Chapter 3

RESEARCH METHODOLOGY

This chapter discusses the methodology which will be used in this study. The first section describes the research design supported by different models which will be used in the development of the proposed system. This chapter also showcases the detailed processes on how the prototype will be developed. Ideally, concepts discussed in this chapter cover the totality from the blue print of the project up to the detailed specification of each components and modules in the project: Research design, business processes and technical considerations which were illustrated using different diagrams. In addition, the bill of materials and costing as well as the evaluation and implementation plans are also discussed in this chapter.

I. Research Design

Design and development phases are far way the most lengthy among the entire project development stages. It is used to establish product requirements and process requirements. It is also used to establish product design and detailed specifications in order to identify design outputs. From the design specifications, several methods are used to start the development of the project. The theories and concepts identified in the chapter 2 of this documentation served as the basis in finalizing the design specifications of this project. Once the design has been finalized, the prototype will be developed. However, the finalized design is not limited to the design specification prior to the actual development. It is always subject for numerous possible changes that needs to be applied in order to satisfy the needs of the project functionalities.

The specification is probably the most important section of a design project if all the research has been carried out. The specification draws on the information collected and presented during the research section. It is a number of straightforward statements, made clearly outlining the nature of the project to be designed and manufactured.

In order to fully discuss the specifications of the project, this documentation outlined the details of the prototype by illustrating several diagrams highlighting the end-to-end details of the project.

II. Project Development

Methodology

Methodology describes a means of performing a task. It exists at many different levels such as corporate methodology, product/ process methodology and design methodology. (Hall, 2012)

The correct choice of a methodology is used to constantly re-examine new ways to improve processes. Similar to any hardware or software development methodologies, the researcher chose a methodology which have been applied in developing the project. The researcher considered numerous factors in choosing the appropriate methodology such as resources, technical skills, and the time constraints. Considering such factors, the researcher decided to make use of a throwaway prototyping.

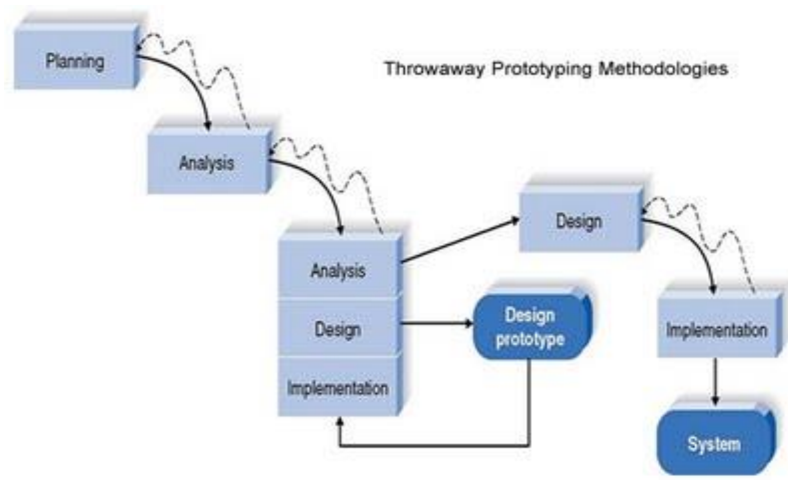

Figure 8: Diagram of Throwaway Prototyping Model

Throwaway Prototyping is a technical mechanism to reduce project risk by exploring factors critical to projects success (Cons, 2012)

A prototype developed as part of a throw-away approach will not form part of the final solution. It is likely to inform the final solution, but the prototype itself will not become part of the final solution.

Throw-away prototypes are a useful way of exploring ideas, and gaining feedback from the client and/or end-user. They tend to be used to answer questions. They are then discarded. (Scot, 2012)

Like any project development methodologies, the standard phases like planning, analysis and design are included in the prototyping methodology. The big difference occurs in the design phase where the prototype is being refined as many times as possible until it meets the requirements of the project. In detail, the throwaway prototyping methodology includes the following phases:

Planning Phase

Planning includes analysis of project requirements in terms of input data and desired output, processing required to transform input into output, cost-benefit analysis, and schedule of the project. The feasibility analysis also includes the technical feasibility of a project in terms of available software tools, hardware, and skilled software professionals. At the end of this phase, a feasibility report for the entire project is created.

Planning phase includes the processes of analyzing the feasibility, developing the work plan and staffing the project. In this phase, the researcher identified the series of tasks which has to be done, created the schedule of activities and identified the feasibility of the project. The researcher conducted numerous data gathering procedures in order to justify the existence of the problem and to further strengthen that present solutions are not yet efficient and effective to be solved.

Analysis Phase

Analysis phase includes gathering, analysis, validating, and specifying requirements. At the end of this phase, the Software Requirement Specification (SRS) document is prepared. SRS is a formal document that acts as a written agreement between the development team and the customer. SRS acts as input to the design phase and includes functional, performance, software, hardware, and network requirements of the project. The researcher finalized the scope of the project. After which, appropriate research was done to identify the needed hardware and software requirements.

Design Phase (Quick Design- Initial Prototype)

Design phase includes translation of the requirements specified in the SRS into a logical structure that can be implemented using various engineering, programming and mechanics principles. The output of the design phase is a design document that acts as an input for all the subsequent phases. The researcher created the initial prototype from scratch. This served as the quick design prototype. Since this prototype is done quickly, the functionalities are not yet complete and accurate.

