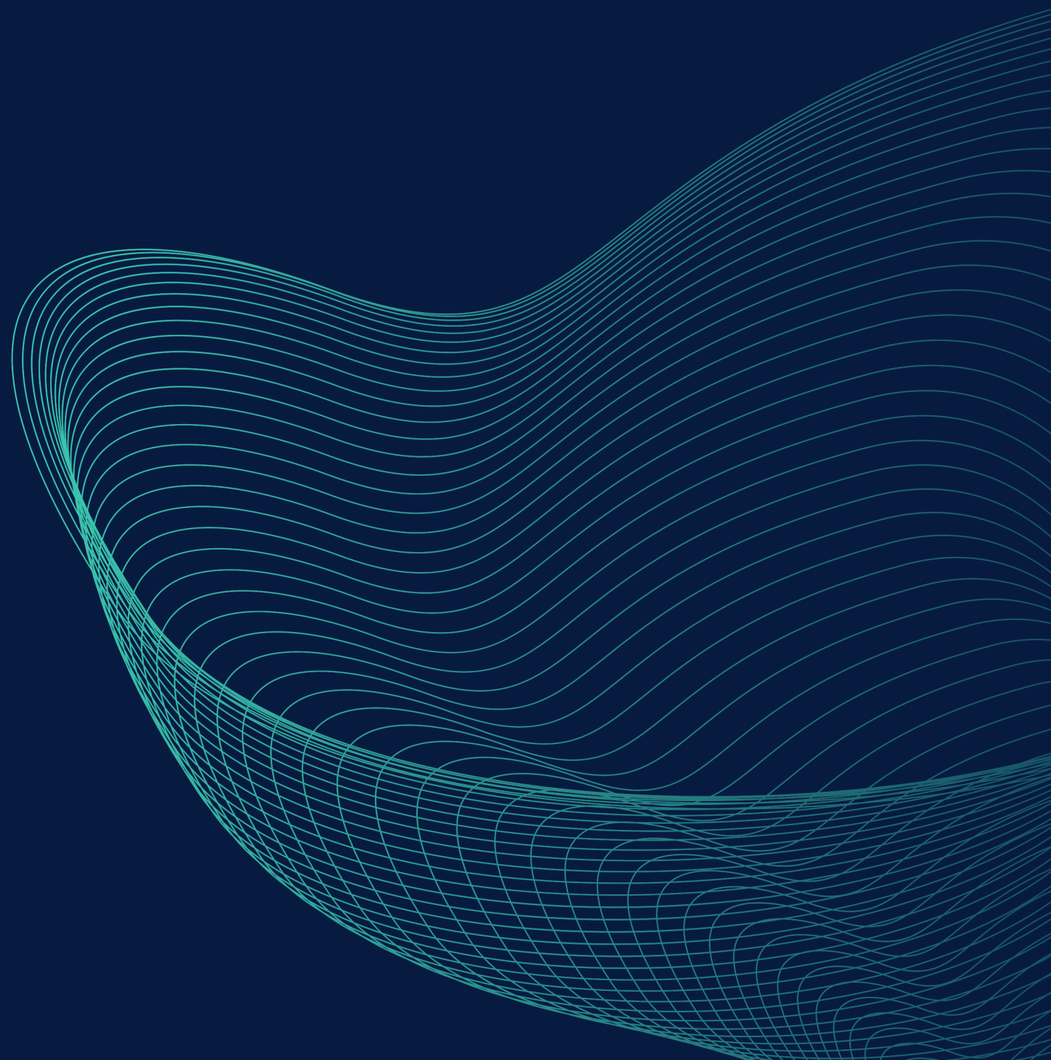
TECHNICAL SERVICE

WE CAN HELP YOU FIX ALL YOUR PROBLEMS



Background

The Tshwane University Of Technology Technical services, is the one who is using the Technical Service system to report technical issues that they are facing. To send a request about a technical issue the requester needs to call or send an email. The admin is the only person who is interacting with the system, acknowledging the request and assigning a task to the relevant technician. Artisan will receive an email about the task and go to the technical service to collect a job card. closing the log provided that the artisan and the requester has signed the job card form.

Problem Statement

Lectures, Supervisors, Residence managers, Lab facilitators and other tut employees occasionally encounter technical difficulties which obstructs them from completing their tasks. When using the current system; the admin has a lot of paperwork, he/ she must enter artisans manually before assigning a task and communication between the admin and artisan rely on phone calls. Additionally, the Admin must interact with the System and staff member who reported the incident. Closing a log is a long procedure. As a result, there are a lot of unclosed logs, some due to staff members not sending feedback. These difficulties slow down productivity and need to be resolved as soon as possible.

Conceptual Solution

Create Technical service system for Tshwane University of Technology to help maintain, manage and solve issues sent by staff members. With the updated version of the Technical service system where the requester will send their request and the admin will view the request and assign the artisan. The artisan will view the task and update the progress of the task to inform both staff and admin about the progress.

Proposed Solution

The technical system's new version will reduce the quantity of paperwork, and the requesters and artisans will be able to engage with the system. The system will be used for all communication, including updates, request tracking, and feedback providing. This new version will help us work more efficiently and respond to requests more quickly.

Introduction

TUT has put in place Technical Services – Help Desk, which aims to resolve technical problems encountered by employees. This is ineffective, tedious, and outdated. We propose an improved version of technical services (the legacy system). The improved technical services system aims to help TUT employees solve technical difficulties effectively. Thus improving productivity and assisting TUT achieve its objectives.

Objective

Implementing a web-based application that will allow staff, admin, and artisans to engage with the system and meet all of their demands would reduce the number of unsolved logs, end paper-based approach and increase productivity. All users will communicate and keep each other up to date in the system. The system will be simple to use and navigate.

User stories

STAFF

1.As a staff I want to send a request about a technical issue so that it can be fixed.
2.As a staff I want to track the request so that I can see the progress of my request.
3.As a staff I want to send feedback so that the admin will be able to see if my request has been solved or not.

ADMIN

1. As an admin I want to login, so that I can access the system.
2. As an admin I want to view the request, so that I can assign the task to the relevant technician.
3. As an admin, I want to set the priority of the request, so that I can inform technician about the task that are important.
4 As an admin I want to assign a task to available technicians, so that they can start working on the request.
5. As an admin I want to view the progress status of the task, so that I can know the progress of the task.
6. As an admin, I want to generate a report, so that I can know which requests are pending, solved and unresolved.
7. As an admin, I want to export a report, so that I can document requests.
8. As an admin I want to close the log, so that I can know which logs have been resolved.

