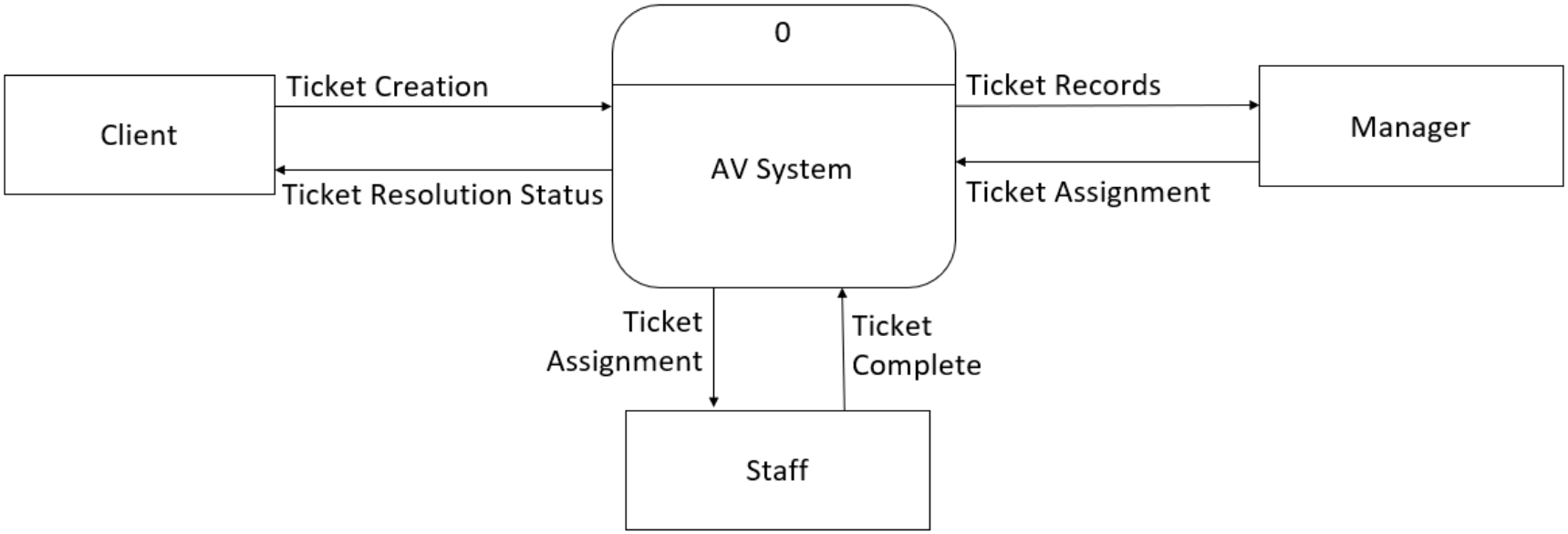
1. The system displays a message “No tickets could be found that match the search input”
2. The user can then input another search or exit the system

Summery Input:	**Source:**	**Summary Output**	**Destination:**
User logs in	User	System displays log of tickets	RT Ticketing System
Request is sent to system	User	System displays ticket information	RT Ticketing System
User navigates the records	User		
User selects ticket	User		

3.0 Deliverable 3

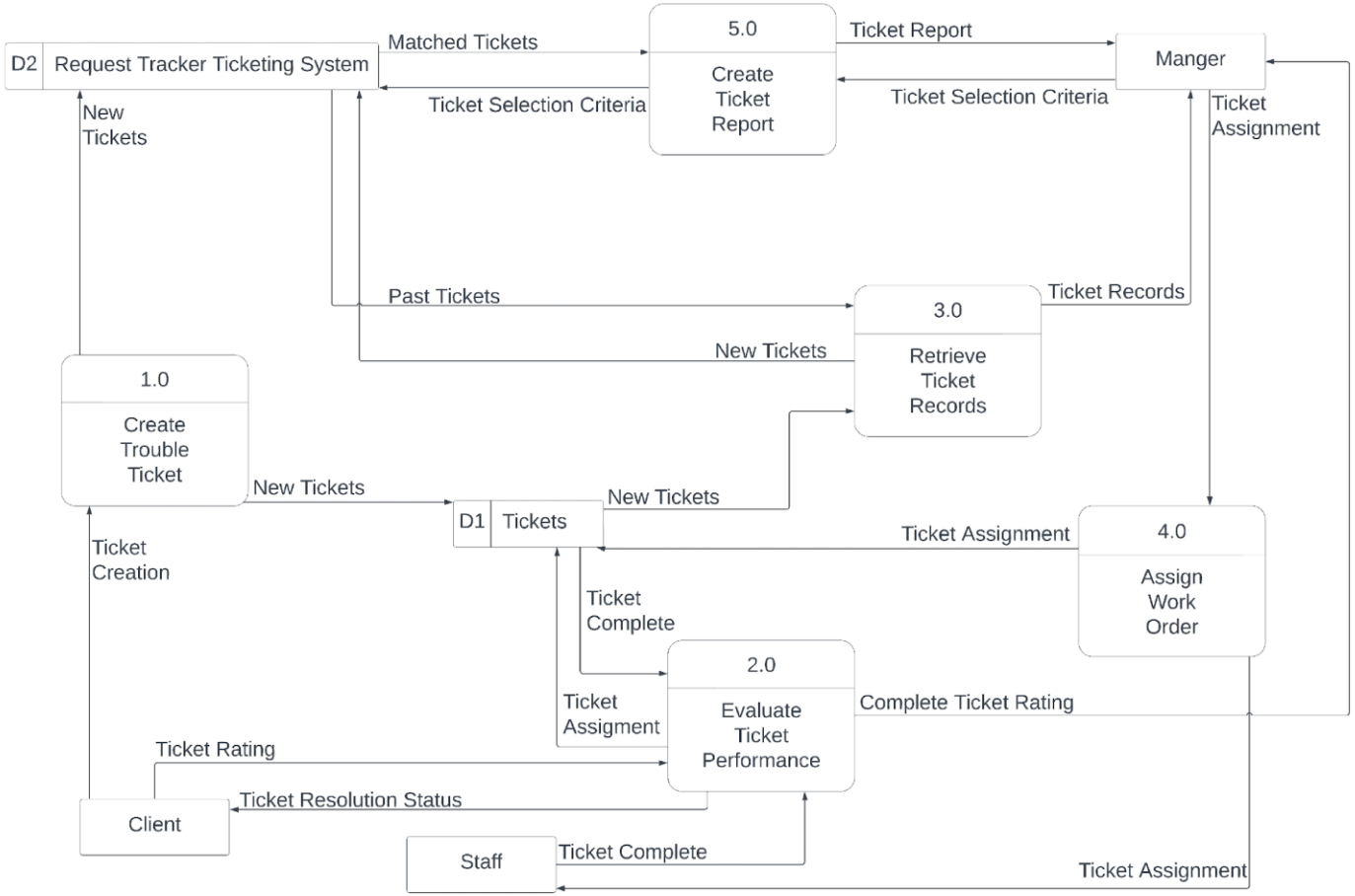
3.1 Context Diagram:

The following diagram is the context diagram which is the top-level data flow diagram. The numbered 0 process is the AV System. The context diagram is used to show the overall systems process. Along with the single process, it is accompanied with several external entities which contribute to the system, these entities are the Manager, Staff, and Client.