Implementation (Quick Design- Initial Prototype)

After series of prototype evaluation, the initial prototype was revised accordingly. This process served as the actual revision stage where correction and comments from the concluded evaluation were applied. After the application, the prototype has been re-evaluated to identify if further errors has been overlooked.

Design Phase (Actual Prototype)

The completion of the initial prototype after a series of evaluation and refinement will be copied to a brand new prototype. This is used to eliminate possible damages done on the initial prototype during the refinement stages.

Implementation (Actual Prototype)

The final product of the design in an actual prototype will be the implementation of such. In this case, the product will be tested using various testing methodologies. Once done, the final project will be documented and recommendations for further enhancements will be addressed.

Project Costing

Cost data for a project may be associated with activities or groups of activities, or with resources, such as tools or equipment. Further, for any project, there may be several ways in which costs need to be summarized and accounted for.

In general, the development of the prototype incur the following expenses:

Table 5: Project Cost Listing

Category	Item	QTY	Amount	Total	Notes
Development/ Hardware					
	Developmental PC	1	0	0	*Company provided*
	Tablet PC	1	0	0	*Company provided*
	Smart Phone	1	0	0	*Company provided*
	Internet Services	1	0	0	*Company provided*
Development/ Software					
	MS SQL Server 2012	1	0	0	*Company provided*
	Visual Studio 2013	1	0	0	*Company provided*
	MS Office 2013	1	0	0	*Company provided*
	Adobe Photoshop CS 6	1	0	0	*Company provided*
Documentation					
	Bond Paper	2	250.00	500.00	
	Printer	1	2,000.00	2,000.00	
	Ink Refills	1	300.00	300.00	
	Binding	5	100.00	500.00	
Other Expenses					
	Transportation			500.00	
	Photocopy			100.00	*Related Studies and Lit*
	Contingency			3,000.00	
GRAND TOTAL				**6,900.00**	

Majority of the tools which will be needed for the development of the project are readily available as they are already provided by International School Manila. Most of the required expenses are for the documentation and other uncategorized items.

System Design

In a study, it is implied to know that the way to solve a large problem is to break it into a set of interacting smaller problems. Each of these smaller problems can then be decomposed into even smaller problems, until after enough iterations to have a problem that can be solved on its own. Each decomposition gives a set of components, and deciding what those components are and how they fit together is the activity of system design.

The following UML Diagrams describe the overall system design of the proposed ISM Service Request Management System. Each diagram shows the functional requirements as well as the movement of data and activities across different entities in the proposed project.

Use Case Diagram

The below use case model shows an overview of the usage requirements for the proposed system. The use cases were e numerated to come up with the actual requirements.

Four primary users were identified as the actors: Employee, IT Staff, Helpdesk Officer and the IT Director. The Employee are the ones requesting for different IT services and assistants. The IT Staff resolves the service requests. Ultimately, the overall turnaround of the service request processes are supervised and maintained by a helpdesk officer. Reports are then submitted to the IT Director. The IT Director’s primary tasks is to be involved in escalated tickets, perform organizational changes based on the IT Staff’s performance, utilization and workloads, and ensure that the needed IT services for the ISM community is well met and properly managed.