User stories

ARTISAN

1. As an artisan, I want to log in, so that I can access the system.
2. As an artisan, I want to view a task assigned to me, so that I can start working on it.
3. As an artisan, I want to update the task progress, so that I can inform the staff and admin about the progress.

Functional requirements

Requirement ID:	Functional Requirement Description	Not Mandatory	Purpose
IDNo.	The system must/should		
FR001	allow staff, admin, artisan to login.	Mandatory	Login to access system functions
FR002	allow staff to submit requests.	Mandatory	so that the problem may be resolved.
FR003	notify the staff of the reference number.	Mandatory	to be aware of the request number and use it to monitor the status of the request.
FR004	allow the staff to track progress.	Mandatory	so that the staff can know about the progress of the request.
FR005	allow staff to submit the feedback.	Mandatory	To determine whether the request has been resolved.
FR006	allow admin to view request	Mandatory	To delegate the task to the appropriate technician.
FR007	allow the admin to set the priority of a log.	Mandatory	To make technicians aware of the critical tasks that must be completed by the deadline.
FR008	allow the admin to assign a task.	Mandatory	So that the artisan can be aware of the request and can begin working on it.

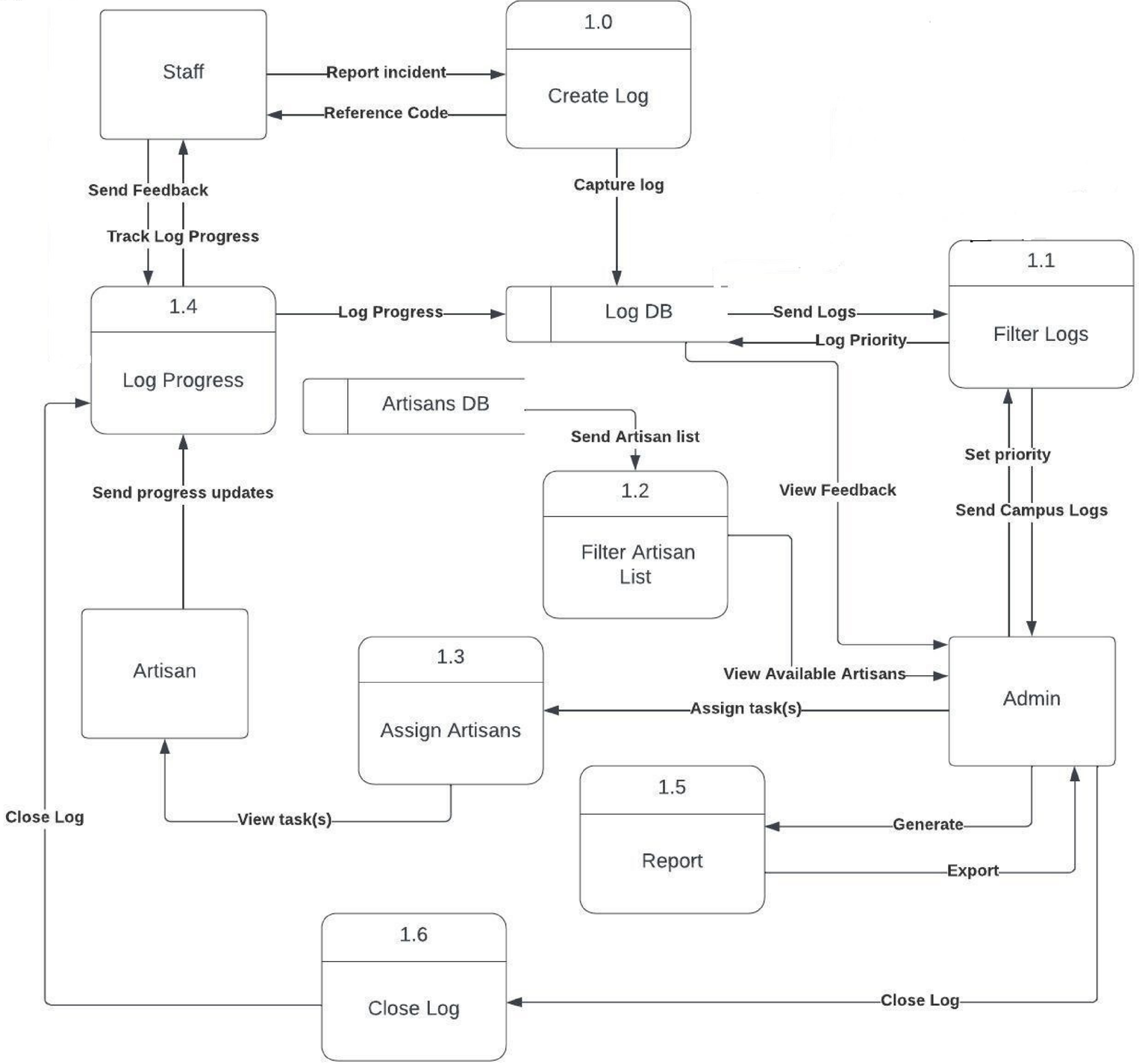
Functional Requirement ID:	Functional Requirement Description	Mandatory/ Not Mandatory	Purpose
FR009	allow the admin to view the artisan’s progress on a task	Mandatory	To keep track of the task's progress.
FR010	allow artisan to view a task.	Mandatory	Begin working on the task/request.
FR011	allow the technician to update task progress.	Mandatory	To keep the requester and admin updated on the status of the request.
FR012	allow the admin to close the log.	Mandatory	The problem has been resolved.
FR013	allow admin to generate the report	Mandatory	To keep track of requests sent, solved, pending, and so on.
FR014	allow the admin to export the report.	Mandatory	Requests must be documented.