3.2 Level 0 Diagram:

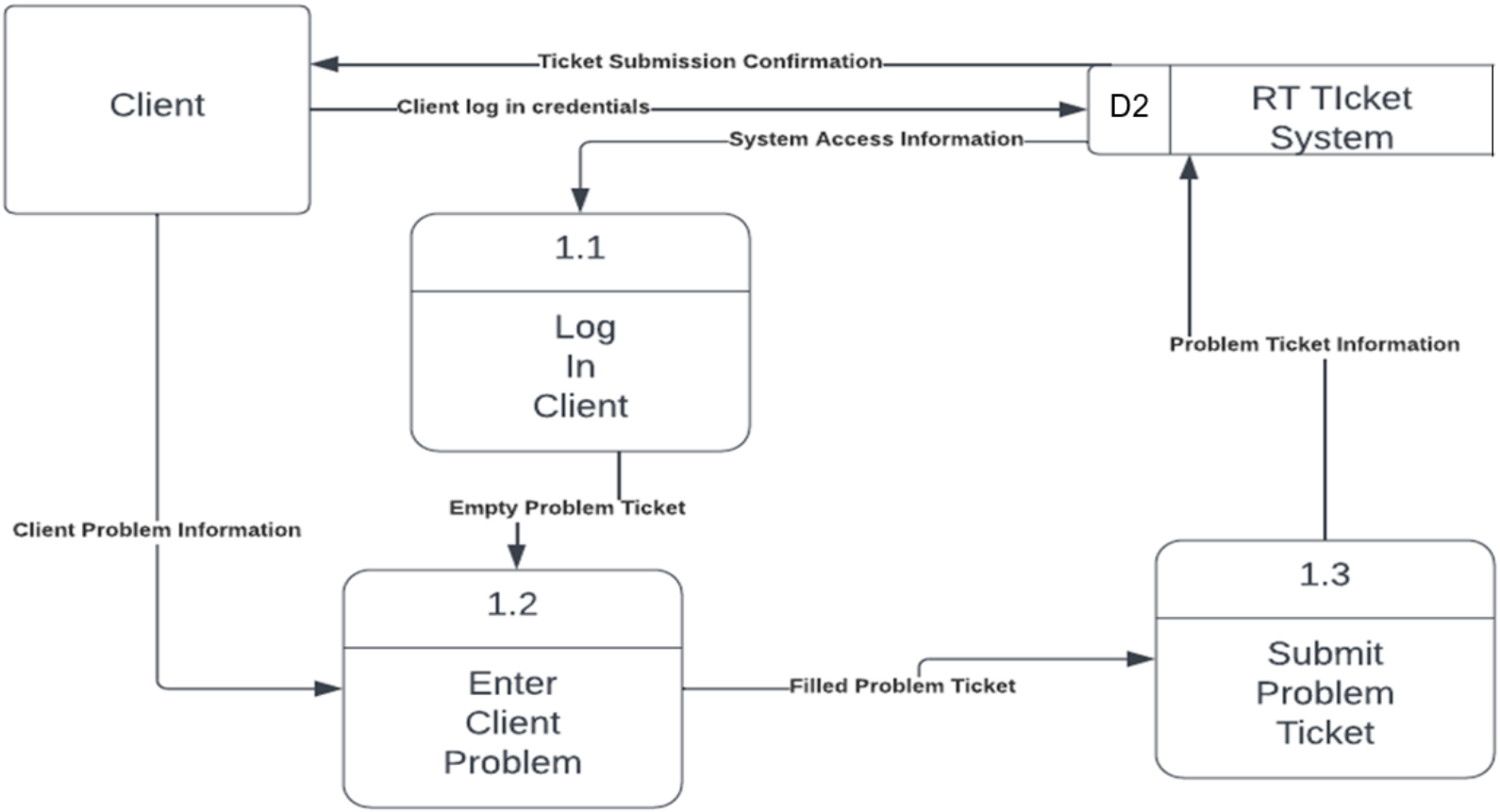
This section contains the level 0 diagram. It is an “exploded” version of the previous context diagram. The 0 process is broken into 5 different sub-processes which are numbered sequentially. Additionally, more data flows and data stores are added. The 5 processes for this level 0 diagram are Create Trouble Ticket, Evaluate Ticket Performance, Retrieve Ticket Records, Assign Work Order, and Create Ticket Report.



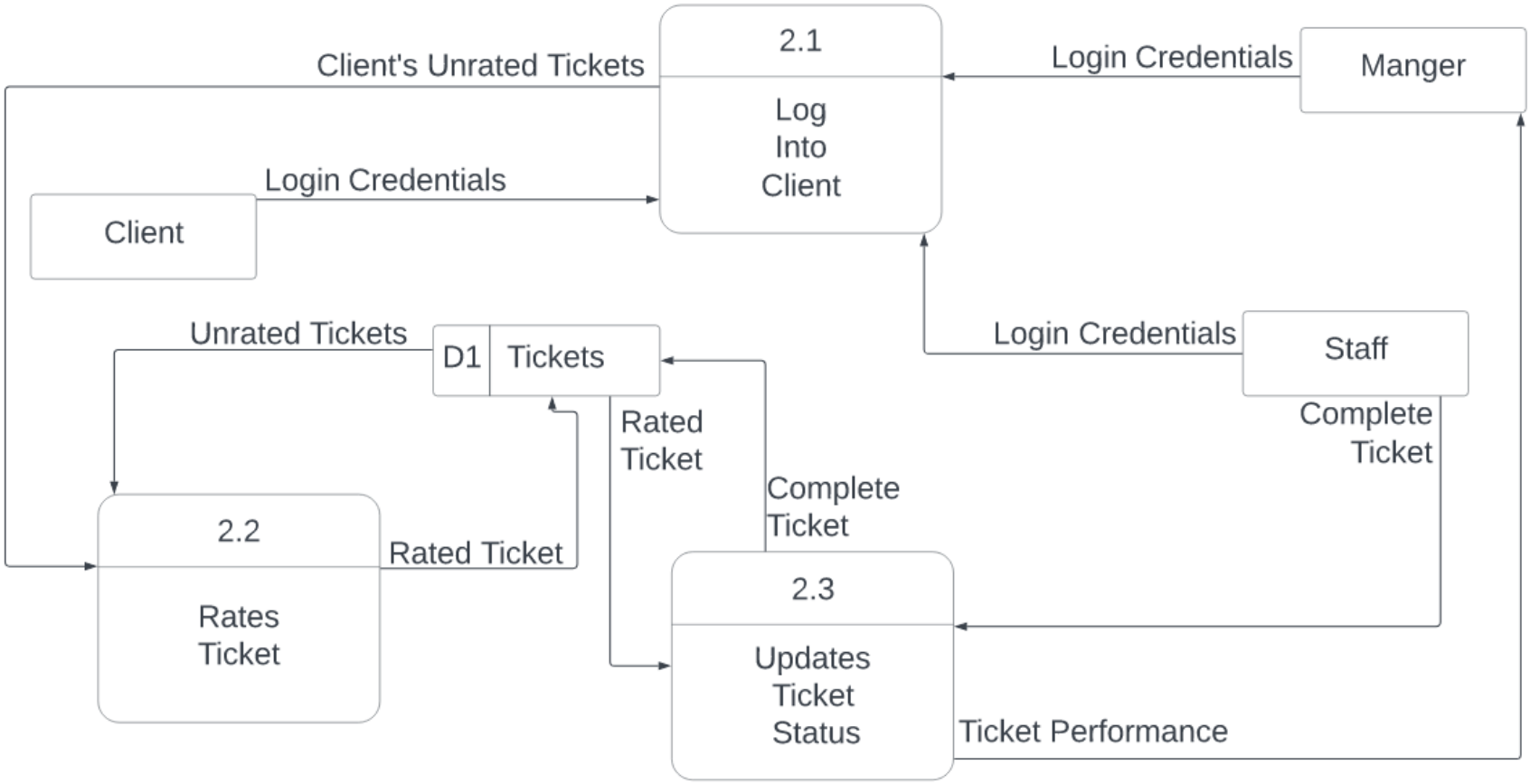
3.3 Level 1 Diagrams:

Next, we have 5 different level 1 diagrams. These are child diagrams of the 5 sub-processes numbered 1.0 to 5.0 sequentially which were again named Create Trouble Ticket, Evaluate Ticket Performance, Retrieve Ticket Records, Assign Work Order, and Organize Past Ticket in the level 0 diagram which are again “exploded” to greater demonstrate how function is performed in these processes.

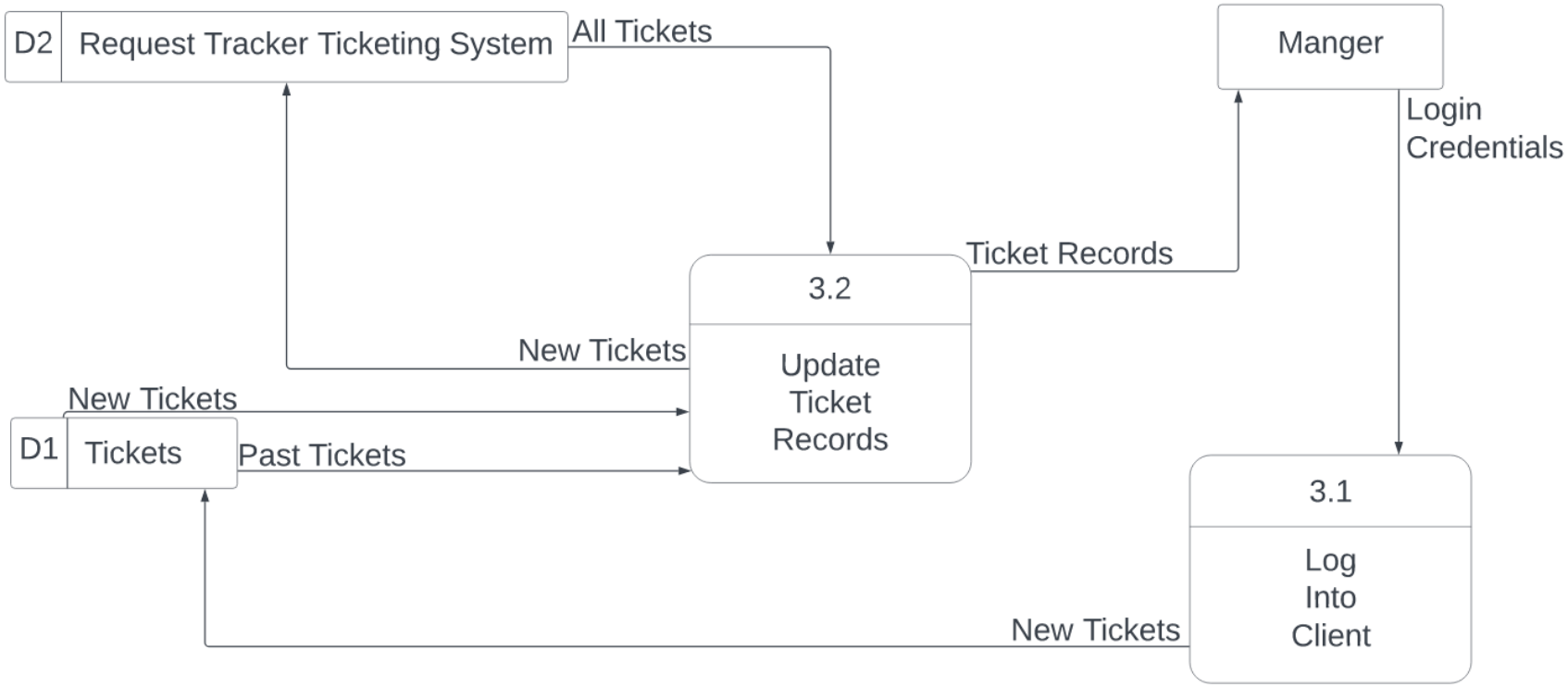
3.3.1 - Level 1 Diagram: Process 1.0:



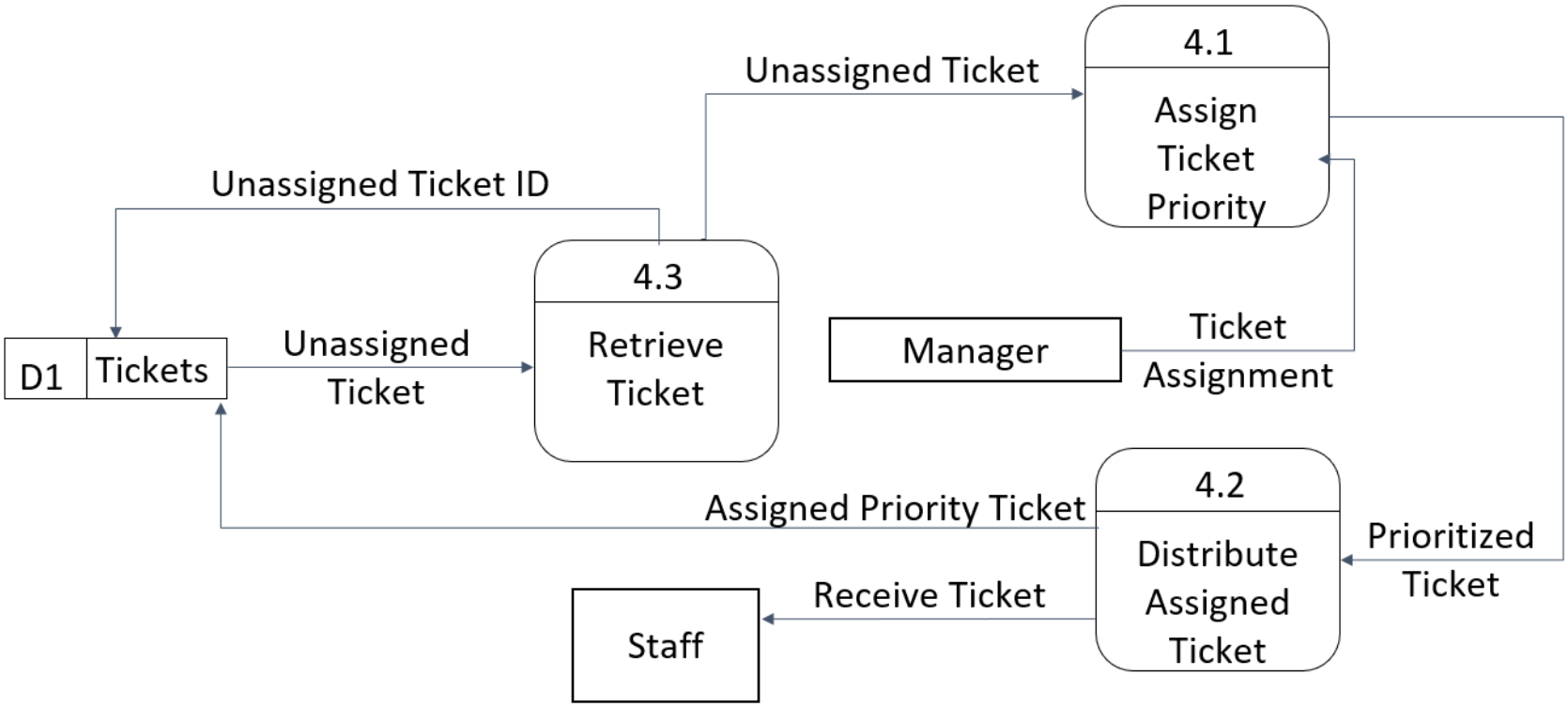
3.3.2 - Level 1 Diagram: Process 2.0:



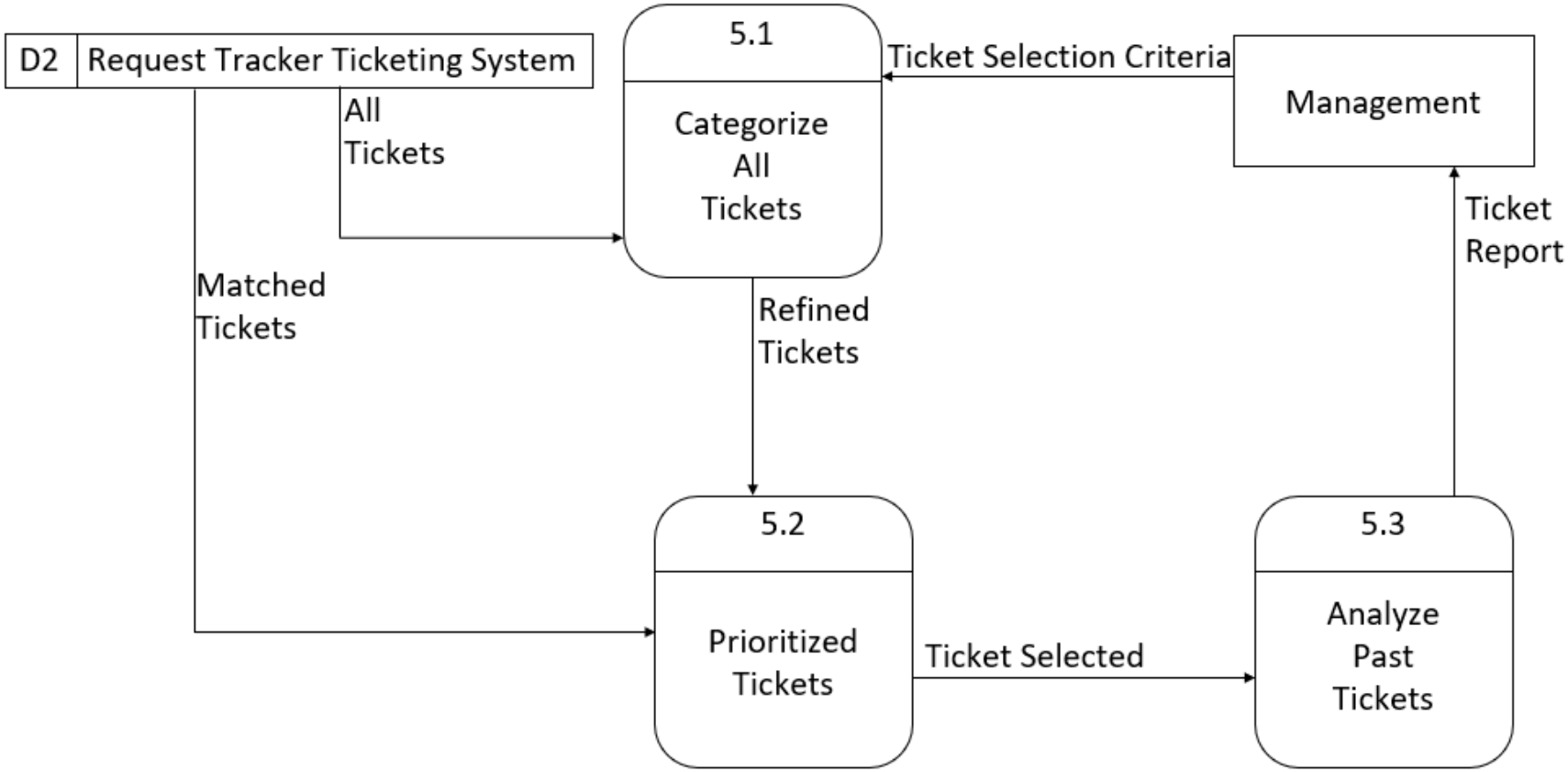
3.3.3 - Level 1 Diagram: Process 3.0:



3.3.4 - Level 1 Diagram: Process 4.0:



3.3.5 Level 1 Diagram: Process 5.0:



3.4 Data Dictionary Definitions:

Finally, we have our data dictionary. Data dictionaries are used to ease the transition from analysis phase to the design phase. In this data dictionary we have 10 different data flows, data stores, and entities that are in the level 0 diagram are shown in the table with corresponding descriptions, instantiated data, examples, and data type constraints.