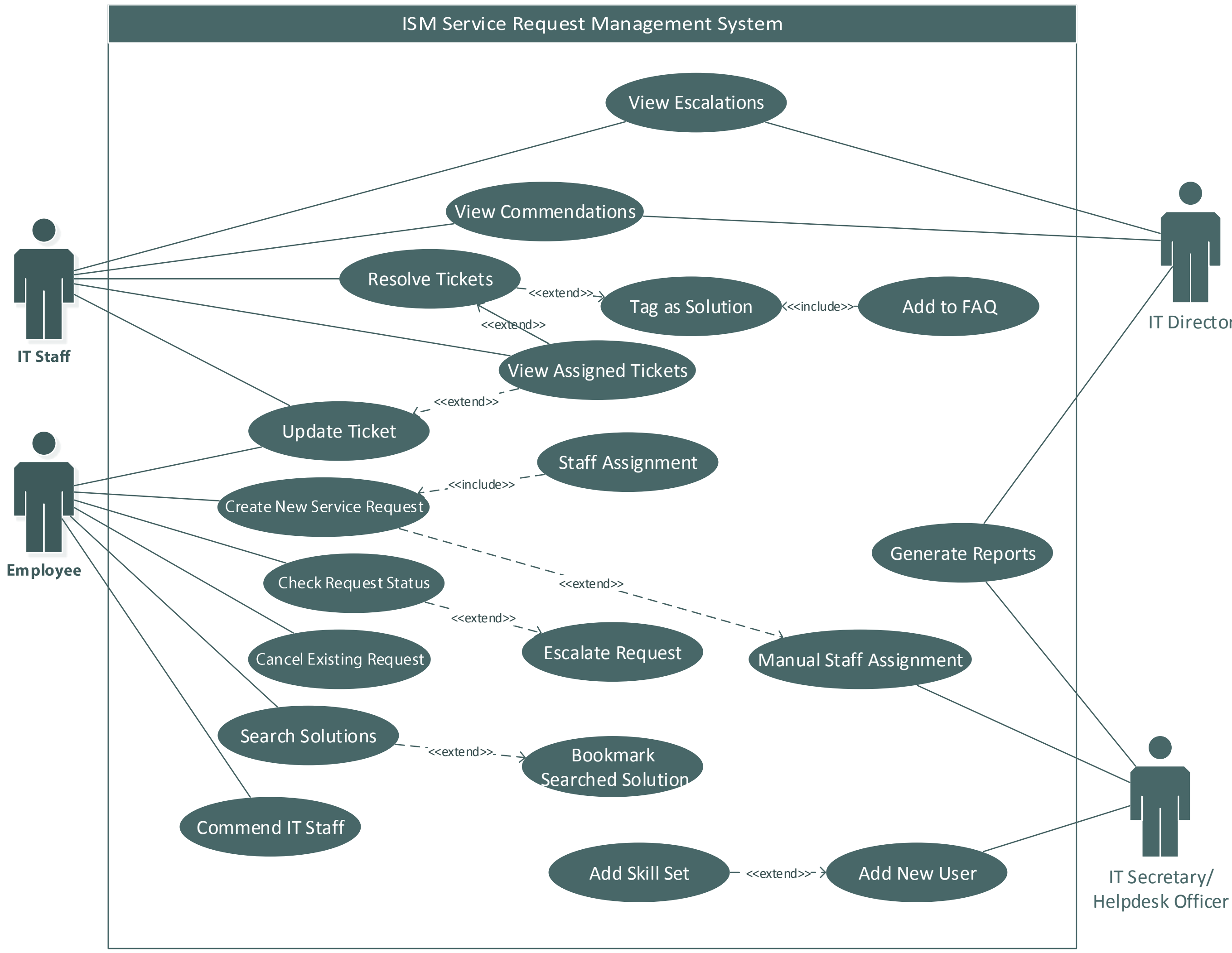

Figure 9: Use Case Model for the proposed ISM Service Request Management System

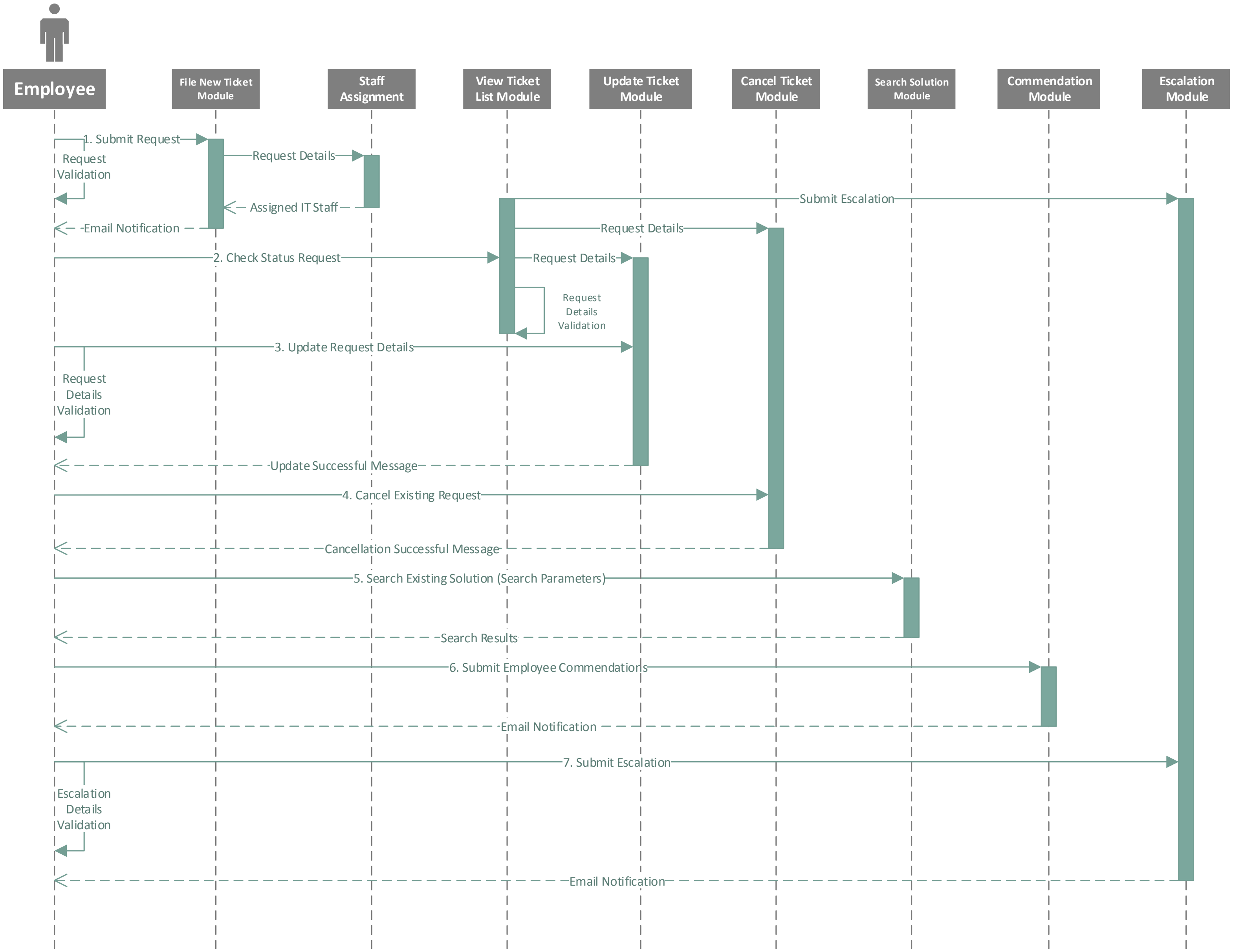

Figure 10: Sequence Diagram: Employee

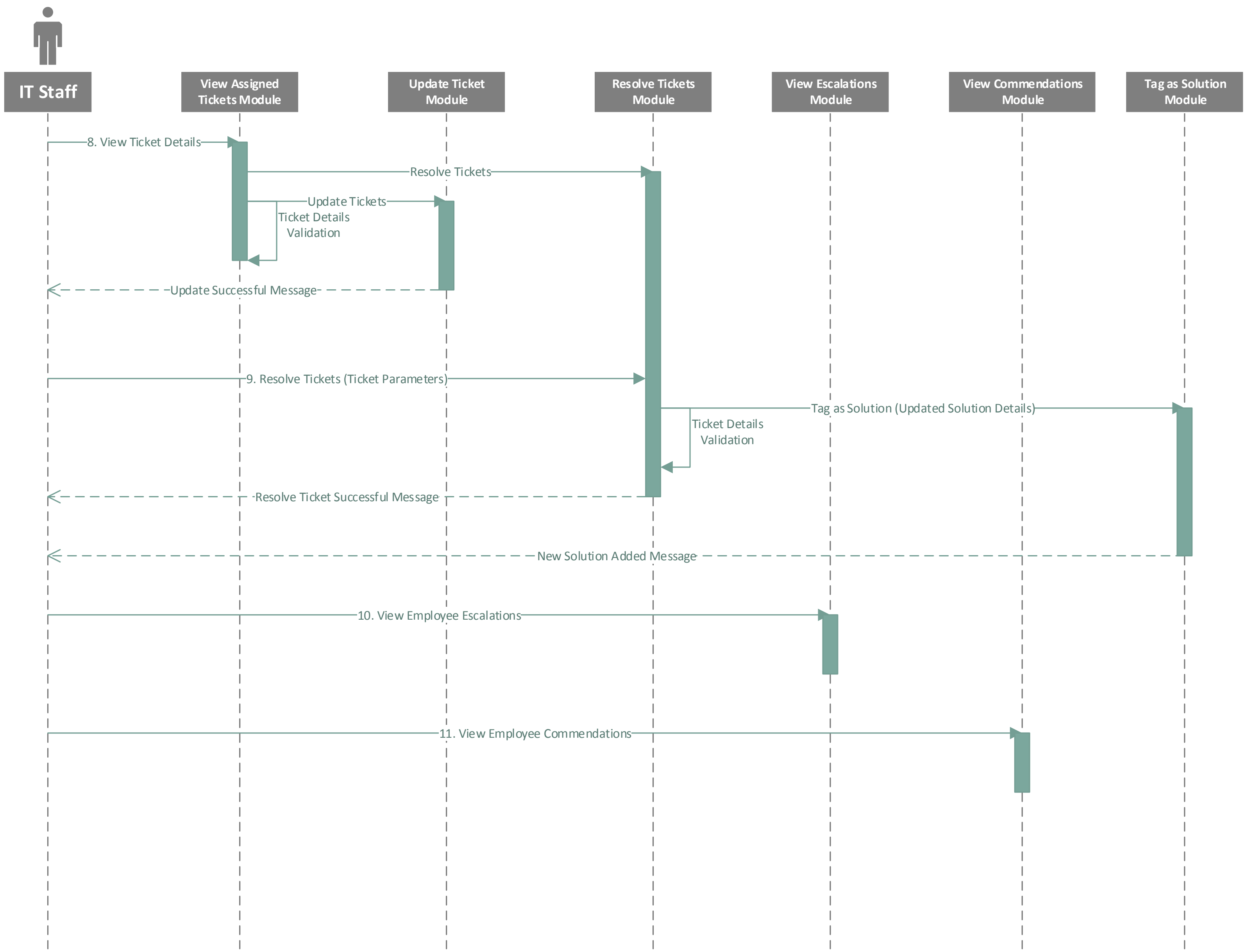

Figure 10: Sequence Diagram: IT Staff

ISM Service Request System Sequence Diagram

The models shown in Figures 9 and 10 are important parts of design analysis. They improve the understanding of the process. The sequence model identifies the core process workflows following the requirements workflow that are dependent upon the use-case model.

It is very notable based on the sequence diagrams that input validations will be highly considered during the design and development of the system. In addition, appropriate error messages will also be integrated to improve the system's user friendliness.

The shown diagrams show the sequence of the appearance of the different modules depending on the activity of the user. Email notifications are also part of confirmation of certain activities or actions made in the system. Notification is an important part of the system for these are used to inform system users of events that have occurred. They can also be used as reminders for matters that have not been taken care of. It is sufficient if these notifications will be sent via e-mail for most of the staff of International School Manila regualrly checks and works with their emailss.