Non-Functional requirements

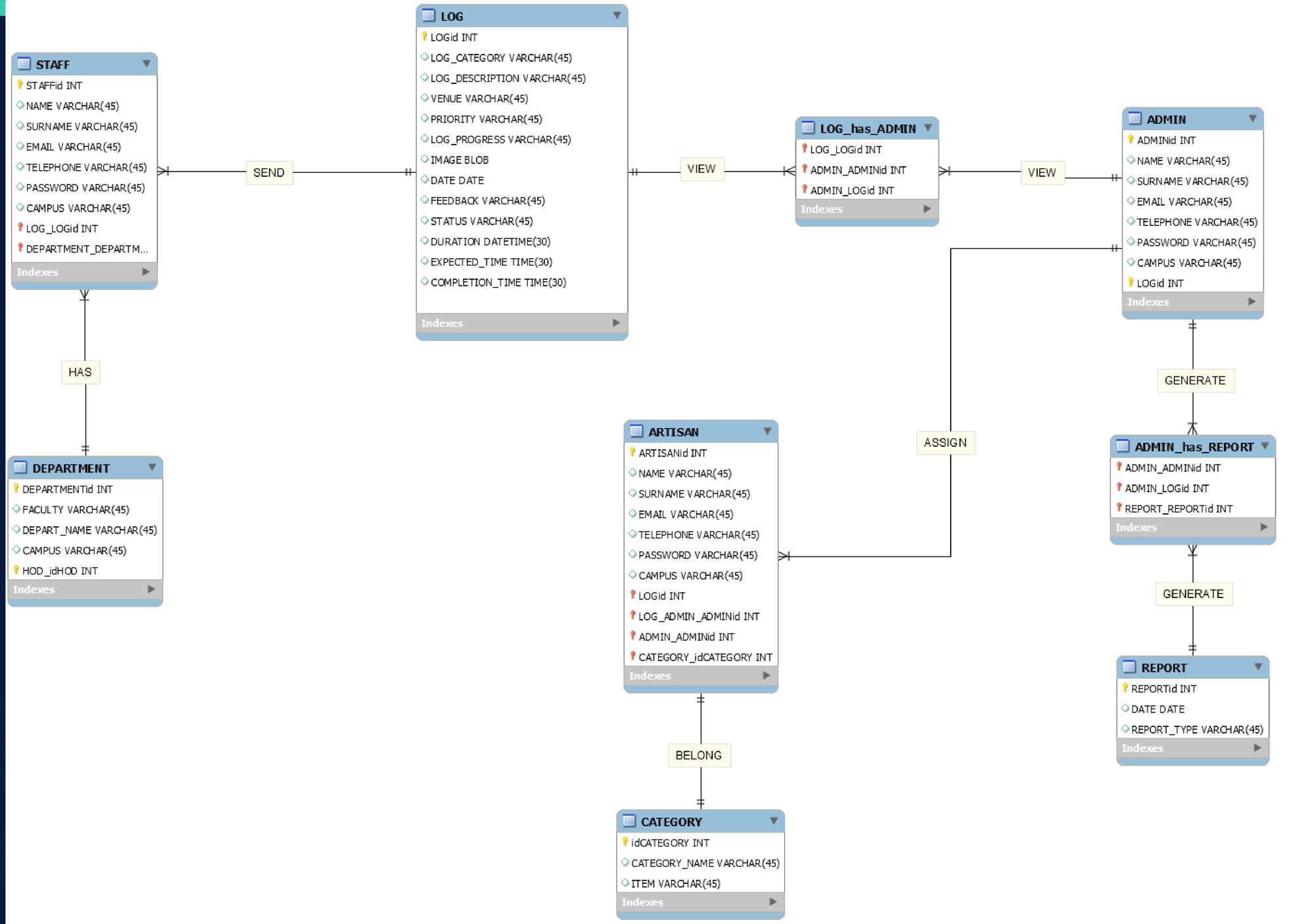
Non-Functional Requirement ID:	Non-Functional Requirement Description	Mandatory/ Not Mandatory	User Expectations
NFR001	Security: The system needs to protect all stored data against malware attacks and unauthorized access.	Mandatory	The system must be able to secure sensitive user data.
NFR002	Scalability: The system must be able to support hundreds of users at the same time.	Mandatory	The system must be able to handle hundreds of users at the same time.
NFR003	Performance: The system must have a responds time of less than thirty seconds.	Mandatory	The system must be able to reply in thirty seconds or less.
NFR004	Usability: The system must be easy for TUT personnel to use.	Mandatory	Each employee must be able to use the system without difficulty.

Non-Functional Requirement ID:	Non-Functional Requirement Description	Mandatory/ Not Mandatory	User Expectations
NFR005	Capacity: The system must have a storage capacity of four terabytes or more.	Mandatory	More data must be stored by the system.
NFR006	Compatibility: The system can be accessed through any devices that has internet access.	Mandatory	The system must be compatible to any device that a user has access to.
NFR007	Reliability: The system must be able to perform all its functions without failure.	Mandatory	The system must be capable of conducting all its operations without fail.
NFR008	Availability: The system needs to be available 24/7.	Mandatory	Access the system anytime.