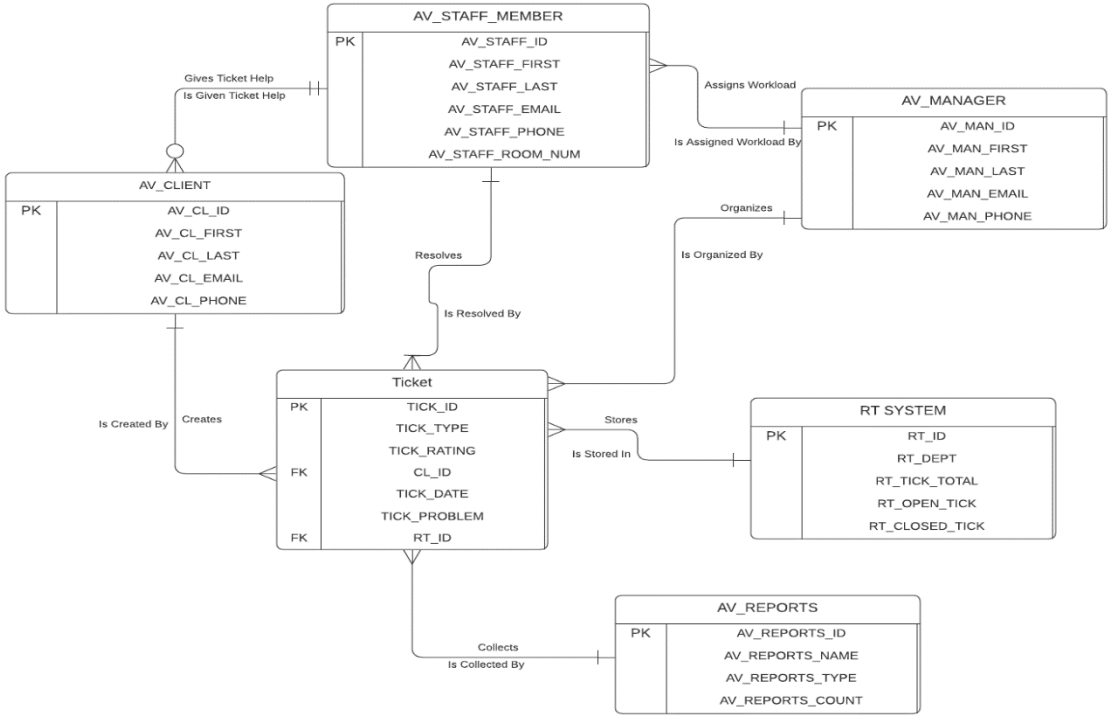
NAME	TYPE	DESCRIPTION	CONTENTS	DATA TYPE
TICKET RATING	Data Flow	The client provides a rating on how well the ticket was solved from a Likert scale 1-5.	Contains a 1-5 rating. Example: 4	INT
CLIENT	Entity	Name of the client that is creating the trouble ticket.	Contains name of a client. Example: John Doe	VARCHAR
TICKET RECORDS	Data Flow	Details of completed tickets that were completed in the past.	Includes details of past tickets. Example: When the ticket was completed, what the issue was, who submitted the ticket	VARCHAR
TICKET SELECTION CRITERIA	Data Flow	Parameters that are used to filter and organize past tickets.	Parameters to search through tickets using keywords. Example: “Speaker” would show tickets that include the keyword.	VARCHAR
TICKET CREATION	Data Flow	Checks to see if the client successfully creates ticket.	Yes or No to see if the client created the ticket.	YES/NO

			Example: Yes	
STAFF	Entity	Name of staff member using the system.	Contains name of a staff. Example: John Doe	VARCHAR
REQUEST TRACKER TICKETING SYSTEM	Data Store	All created ticket documentation.	Stores all data about tickets that have been created. Example: “Ticket Name, Creation Date, Priority”	VARCHAR
TICKETS	Data Store	Tickets that have been assigned and not assigned to a staff member to be solved.	Stores the name of tickets that have been both assigned and not assigned. Example: “Assigned Ticket Name”	VARCHAR
MATCHED TICKETS	Data Flow	Tickets that are filtered through set parameters to meet the requirements needed.	These are ticket names that correspond to the search used. Example: “Speaker Issue Ticket Name”	VARCHAR

4.0 Deliverable 4

4.1 Entity Relationship Diagram

This is the Entity-Relationship diagram for the system. There are 5 entities: Client, Staff Member, Ticket, Manager and RT System.



Assumptions:

- There are no NULL values
- Tickets are Created by Clients one after another (not at the same time)
- Tickets are Resolved by Staff Members one after another (not at the same time)

4.2 Alternative Matrix 1

This is the first Weighted-Alternative matrix for the data modeling document. Several criteria were created and each one was assigned a weight and a score. This alternative matrix weighs different options for the system against technical, economic, and organizational criteria.

Evaluation Criteria	**Relative Importance**	**Cloud Ticketing System**	**Score (1-5)**	**Weighted Score**	**Server Ticketing System**	**Score (1-5)**	**Weighted Score**	**Website Ticketing System**	**Score (1-5)**	**Weighted Score**
Technical Issues:										
Tech Support	15	With a cloud ticketing system, software support will be handled by off-site contractors.	4	60	With a server ticketing system, on-site staff is responsible for technical issues.	2	30	With a website ticketing system, there may be more downtime and less support for users outside of normal business hours.	3	45
Hardware	10	With all the necessary resources being located on the cloud there is no hardware needed	5	50	Hardware must be purchased, leased by internal sources, making it very expensive.	1	10	Hardware will be off-site. No additional hardware will be required onsite.	5	50
Software	15	The only software you really need is to access the cloud.	4	60	Software must be purchased or rented from a vendor, making it a very expensive option.	2	30	Software will be installed on off-site network. Additional software is required to access and modify the website.	3	45

Network Requirements	10	If the network is down the whole system would be down unless a backup was implemented.	2	20	Bandwidth and number of cables must be increased to support the server.	2	20	No additional network components or bandwidth is required.	5	50
Economic Issues:										
Cost	20	There are many cloud options where you can do a yearly payment or monthly payment. However, since its external keeping track of the cloud, the price could change unless contracted.	2	40	Leasing or purchasing server equipment is expensive. Having on-site hardware may also require upgrading circuits to accommodate for the increase in power consumption.	5	100	Many website hosting companies offer a monthly/yearly payment plan. Occasionally there are some upfront fees.	1	20
Integration	5	As long as the system has the correct software to run the cloud services and network capability, it integrates easily.	5	25	Integration will the responsibility of paid on-site staff. This is a lengthy process.	2	10	Integrating a website will be simple but may take time, as changes may require approval from the vendor.	3	15
Customization	10	Many cloud-based ticketing systems provide many different options depending on what the company needs and you are able to change those options at any time you want (At the right price)	5	50	A server-based ticketing system allows for full customization of services and functions.	5	50	Although many websites offer tools and services for making the website fit the needs of the users/operators, there is typically limits.	1	10
Org. Issues:										
Learnability	5	The system would work very similarly to the system they already have in place so it would not be difficult for the employees to learn.	4	20	The system would require extensive training for operators and some training for users.	2	10	The system will be built on an existing platform, making it easy to use and train operators.	4	20

Maintainability	5	The company itself does not have to worry about maintaining it, it would all be on the third-party cloud service provider	5	25	Maintaining a server-based ticketing system will be easy since all hardware is on-site, however it may be more expensive.	3	15	Since the website is held offsite, there is no need for maintenance for staff. There is typically scheduled downtime for updates which may impact users and operators.	3	15
Availability	5	If the company's network connection is stable there would be nothing to worry about but as that is not guaranteed and unless there is a backup there would be no service during that down time.	3	15	Components for servers are readily available. Additionally, most components are universal, meaning that components may be borrowed from other machines.	4	20	There may be scheduled downtime outside of business hours. Depending on the product purchased, support may not be available at all times in the event of an outage.	2	10
TOTAL	100		39	365		28	295		30	280

The chosen alternative is the cloud ticketing system. After careful analysis, we find that the optimal alternative is the cloud ticketing system. This is based on many factors including Availability, Maintenance, Cost etc. Although a server-based ticketing system and website-based ticketing system are great options, they are not efficient enough for the AV department's needs.

4.3 Alternative Matrix 2

This is the second alternative matrix. Nonfunctional requirements were put into different categories (Operational Requirements, Performance Requirements, Security and Cultural and Political) and matched up with different architectures. A client-side server would be chosen.

Requirements	**Server-Based**	**Client-Based**	**Thin-Client Server**	**Thick-Client Server**
Operational Requirements:				
1.1 The system should be easy to use and navigate.		✓	✓	
1.2 The system should have high availability and reliability to ensure tickets are processed and resolved in a timely manner.	✓			✓
1.3 The system should be scalable and able to handle a large volume of tickets.	✓	✓		✓
Performance Requirements:				
2.1 The system should have fast response times to ensure efficient ticket resolution.	✓	✓	✓	
Security:				

3.1 The system should have secure login and access controls to prevent unauthorized access to sensitive information.	✓	✓	✓	
Cultural/Political:				
4.1 The system should be sensitive to the culture and politics of the organization, such as adhering to relevant policies and regulations.		✓	✓	

For the 1.1 operation requirement, client-side based was chosen over server-side based as the system being easy to use and navigate depends mostly on how the web page is oriented. Client-side based is faster as it does processes straight on the client rather than on the server making it more convenient for navigation as it would not require an intense process. Thin client was also chosen for 1.1 as the server does most of the computations so the client has less load and is smoother to operate. For the 1.2 operation requirement both server based and thick client was chosen. This is because thick clients are not dependent on the server and can operate without a network connection making them more reliable than thin clients. Server-side based was also chosen so operations are done before the application reaches the client, this is to again ensure availability and reliability. For the 1.3 operation requirement both server and client side are applicable but thick client was chosen over thin client. This is because a thick client is much more flexible than a thin client as it uses its own hardware rather than the servers.

4.4 Hardware and Software Specifications

This is a Hardware and Software Specification. This is a requirement for the kind of hardware and software that will be used in the system. This document details specific hardware (processors and memory) and software (Operating Systems and applications) for the system.