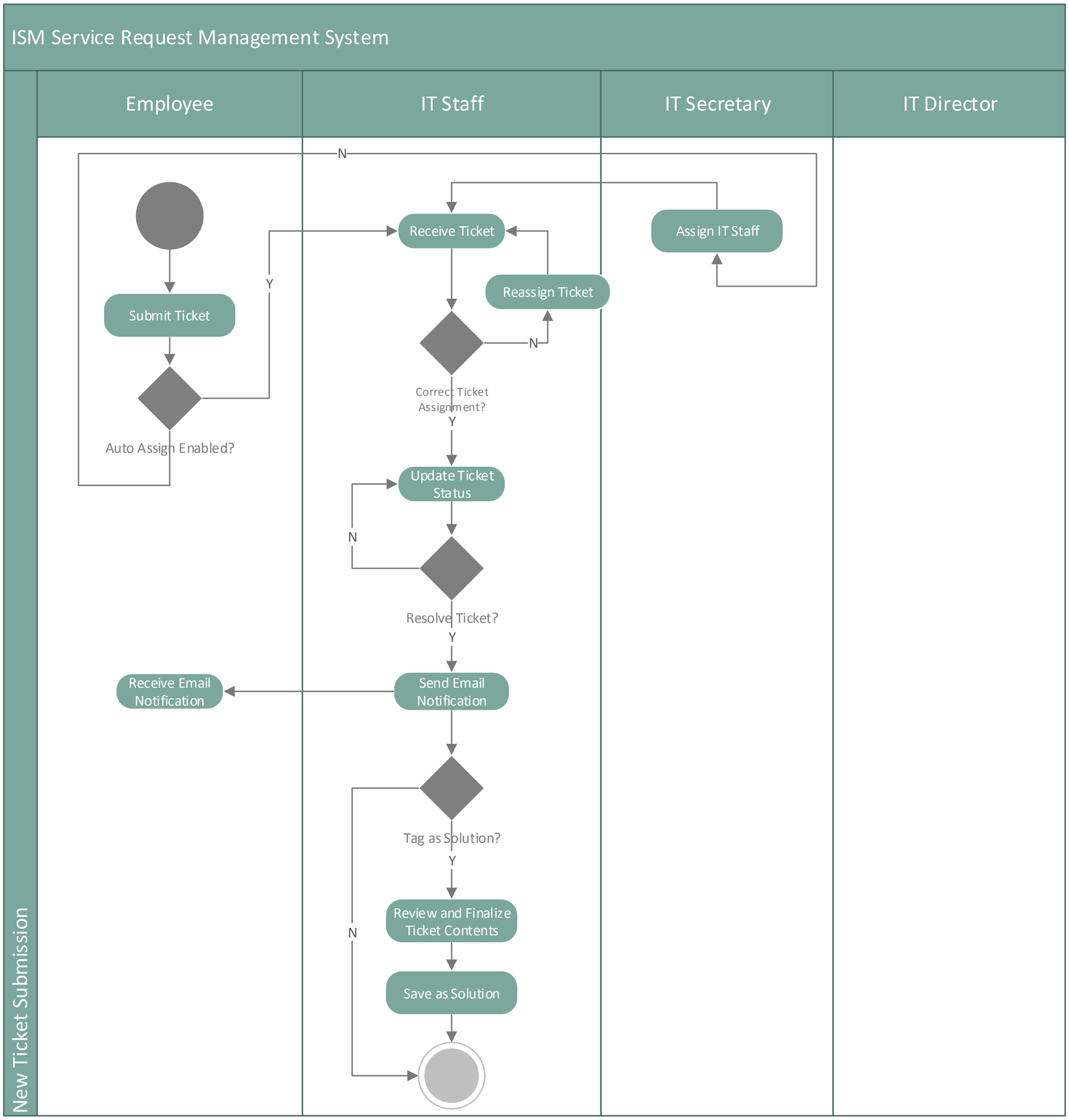

Figure 12: Activity Diagram- New Ticket Submission

New Ticket Submission

The process starts with the ISM employees who will be requesting for a service request. If the auto assign module of the system is enabled, a specific IT staff will automatically assigned for the new request. On the other hand, if auto assign is not enabled, the Helpdesk Officer or the IT Secretary would have to manually assign it to a specific IT staff. The assigned IT staff will receive a notification of a service request via email. In certain conditions that the assigned IT staff is not capable of solving the service request, the IT staff can re assign the ticket to other IT staff. The assigned IT staff will be updating the ticket status whenever needed and will stay in contact with the requestor for possible additional requirements or documents needed. Once the ticket or service request has already been resolved, the requestor will be notified. The IT Staff will also have the option to tag the ticket as part of the solutions in the knowledge database for documentation and can serve as future reference.