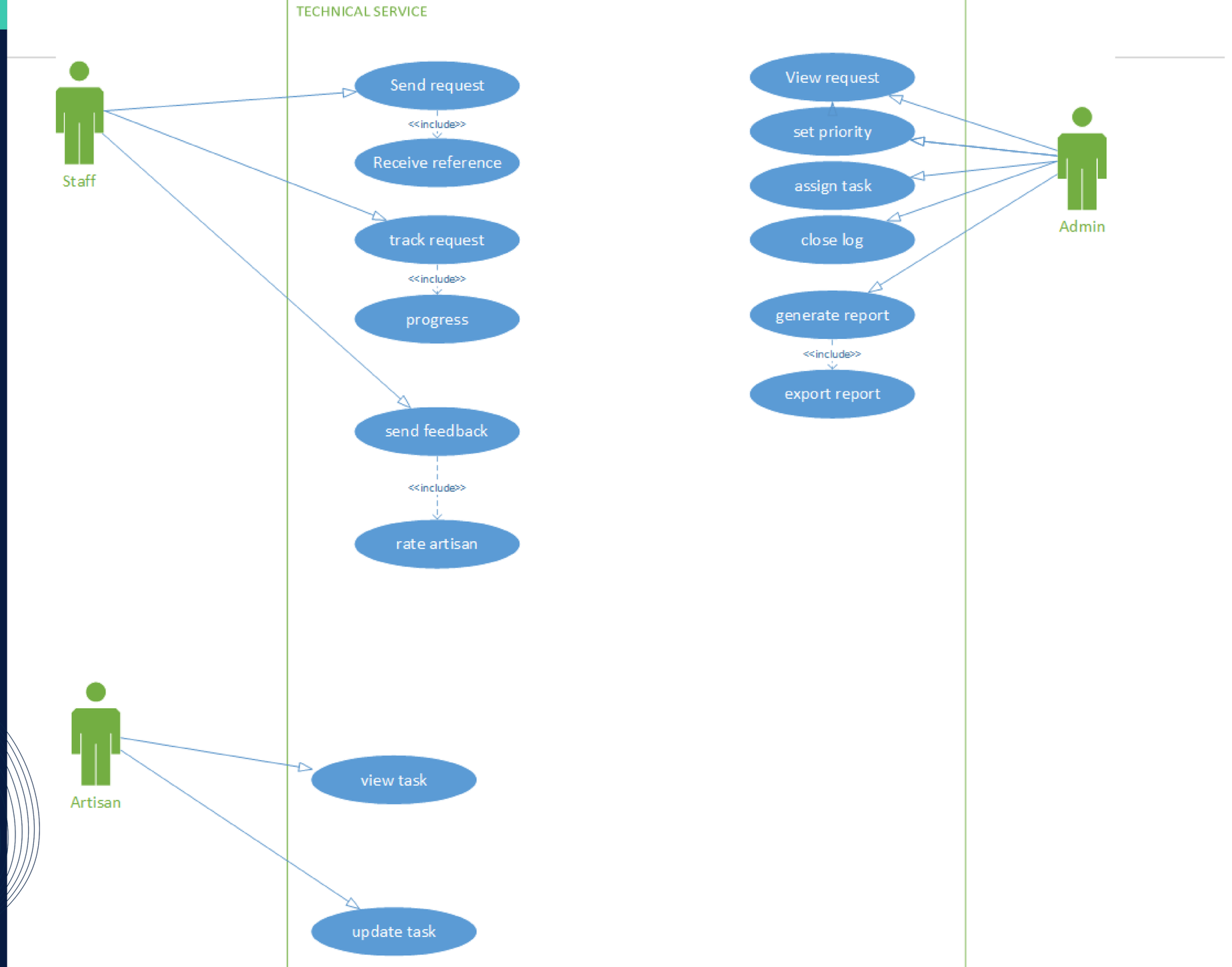
DATA FLOW DIAGRAM



ERD

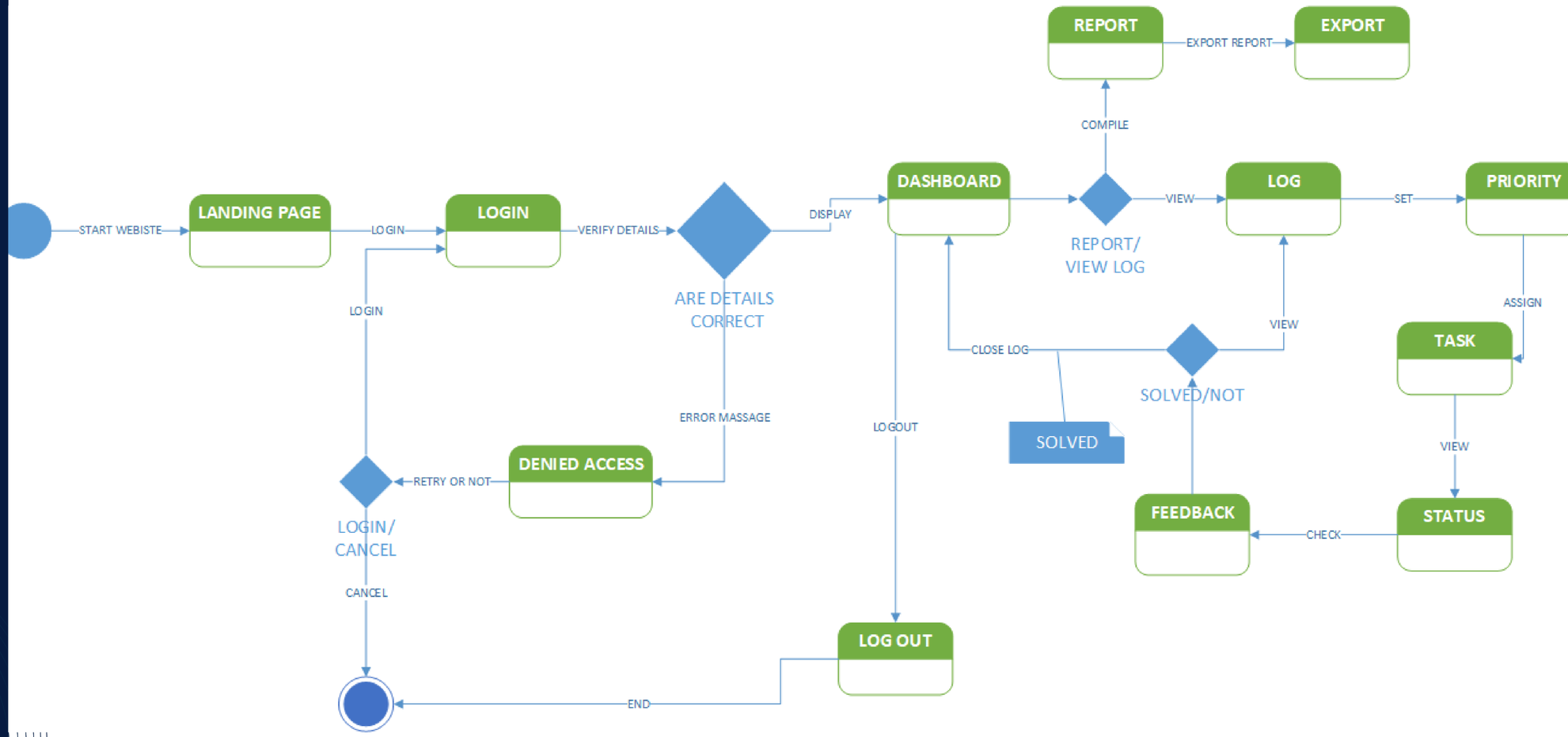

STAFF
STAFFid INT
NAME VARCHAR(45)
SURNAME VARCHAR(45)
EMAIL VARCHAR(45)
TELEPHONE VARCHAR(45)
PASSWORD VARCHAR(45)
CAMPUS VARCHAR(45)
LOG_LOGid INT
DEPARTMENT_DEPARTM...
Indexes
SEND
LOG
LOGid INT
LOG_CATEGORY VARCHAR(45)
LOG_DESCRIPTION VARCHAR(45)
VENUE VARCHAR(45)
PRIORITY VARCHAR(45)
LOG_PROGRESS VARCHAR(45)
IMAGE BLOB
DATE DATE
FEEDBACK VARCHAR(45)
STATUS VARCHAR(45)
DURATION DATETIME(30)
EXPECTED_TIME TIME(30)
COMPLETION_TIME TIME(30)
Indexes
VIEW
LOG_has_ADMIN
LOG_LOGid INT
ADMIN_ADMINid INT
ADMIN_LOGid INT
Indexes
VIEW
ADMIN
ADMINid INT
NAME VARCHAR(45)
SURNAME VARCHAR(45)
EMAIL VARCHAR(45)
TELEPHONE VARCHAR(45)
PASSWORD VARCHAR(45)
CAMPUS VARCHAR(45)
LOGid INT
Indexes
HAS
DEPARTMENT
DEPARTMENTid INT
FACULTY VARCHAR(45)
DEPART_NAME VARCHAR(45)
CAMPUS VARCHAR(45)
HOD_idHOD INT
Indexes
ARTISAN
ARTISANid INT
NAME VARCHAR(45)
SURNAME VARCHAR(45)
EMAIL VARCHAR(45)
TELEPHONE VARCHAR(45)
PASSWORD VARCHAR(45)
CAMPUS VARCHAR(45)
LOGid INT
LOG_ADMIN_ADMINid INT
ADMIN_ADMINid INT
CATEGORY_idCATEGORY INT
Indexes
ASSIGN
GENERATE
ADMIN_has_REPORT
ADMIN_ADMINid INT
ADMIN_LOGid INT
REPORT_REPORTid INT
Indexes
GENERATE
REPORT
REPORTid INT
DATE DATE
REPORT_TYPE VARCHAR(45)
Indexes
BELONG
CATEGORY
idCATEGORY INT
CATEGORY_NAME VARCHAR(45)
ITEM VARCHAR(45)
Indexes

USE CASE


TECHNICAL SERVICE
Staff
Send request
<<include>>
Receive reference
track request
<<include>>
progress
send feedback
<<include>>
rate artisan
View request
set priority
assign task
close log
generate report
<<include>>
export report
Admin
Artisan
view task
update task