	Standard Client	Standard Web Server	Standard Application Server	Standard Database Server
Operating Systems	Windows 10 or up	Apache HTTP Server	Windows Server	Oracle
Special Software	Microsoft Word	Apache	Apache Tomcat	Oracle Server
Hardware	Intel Xeon 8468V Processor ITB Internal SATA Hard drive for Desktops	Intel Xeon D-2745NX Processor Dell T6TWN 1.2TB 10k Sas 6gbs 2.5in internal brand-new hard drive	Intel Xeon Processor E3-1240 Dell T6TWN 1.2TB 10k Sas 6gbs 2.5in internal brand-new hard drive	Intel Xeon E-2378 Processor PS6000X Dell EqualLogic 4.8TB – 9.6TB Storage Array
Network	Xfinity WIFI	Monoprice Cat6 Ethernet Bulk Cable - Solid, 550MHz, UTP, CMR, Riser Rated, Pure Bare Copper Wire, 23AWG, 1000ft, Purple, (UL)	Monoprice Cat6 Ethernet Bulk Cable - Solid, 550MHz, UTP, CMR, Riser Rated, Pure Bare Copper Wire, 23AWG, 1000ft, Purple, (UL)	Monoprice Cat6 Ethernet Bulk Cable - Solid, 550MHz, UTP, CMR, Riser Rated, Pure Bare Copper Wire, 23AWG, 1000ft, Purple, (UL)

These servers are used to help create the Cloud System environment picked in the Alternative Matrix 1

5.0 Deliverable 5

5.1 User Interface Design:

This first section consists of our User Interface Design. This section will showcase the design principles that were taken into consideration to ensure consistency amongst the design of the system as a whole.

The consistency of a user interface is an important part of a system. Consistent systems allow users to predict what to expect in a system which can improve user efficiency with the system.

To achieve a consistent user interface, it is important to follow established interface standards and design principles. Minimizing user effort is an example of one such principle, excessive navigation can result in wasted time. One way to gauge user effort is through the “three clicks rule,” which suggests the distance between the main menu and a user interface element should not exceed three clicks.

Understanding the users of a system is also essential for consistency in a user interface. This ensures that the system provides a uniform experience for all users. The use of use scenarios, which outline the steps a user performs in a system, is a useful technique for understanding the users of a system. For example, a shopping website may create use scenarios for two different types of users, a focused user that is shopping for a specific product as well as an exploratory user that is just browsing the website for pleasure. The focused user would look up models of a specific product, browse through the available options, filter through their options by price, brand, and reviews, select a product to purchase, and may contact support with questions. Whereas the exploratory user would browse through the front page of the website browsing through suggestions and recommendations for various products. They would then select a few products that catch their attention and potentially buy a few of them and may also contact customer support with questions. With both use cases in mind, a user interface designer can see that both use cases have similarities, such as selecting a product, purchasing it, and contacting customer support with questions. The designer can create a more consistent user interface by ensuring the buttons that perform these functions are located in the same location for both use cases. In summary, understanding the users of a system creates a consistency that allows returning users that change use cases to have some familiarity with the system and prevent possible confusion.

The principle of organizing the interface should be considered when designing a user interface. By adhering to the principle, an easier, more satisfactory experience for users can be created. To plan the organization of a system’s interface, an interface structure diagram is created. One way an interface structure diagram can be drawn is by drawing each interface element as a box, giving it a unique number and name. Each element is then connected with lines to navigate from one menu to another. By visualizing the connections between each element, inconsistencies between interfaces can be more easily identified. In the case of website development, It may be useful to draw a sitemap to organize the content and identify inconsistencies. Sitemaps have no standard format; some are drawn as a hierarchy whereas others are drawn as a series of nodes.

Defining interface standards helps increase consistency across a system by providing a template for designers and developers to follow. The template ensures that the interfaces are consistent across the system. The first of the four standards are the interface metaphor, using a concept from the real world to

model an information system. The metaphor allows the user to predict what features the interface might provide, even without use of the system. For example, the use of a metaphor is the file and folder analogy for the data management system on a computer. The user would expect functionality consistent with a real-life file and folder with the system, which is the functionality provided. The second standard to define is for interface objects and actions. The terminology to define them should be consistent with the expectations of the customer and should be straightforward and easy to understand. For example, defining the action of purchasing an item to be referred to as purchasing instead of buying. Interface icons should also be defined for the user interface. For common functions that exist elsewhere, icons developed by others is a simple but effective approach. For example, an icon of a trashcan is easily understood to indicate delete as this meaning is consistent with other software. The last standard to define is the interface template, the general appearance of all interface components. This includes components such as the basic layout of the interface, color scheme, font style and font size. The template combines the other three standards: metaphors, objects, actions, and icons into a single template to ensure user interface consistency throughout the system.

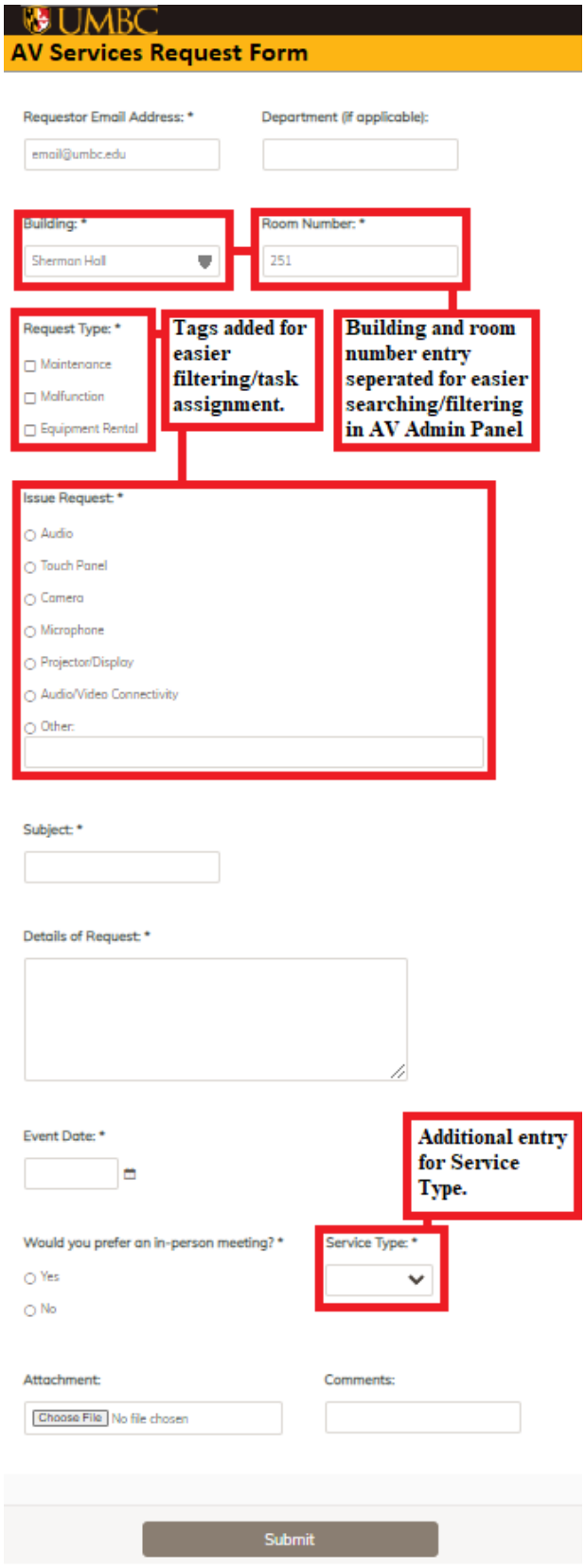
One of the most important design principles to achieve consistency in a user interface is the principle of having consistent design grammar. In English, there are common conventions for proper grammar. For example, “I love IS436.” follows proper grammar and has meaning, whereas “Love I IS436” does not. Similarly, there are common conventions established in user interface design. Rectangles that change color when a user mouses over them are typically buttons, and rectangles that dynamically change to the completion of a task are progress bars. By following the conventions users are already familiar with, a more consistent experience can be achieved. Adhering to a consistent grammar order is important as well. The user interface can have object-action order or action-object order. In object-action order, the user specifies the object then the action to be performed on the object whereas in action-object order, the user specifies the action to be performed, then the action. Staying consistent with one of these choices will help keep the user interface consistent.

5.2 Design Prototypes:

This section contains our design prototypes, of which there are five in total. The first is the AV Services Request Form, followed by the AV Services Admin Panel. The form is what the user would use when creating a ticket, and the admin panel is the interface the admins would see when searching through tickets. The next design prototype is the Loaner Equipment Request, this is the form used when a user would like to loan out equipment such as a laptop from the AV department. The fourth is RT at a Glance, this is the design interface, an admin would see when they log on to the system. The last and fifth design interface is the Detailed Ticket View, which shows the admin a detailed list of information from an individual ticket along with customer information and a communication stream.