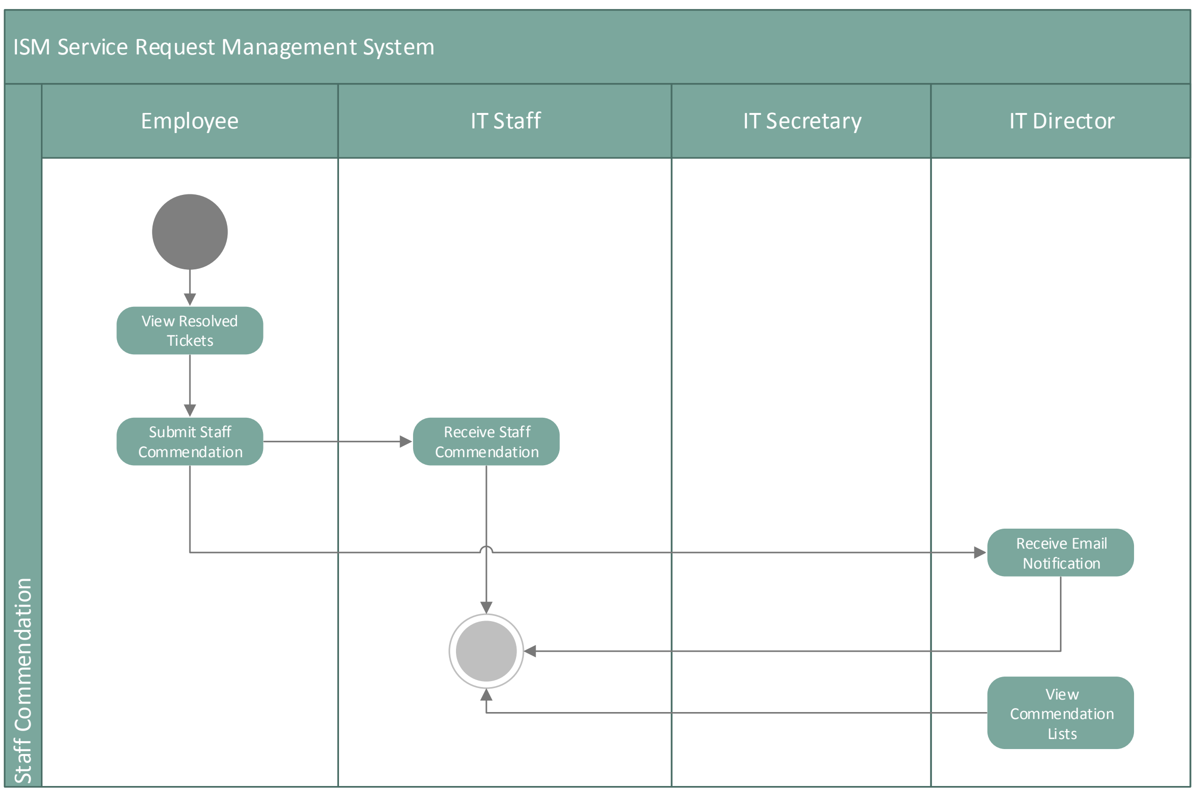

Figure 13: Activity Diagram- Staff Commendation

Staff Commendation

After a turnaround of requesting a ticket to solving it, the employee can commend on the performance of the IT Staff. From the list of resolved tickets, the employee can submit a staff commendation. The employee should choose a specific IT Staff, provide the details of the commendation and submit it. The IT Staff together with the IT Director will receive an email notification to the commendation submitted by an employee.

Ticket Escalation

In contrary to a commendation, there are instances that several tickets were not prioritized, delayed or unresolved that ends up unsatisfied customers (employees). Escalating a ticket allows a ticket to maintain its status, but reflect that the ticket has been escalated for further research, help, or approval, involving the IT Director.

As shown in Figure 14, the employee can view the list of unresolved tickets. From the list, the employee can select one and submit it for escalation. Both the assigned IT Staff and the IT Director will be notified via email with such escalation. Several communication (external to the system) will happen to resolve the ticket. Once resolved, the ticket will be closed and similar to the previous process, the IT staff has the option to document it and add it to the knowledge database for future references. Escalated tickets which were solved are more likely needs to be documented.