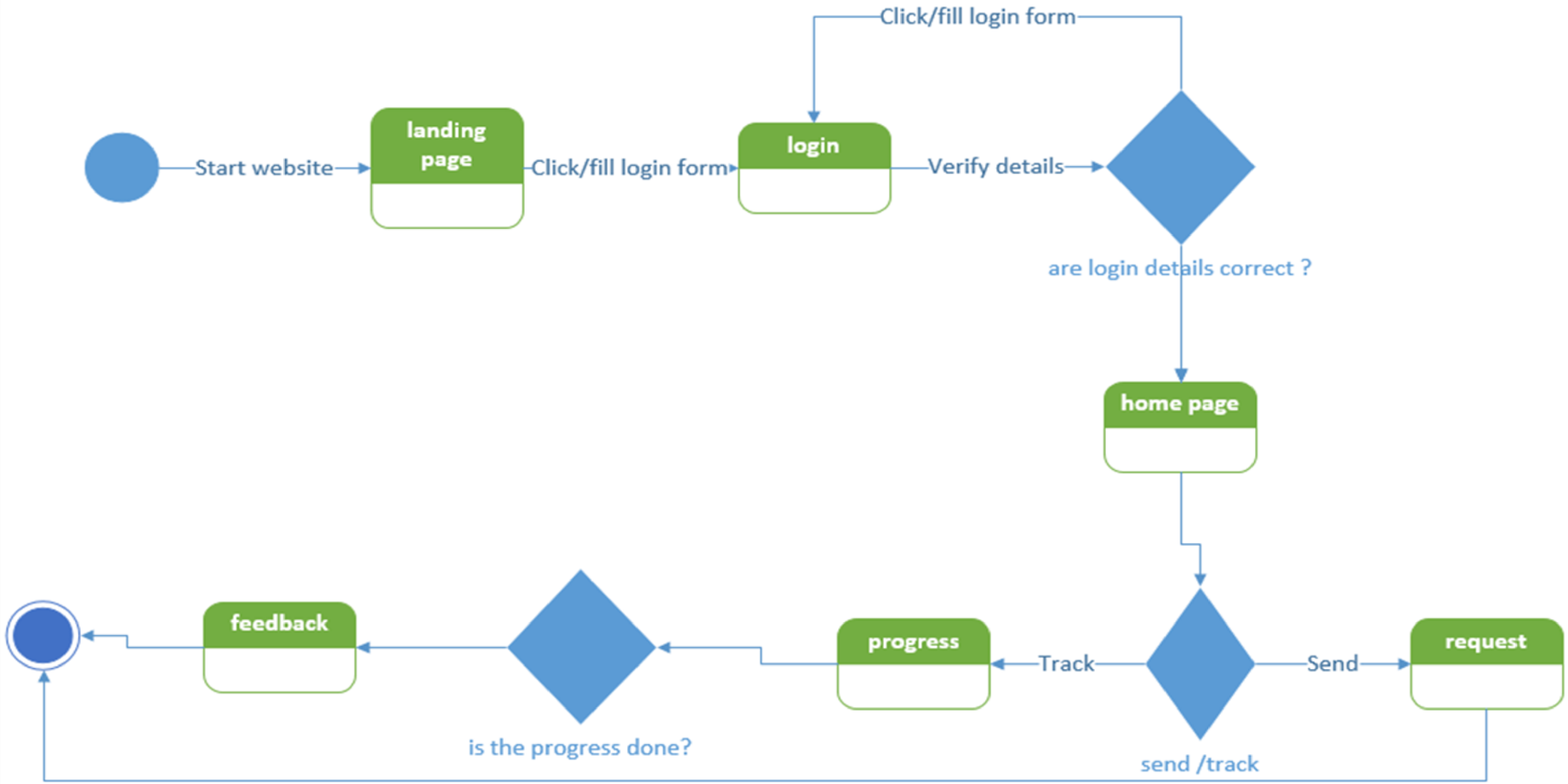
STATE MACHINE

ADMIN



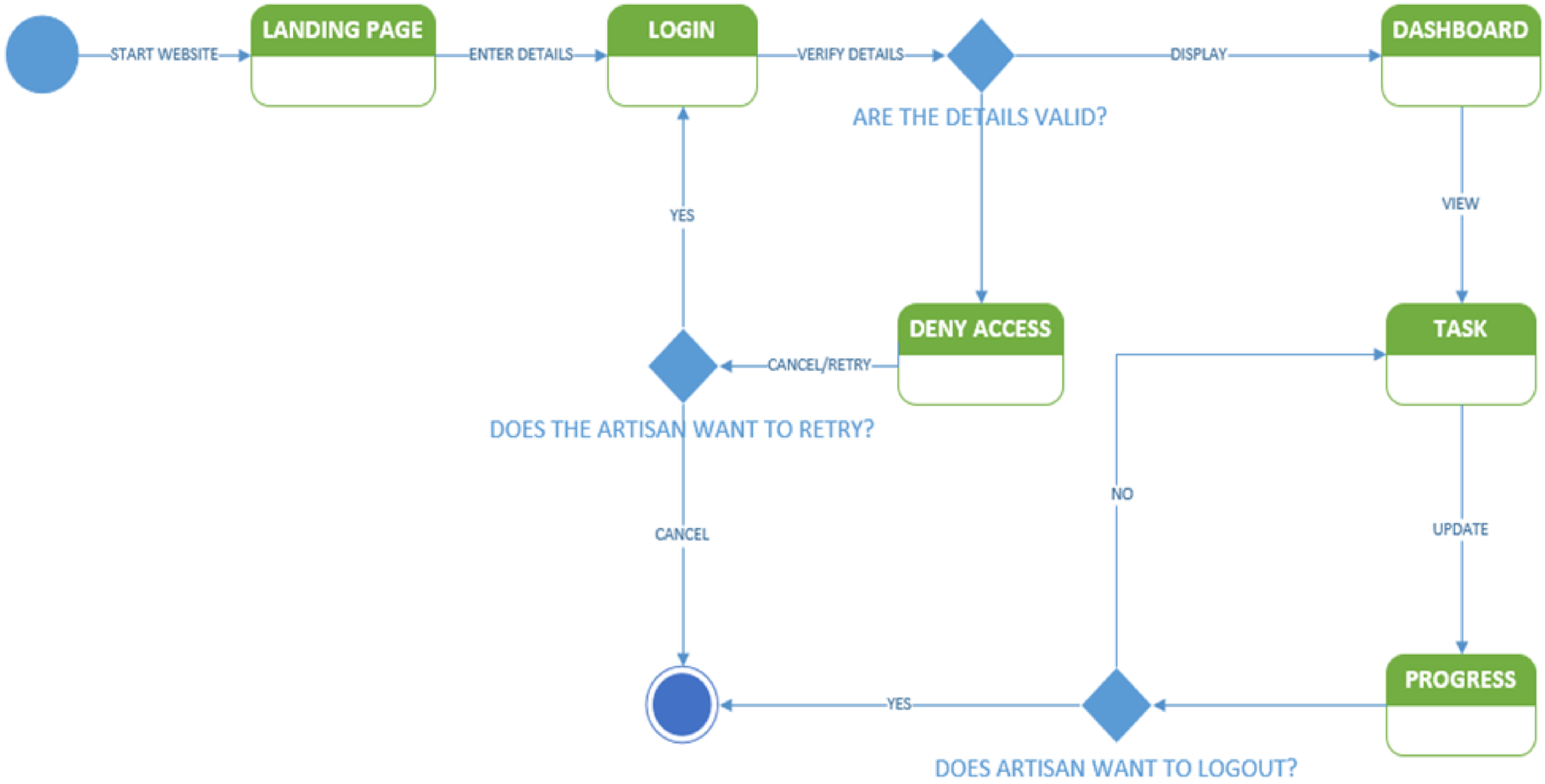
STATE MACHINE

STAFF

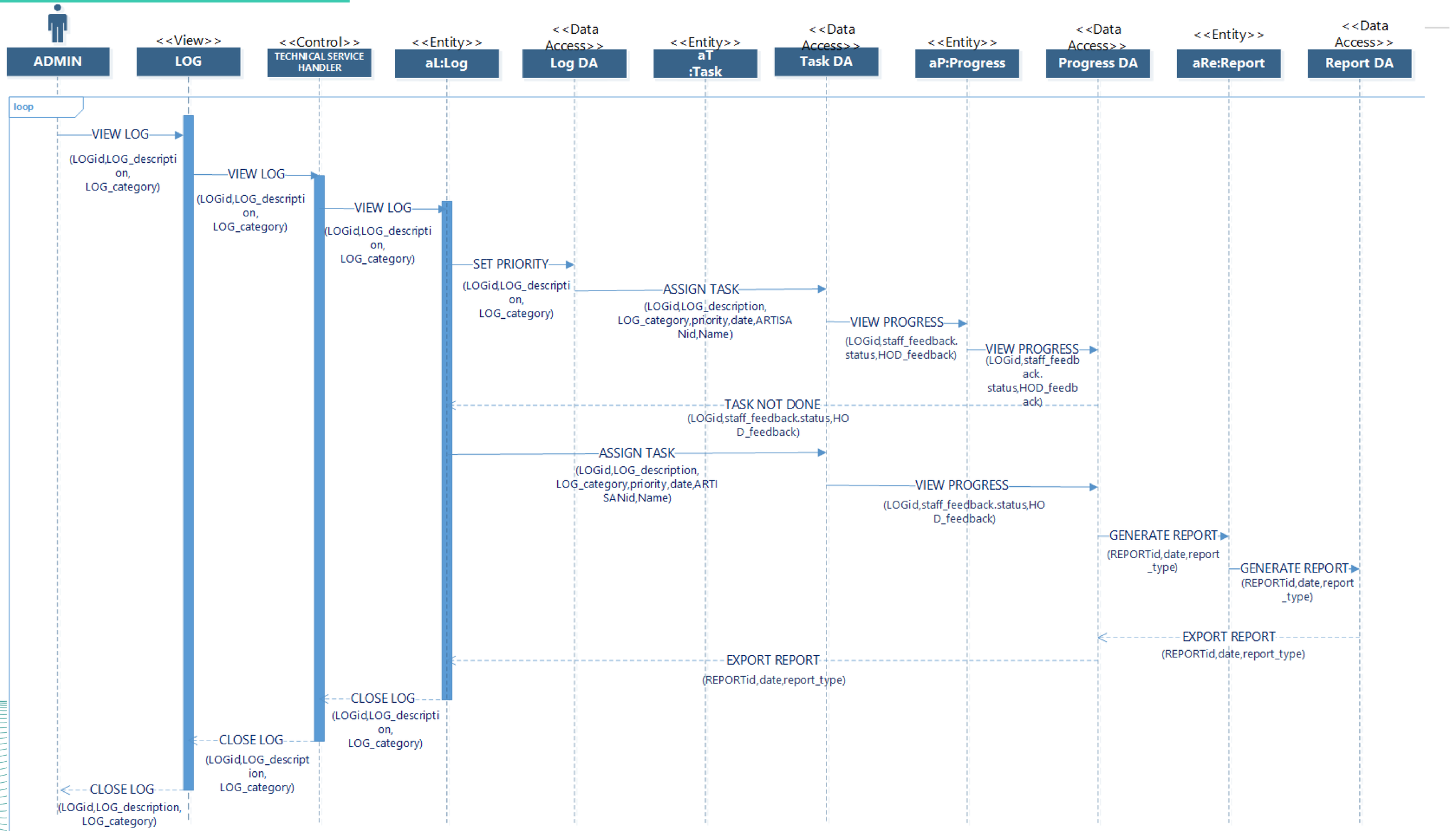


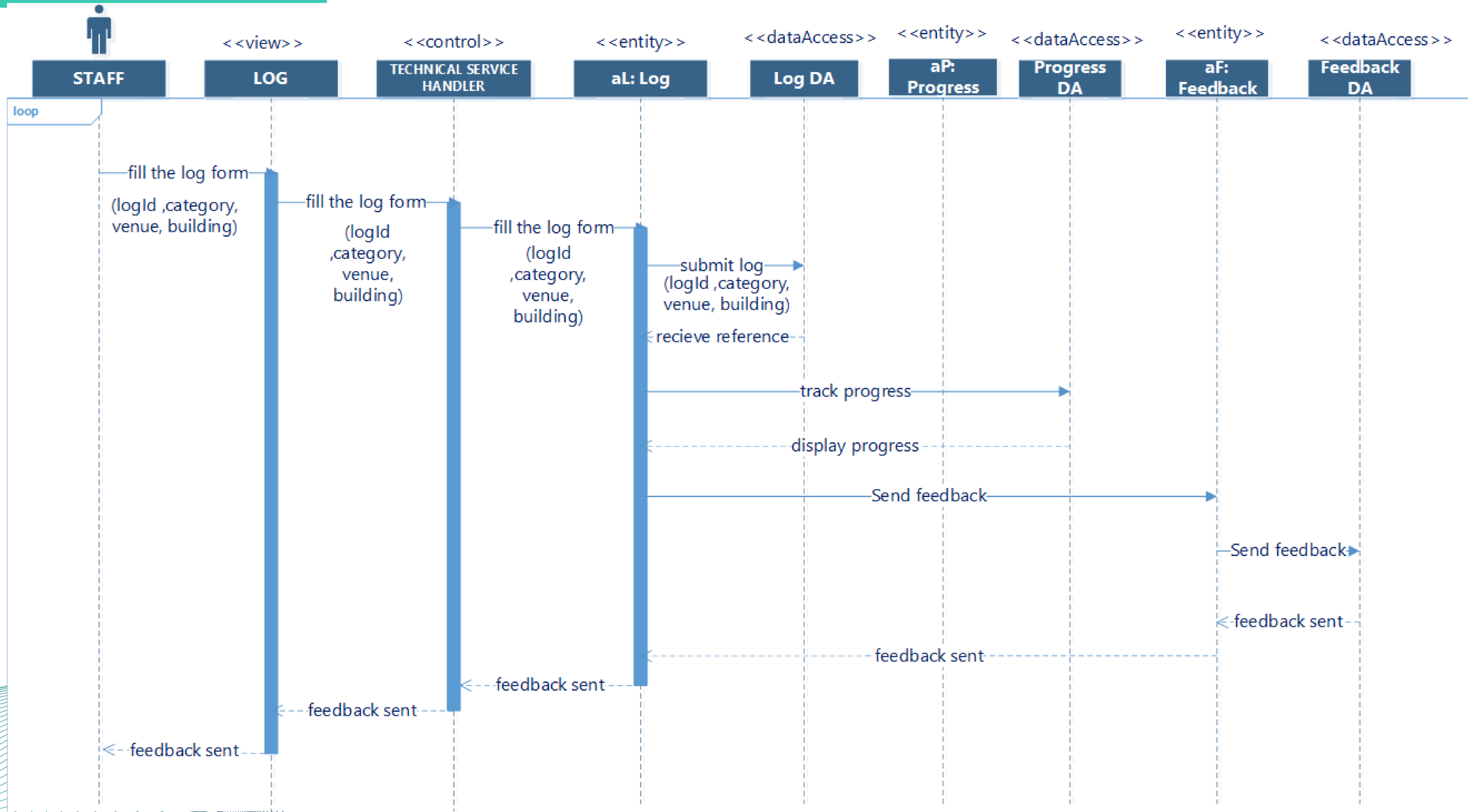
STATE MACHINE

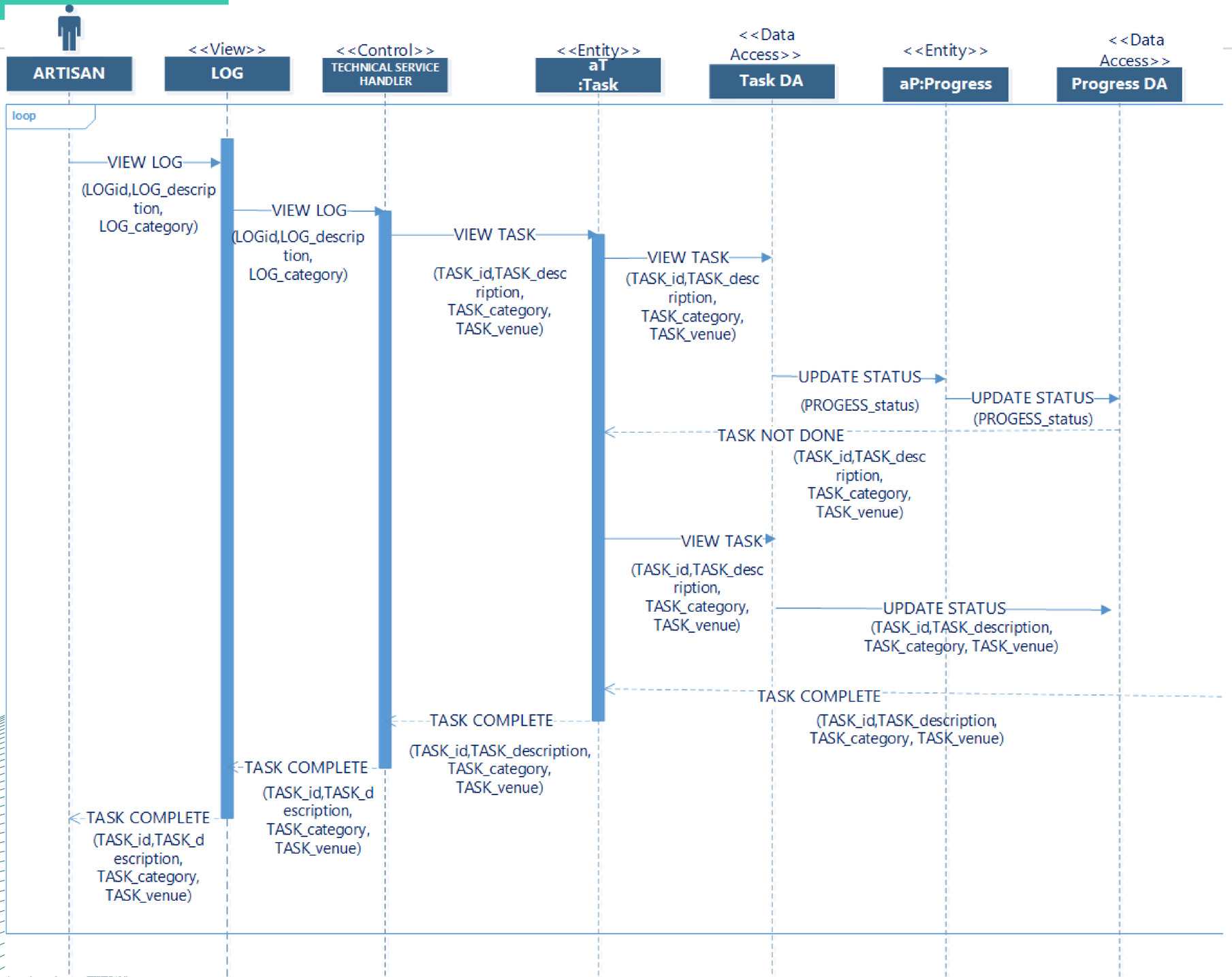
ARTISAN



SEQUENCE DIAGRAM


ADMIN
<<View>>
LOG
<<Control>>
TECHNICAL SERVICE HANDLER
<<Entity>>
aL:Log
<<Data Access>>
Log DA
<<Entity>>
aT :Task
<<Data Access>>
Task DA
<<Entity>>
aP:Progress
<<Data Access>>
Progress DA
<<Entity>>
aRe:Report
<<Data Access>>
Report DA
loop
VIEW LOG
(LOGid,LOG_description, LOG_category)
VIEW LOG
(LOGid,LOG_description, LOG_category)
VIEW LOG
(LOGid,LOG_description, LOG_category)
SET PRIORITY
(LOGid,LOG_description, LOG_category)
ASSIGN TASK
(LOGid,LOG_description, LOG_category,priority,date,ARTISANid,Name)
VIEW PROGRESS
(LOGid,staff_feedback.status,HOD_feedback)
VIEW PROGRESS
(LOGid,staff_feedback.status,HOD_feedback)