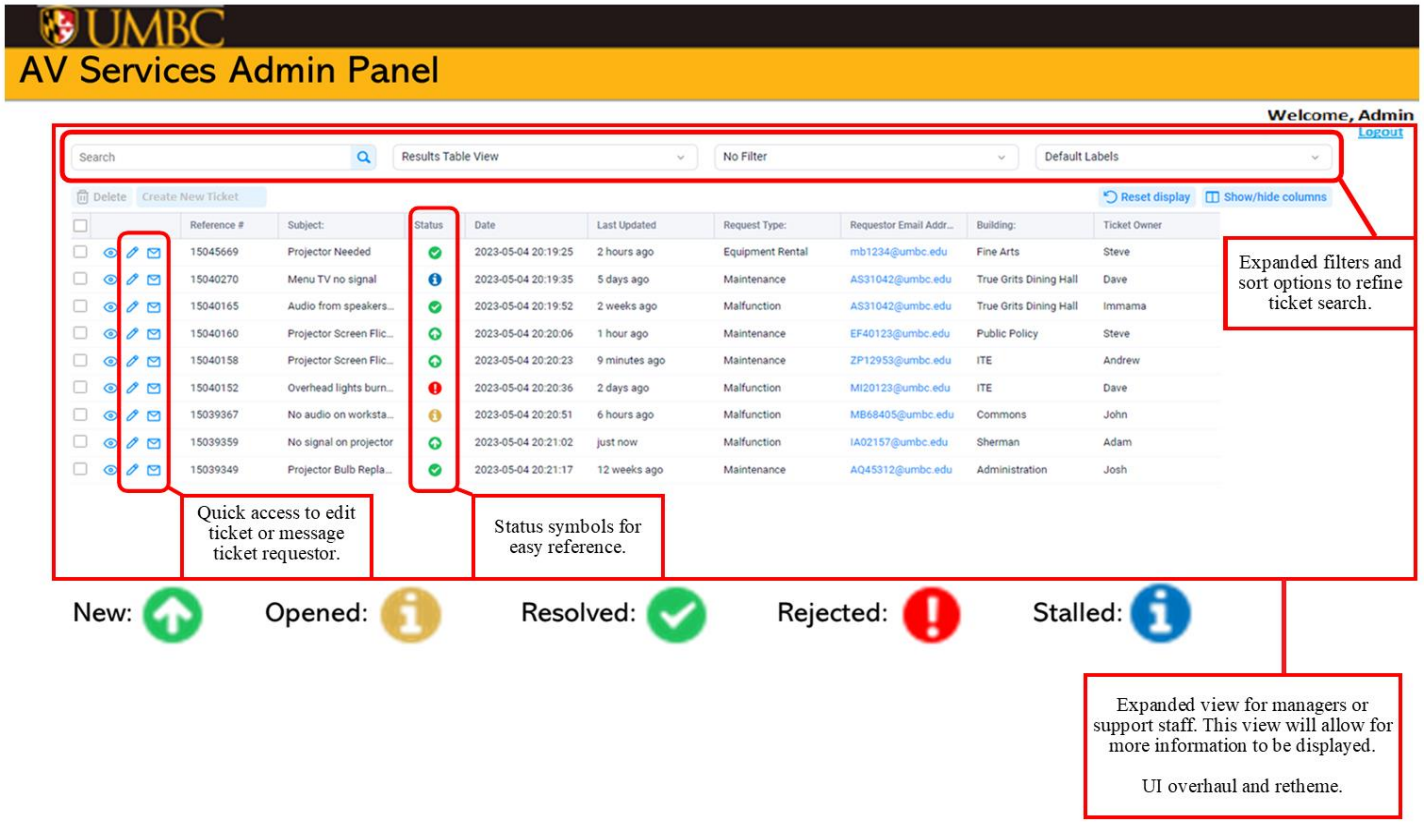
5.2.1 AV Services Request Form:

The current AV Services Request Form had many downfalls. This includes non-standardized naming conventions for buildings and classroom numbers, making it difficult to search tickets by building names and room numbers. Additionally, the current form is not very interactive and is more professor oriented than being used for requests from the general UMBC population. Overall, our team wanted to ensure that we met the needs of both the user and the support staff.


UMBC
AV Services Request Form
Requestor Email Address: *
email@umbc.edu
Department (if applicable):
Building: *
Sherman Hall
Room Number: *
251
Request Type: *
Maintenance
Malfunction
Equipment Rental
Tags added for easier filtering/task assignment.
Building and room number entry seperated for easier searching/filtering in AV Admin Panel
Issue Request: *
Audio
Touch Panel
Camera
Microphone
Projector/Display
Audio/Video Connectivity
Other:
Subject: *
Details of Request: *
Event Date: *
Additional entry for Service Type.
Would you prefer an in-person meeting? *
Yes
No
Service Type: *
Attachment:
Choose File No file chosen
Comments:
Submit

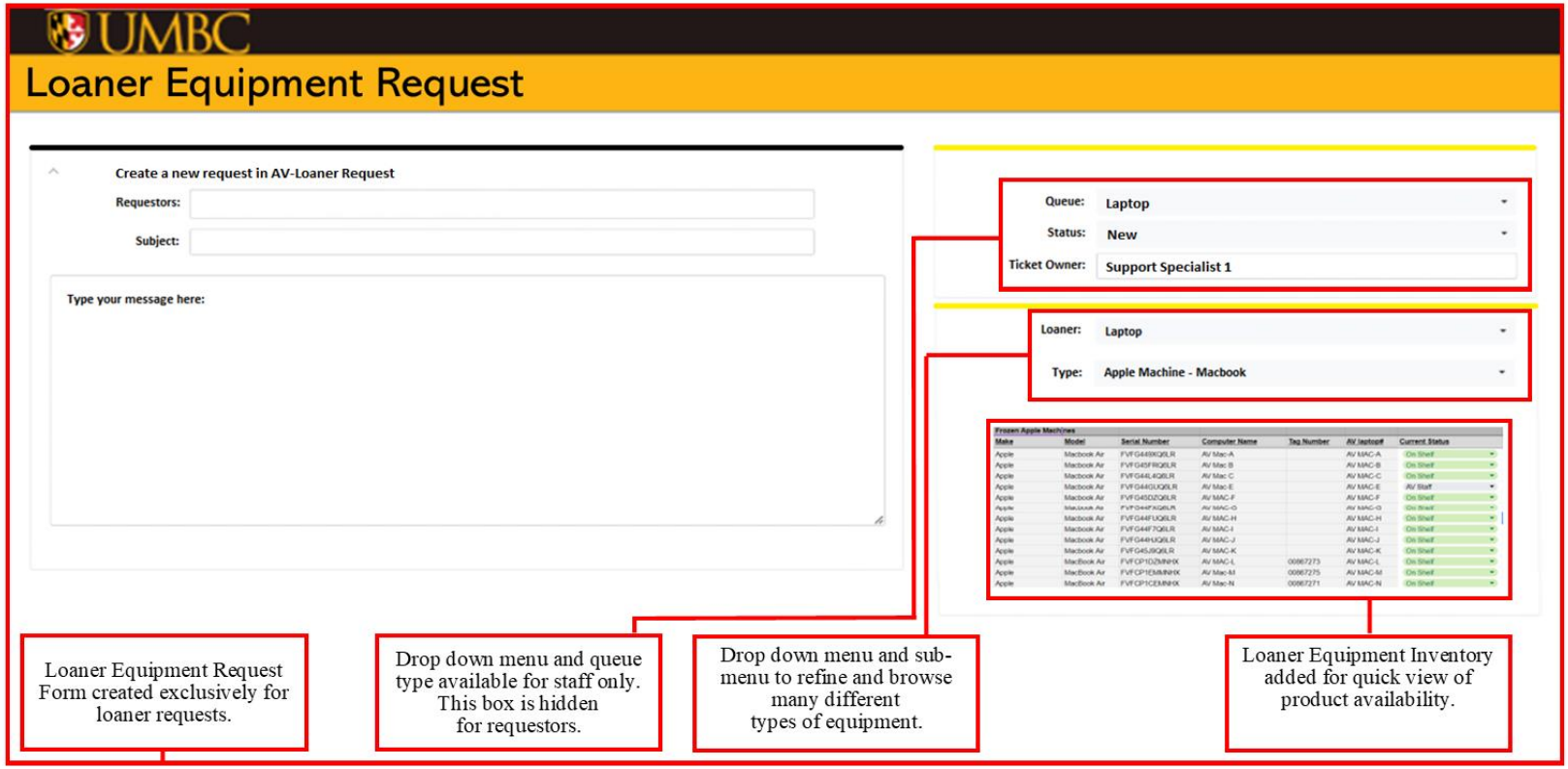
5.2.2 AV Services Admin Panel

The current AV Services Admin Panel was very daunting to look at. There was an overload of information in every pixel of the screen. The main goal of redesigning the admin panel was to make sure that managers and staff can know just enough information to do their job without having to read every detail of the page. This does not mean that information will be deleted. It will instead be relocated to another place (AV Services Ticket Details) if a specific piece of information was needed. Additionally, there were very few options to sort and filter information. To make sorting and filtering quick and effortless, we added three different quick filter dropdown bars.