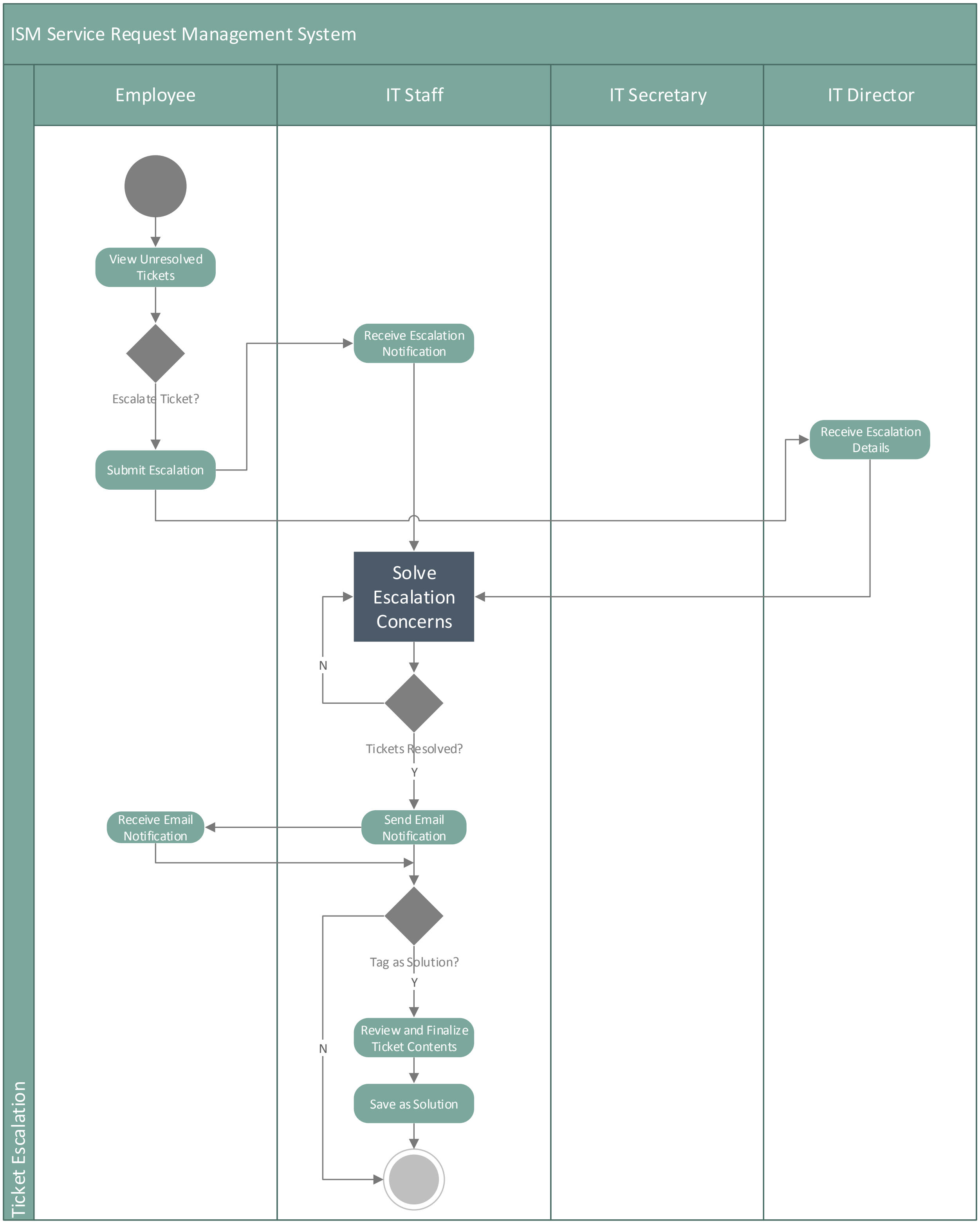

Figure 14: Activity Diagram- Ticket Escalation

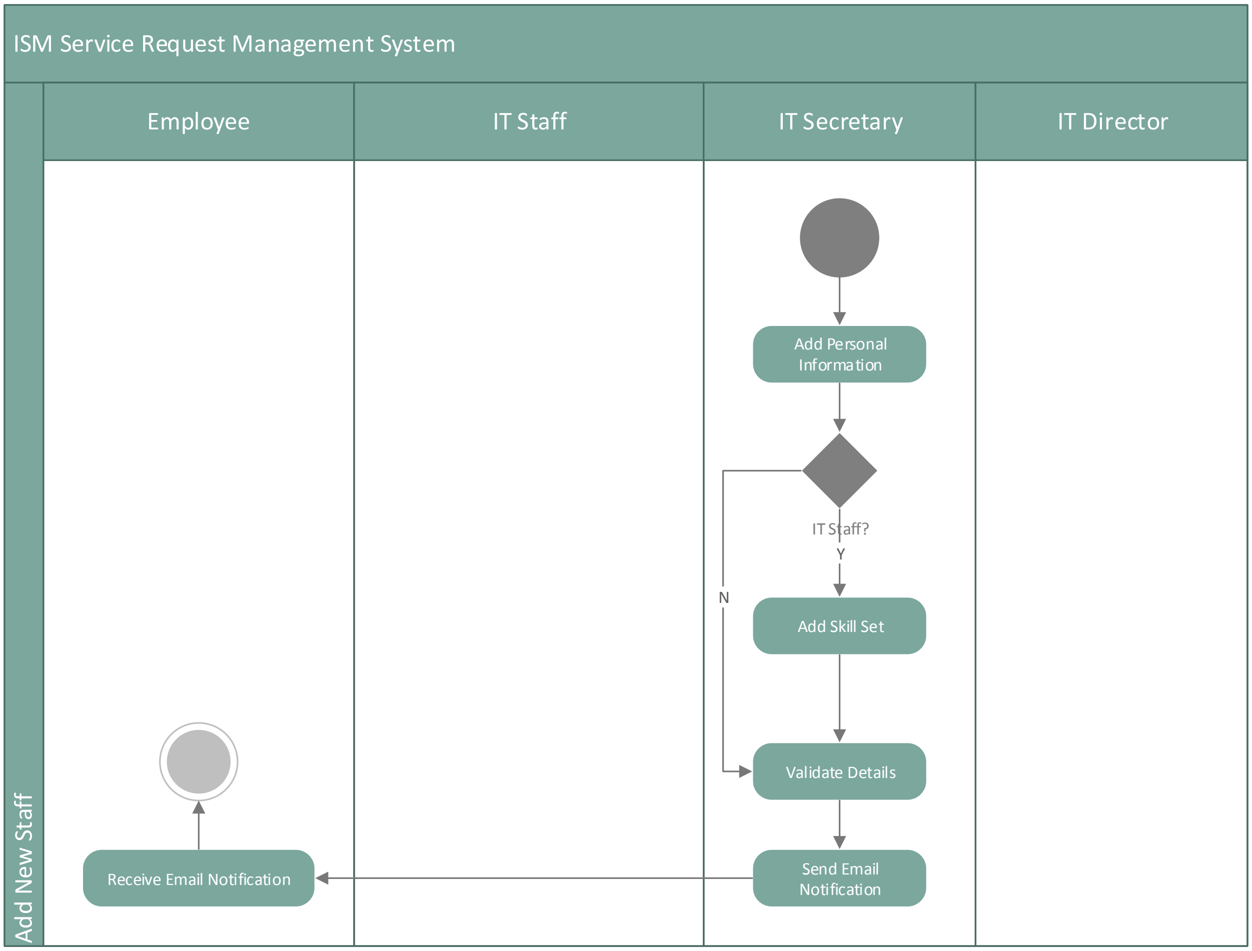

Figure 15: Activity Diagram- Add New Staff

To cater newly hired employees, the IT Secretary adds them to the system. However, it is important to classify if the new employees are IT Staff or not. Additional details need to be submitted if an employee is an IT Staff. These includes the Skill Set and schedule in order for these employees to received assigned service requests.