TASK NOT DONE
(LOGid,staff_feedback.status,HOD_feedback)
ASSIGN TASK
(LOGid,LOG_description, LOG_category,priority,date,ARTISANid,Name)
VIEW PROGRESS
(LOGid,staff_feedback.status,HOD_feedback)
GENERATE REPORT
(REPORTid,date,report_type)
GENERATE REPORT
(REPORTid,date,report_type)
EXPORT REPORT
(REPORTid,date,report_type)
EXPORT REPORT
(REPORTid,date,report_type)
CLOSE LOG
(LOGid,LOG_description, LOG_category)
CLOSE LOG
(LOGid,LOG_description, LOG_category)
CLOSE LOG
(LOGid,LOG_description, LOG_category)


STAFF
<<view>>
LOG
<<control>>
TECHNICAL SERVICE HANDLER
<<entity>>
aL: Log
<<dataAccess>>
Log DA
<<entity>>
aP: Progress
<<dataAccess>>
Progress DA
<<entity>>
aF: Feedback
<<dataAccess>>
Feedback DA
loop
fill the log form
(logId ,category, venue, building)
fill the log form
(logId ,category, venue, building)
fill the log form
(logId ,category, venue, building)
submit log
(logId ,category, venue, building)
recieve reference
track progress
display progress
Send feedback
Send feedback
feedback sent
feedback sent
feedback sent
feedback sent


ARTISAN
<<View>>
LOG
<<Control>>
TECHNICAL SERVICE HANDLER
<<Entity>>
aT :Task
<<Data Access>>
Task DA
<<Entity>>
aP:Progress
<<Data Access>>
Progress DA
loop
VIEW LOG
(LOGid,LOG_description, LOG_category)
VIEW LOG
(LOGid,LOG_description, LOG_category)
VIEW TASK
(TASK_id,TASK_description, TASK_category, TASK_venue)
VIEW TASK
(TASK_id,TASK_description, TASK_category, TASK_venue)
UPDATE STATUS
(PROGESS_status)
UPDATE STATUS
(PROGESS_status)
TASK NOT DONE
(TASK_id,TASK_description, TASK_category, TASK_venue)
VIEW TASK
(TASK_id,TASK_description, TASK_category, TASK_venue)
UPDATE STATUS
(TASK_id,TASK_description, TASK_category, TASK_venue)
TASK COMPLETE
(TASK_id,TASK_description, TASK_category, TASK_venue)
TASK COMPLETE
(TASK_id,TASK_description, TASK_category, TASK_venue)
TASK COMPLETE
(TASK_id,TASK_description, TASK_category, TASK_venue)
TASK COMPLETE
(TASK_id,TASK_description, TASK_category, TASK_venue)

Recommendation

Technical service system will be a system that will be easy to navigate and it will make things easier for the Tshwane University Of Technology staff members. It will help them to save time by decreasing a lot of work that the admin will be doing by allowing other users to also interact with the system. By using this system TUT will no longer have some many unresolved requests or logs.

Conclusion

The Technical Services department is overloaded with unclosed and unresolved requests from TUT employees. This is an obstacle to employees' productivity as it generally prevents them from doing their work. The current system is unable to meet the demands of the institution. We conclude that the system needs to be updated to quickly resolve incidents experienced by TUT employees.

ANY QUESTIONS?

Thank you!!