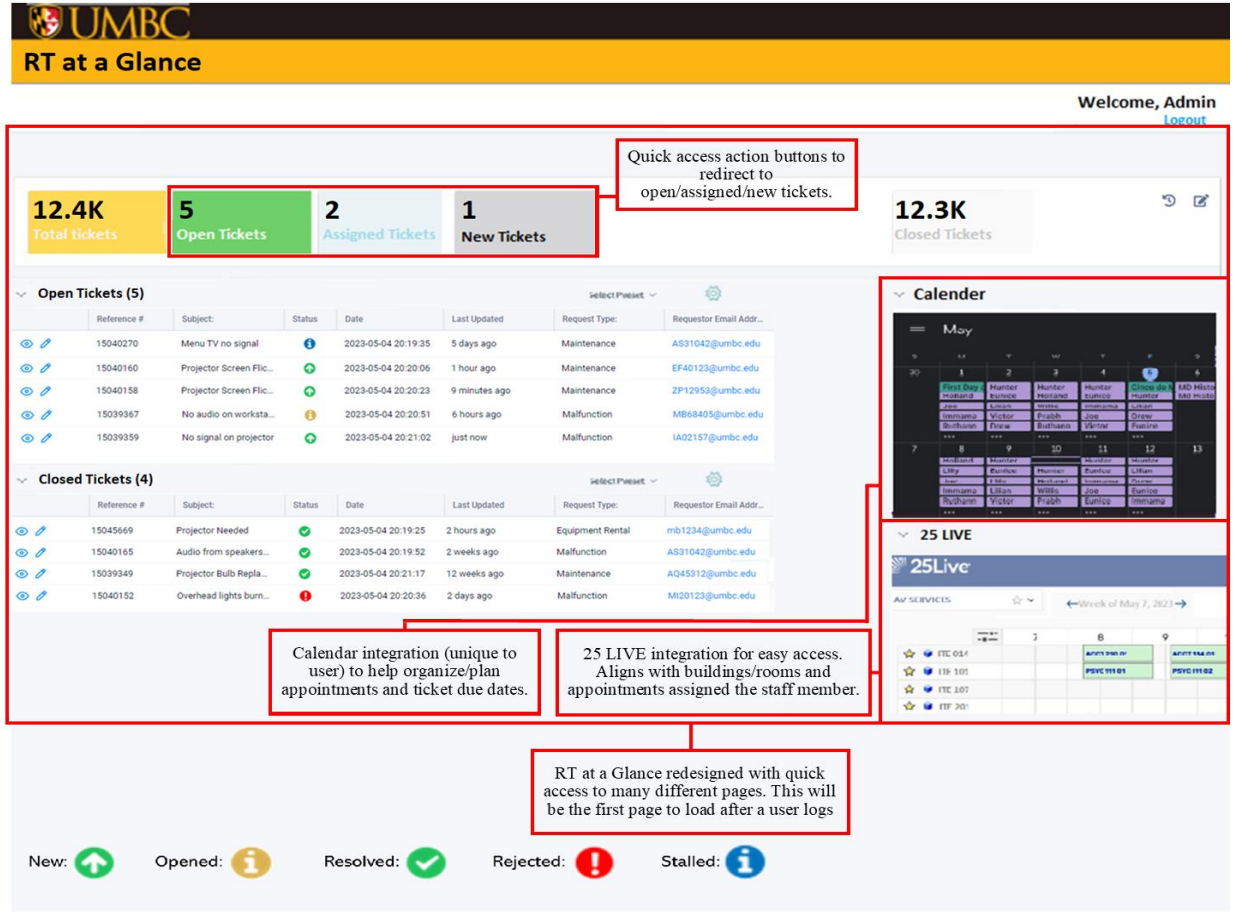
5.2.3 AV Services Loaner Equipment Request:

Audio/Visual repairs and maintenance are only half of what the AV services team offers. The other crucial component of AV services is their Loaner Equipment program, which allows students and professors to rent out equipment (microphones, computers, etc.) in order to complete tasks and host events on campus. This warranted the creation of a separate request form specifically for loaner equipment. Our team's goal was to make it simple for both the user and support staff. This includes drop-down menus which show the many different rental options AV Services offers. Additionally, a live inventory view is given to the requestor to show the different devices that are on the shelf currently. The integration of the inventory system is important to help fulfill the needs many different needs of staff and students.



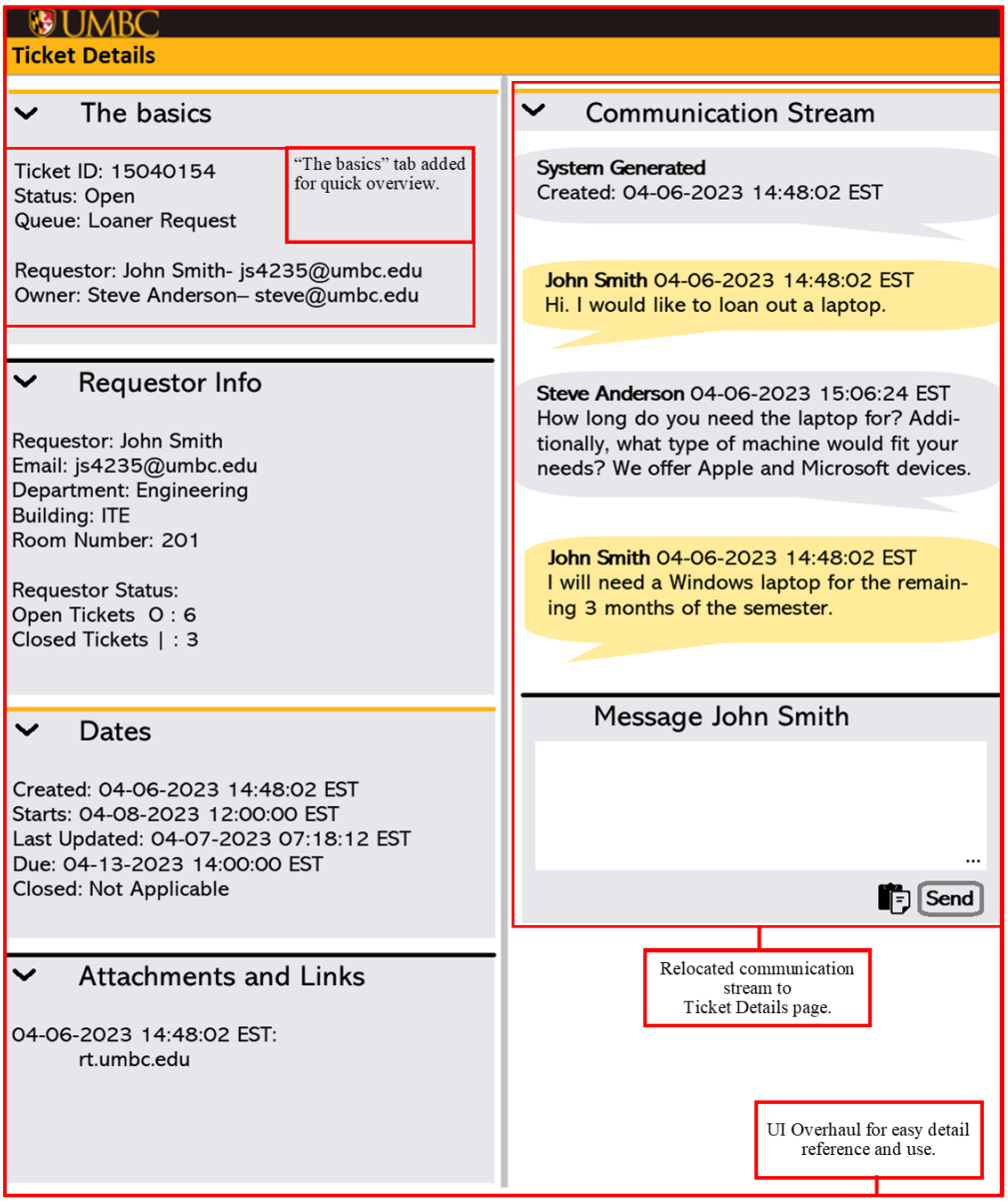
5.2.4 AV Services - At a Glance View:

It's important for staff and management to get a short glimpse of what is going on with their workday. Our team made a big effort to understand the needs of the staff when it came to getting a quick reference of what their day is like ahead. The "At a Glance" view gives a quick overview of everything necessary for staff to be efficient with planning their schedule. This includes integrating a calendar system (such as Google or Workday calendar) to show what appointments or events they have ahead. In addition to the calendar view, it was crucial to have a plug-in for 25Live so that support staff can see if a certain classroom or office was occupied so that they did not interfere with classes or meetings while completing their task.



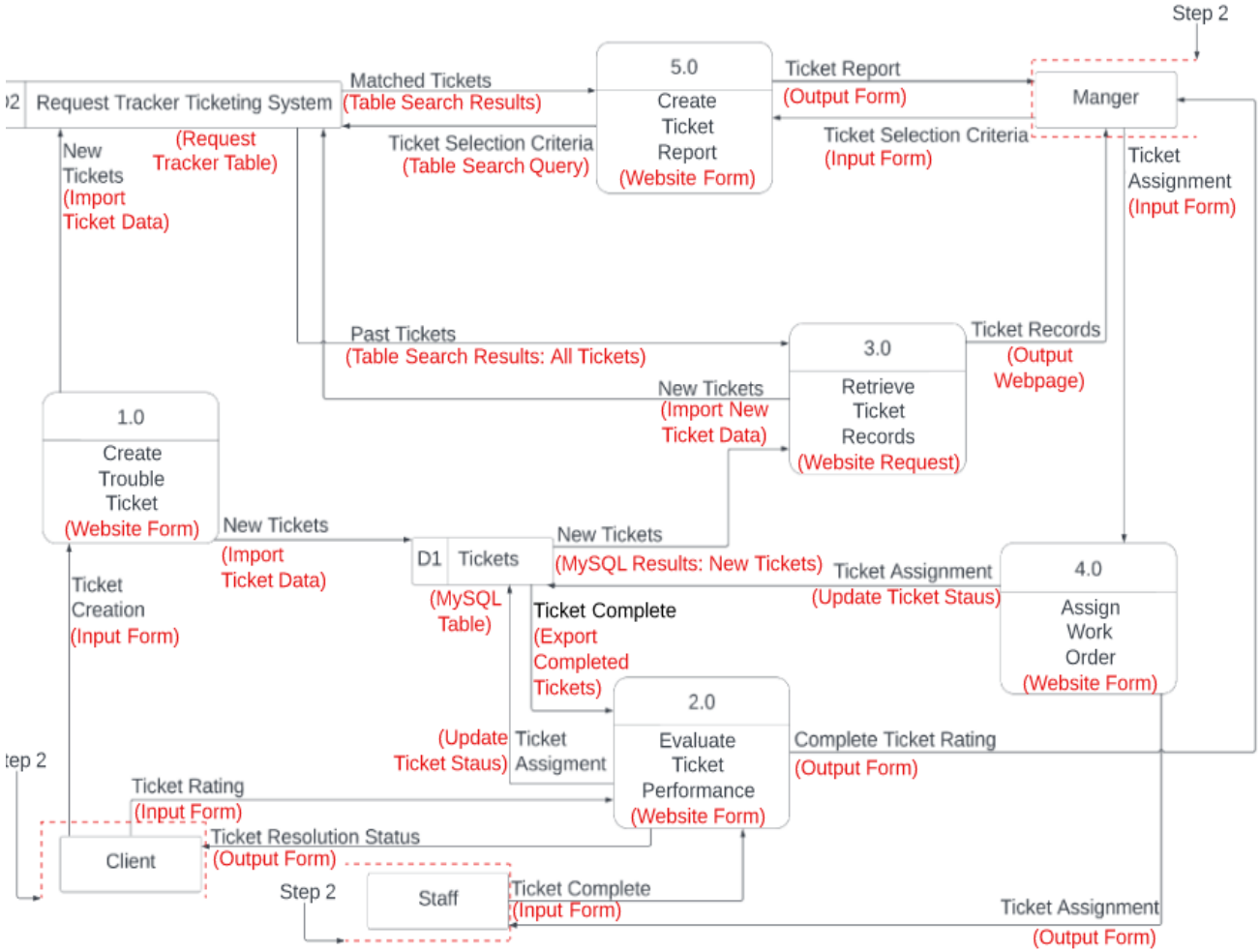
5.2.5 AV Services Ticket Details:

One of the main concerns from the initial interview of our site contact at AV Services was that the current page for ticket details was too crowded with information and interactable boxes. The main goal of the overhaul of this page was to bring all of the useful information forward and relocate small details to dropdowns in easily accessible places on this page. Additionally, our team wanted to make sure that support staff had easy access to communication with the requestor while having the request details readily available. The chat box on the right side of our design prototype shows a typical interaction with a requestor and allows for the support staff to quickly reference any details. They can then forward messages to the requestor on the same page, making updating and responding to requests a simple task.



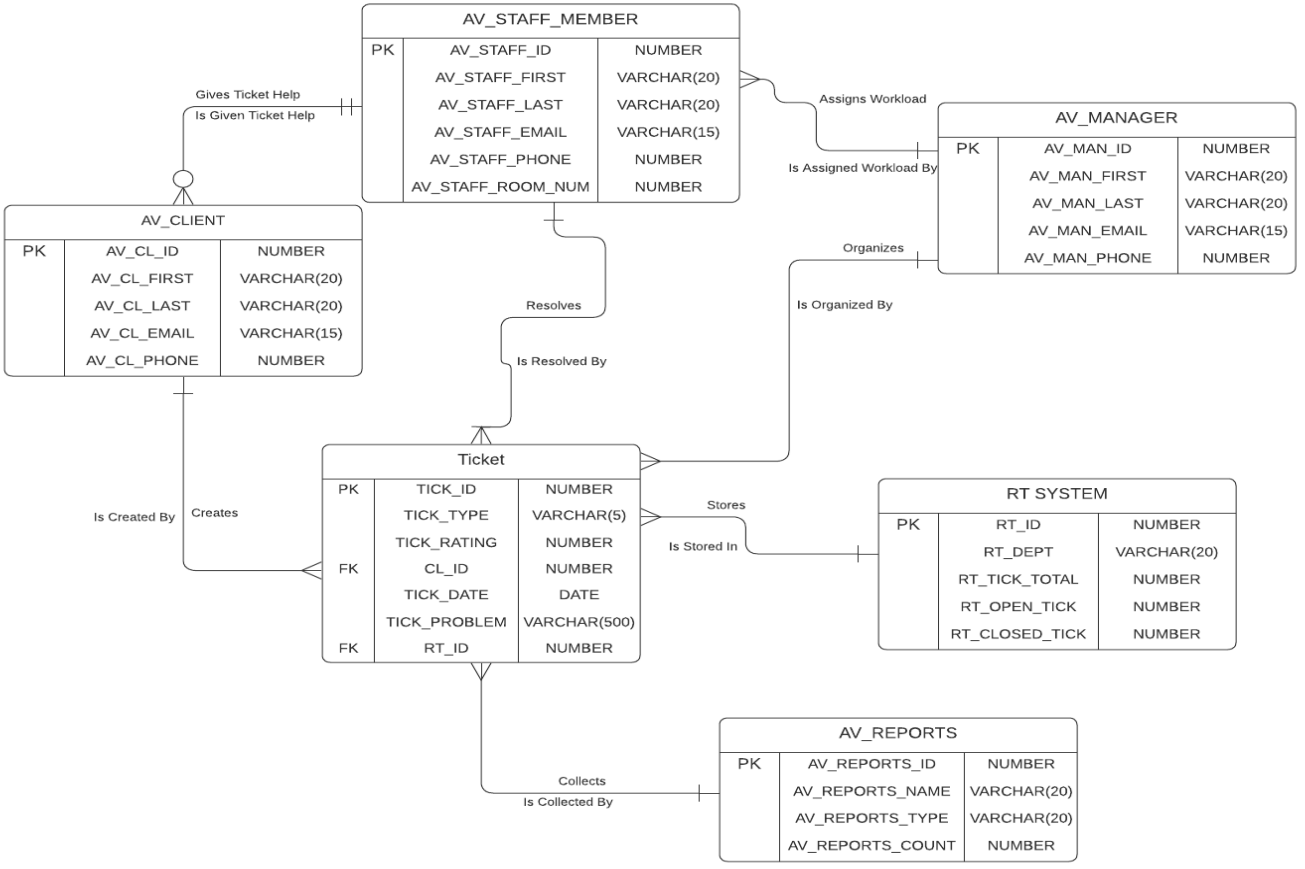
5.3 Physical Process Models:

This next section showcases our physical process model, this is the Level 0 diagram. This diagram was converted from a logical diagram into a physical diagram using the diagramming technique used in the "Process Modeling" lecture. This includes steps such as adding system-related data stores, flows, and processes as well as updating the data elements in the data flows.



5.4 Physical Data Models:

Lastly, we have our physical data model. This model is our Entity Relationship Diagram which was then converted into a detailed physical Entity Relationship Diagram. Some changes to convert it into a physical data model includes adding data type constraints for each field and fixing any identified violations.



Appendix

6.0 Administrator Presentation Notes:

Deliverable 1 Notes:

-Add missing parts of the document
-Add more to problem summary
-Must be clearer for the summary
-Meet at least 250-word count for Business Requirements Description
-Same for business value

Deliverable 2 Notes:

-Generate/Create Trouble Tickets
-Add introductory narrative
-Explain why each functional requirement is important
-Fix spacing between tables
-Reference initial interview transcripts to write better requirements

Deliverable 3 Notes:

-Request Tracking System as data storage.
- (Assign help) -> Don’t put the verb in the data flow name.
-This involves the whole process itself.
-No diagonal lines.
-Single Name Entity names (Manger, staff, client).
-Set parameters is not a valid data flow name:
-Ticket Selection Criteria
-Organize past tickets - Selected Tickets.
-Provide Support (not a good dataflow name). D
-Ticket Resolution Status.
-Narrow it down to the department
-Lines got to be as straight and right-angled as possible
-"Ticket in Parameters" - change to, "ticket selection criteria"
-Provide Support - Change to, "Ticket Status"?
-Consistent size throughout.
-No diagonal lines. Right Angles are preferred. best would be vertical and Horizontal lines.
-0.0 --> 0
- Replace "0.0" with "0"
- Scope down "UMBC DoIT Help Request System" to "AV System"
- Make Request Tracker Ticket Storage a data store
- "Manger, Staff, Client" entity names for Context Diagram

- Change "tickets in parameters" to "Ticket selection criteria"
- Change "Specified Tickets" to "Selected Tickets"
- Change "Provide Support" to "Ticket Status" or "Ticket Resolution Status"
- Change "Assign Help" to "Ticket Assignment" (Try to eliminate verbs)

Deliverable 4 Notes:

-Integrity Violations: RT Systems - Ticket ID is going in the wrong direction. Flip the relationship the other way.
-Add more attributes to RT since it is an intersection entity.
-Same thing with Manager/Staff Member^. Adding First_name, Last_name Try to have entities with similar amounts of attributes.
-Must add reporting entity on the right side of the RT system and have a ticket connected to it.
-Differentiate AV from RT. Include AV more
-Change "helps" and "is helped by" to a more directed verb Work assignment.
-"Manager Assigns Workload" etc. Manager and AV services have a one-to-one relationship

Deliverable 5 Notes:

- Stuff you added to make it clear (circle elements that are added and justification)
- Use call outs
- Include descriptions (a couple per print)
- Ignore the comment about reports
- Change request type and Issue. - request type: Loaner option missing.
- "Add theses issue types"