III. Operation and Testing Procedure

Data Gathering

One of the most important factors in building the system is gathering information. As a personal employee of International School Manila, the researcher has been very familiar with the overall operations of the IT Department. The researcher conducted an internet and library research and interview to further gather required information. The information gathered will be used to analyze and propose solutions to the problems being encountered with their current transaction processes. The proponents will also use the library to have a reference and guide.

Description of Activities

Interview

Interview is the verbal asking of questions in a formal manner, especially one arranged for the assessment and qualifications of the application. The researcher conducted an interview with the Helpdesk Officer Ms. Evelyn Mendoza, to gather information and identify the difficulties that she is currently experiencing when it comes to helpdesk transactions. The objectives identified on the earlier part of this documentation served as guidelines in asking questions during the interview. The result of the interview justified the current problems being experienced in the overall IT Service Request processes.

Questionnaire

The researcher also prepared questionnaires which were used to acquire additional information. These were given to the IT Staff for their prudent analysis of questions and answer. This is a different point of view with regard to the service request process as the entity receiving the requests.

Research

Research is an important fact finding technique, where it is thoroughly ordered through application. It can include reviewing journals, periodical and tools to obtain background information, technical materials and news about industry. The researcher used different books pertaining to the different theories related with the system to be developed.

Library and Internet Research

The researcher also gathered information in the library through research to find some reference and for the different theories adopted in the study as guides and basis for the documentation of the proposed system.

The researcher also considered searching through the Internet that gave valuable additional ideas in developing the proposed project.

Observation

Observation is an activity of a sapient or sentiment living being, which is senses and assimilates the knowledge of a phenomenon in its framework of previous knowledge and ideas. The researcher observed the usual transactions and obtain to identify the general problem with the current system. This technique was often used when validity of data collected through other methods was in question or when the complexity of certain aspects of the system prevents a clear explanation by the end-users. The observation results are used to back up the results of the interview to find any information that is missed out in the interview.

Deployment Plan

It is very important to consider the adequacy of operational and development test resources (including infrastructure and personnel), policies, and procedures to ensure appropriate testing of the proposed system both during development and before operational use.

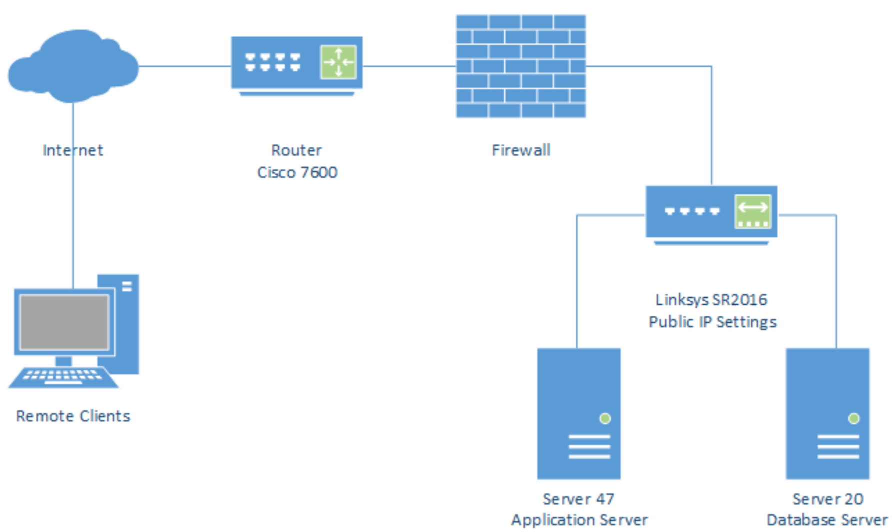

Figure 16: System Deployment and Implementation

The proposed system will be implemented in the current network environment/ infrastructure of International School Manila. Currently, there are numerous servers which handles different applications present at the school.

The database will be deployed in Server 20 as a database server along with other existing databases. However, the capacity was highly considered to ensure that the database of the proposed system will not have any capacity constraints on the said server.

The web application on the other hand will be deployed in the application server (Server 47). Such server is a newly purchased application server which is intended for the ISM Service Request Management System and other small systems or projects which are lined up to the development team of the organization.

The router where the servers are connected will be configured in such a way that the proposed system can be accessible to the desired environment and set of audience. It is by default, also connected to the firewall of the school for security purposes though system security will also be highly considered during the development of the system per se.

With the existence of the current technologies available at the International School Manila, it can be assured that the deployment of the proposed system into its actual working and operational environment will be successful.

IV. Evaluation Procedure

Following the successful implementation of the proposed system, it will be then evaluated using the ISO 90126 standards. Ultimately, it is composed of 5 primary measures with corresponding metrics. Enumerated below are the measures used in evaluating the proposed system. Selecting only those metrics which are applicable to the proposed system, the researcher ended up with the following:

Functionality

Table 6: Functionality Metrics for Suitability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Functional implementation completeness	How complete is the implementation according to requirement specifications?	Do functional tests (black box test) of the system according to the requirement specifications. Count the number of missing functions detected in evaluation and compare with the number of function described in the requirement specifications.	X = 1 - A / B A = Number of missing functions detected in evaluation B = Number of functions described in requirement specifications	0<=X<=1 The closer to 1.0 is the better.
Functional specification stability **(Volatility)**	How stable is the functional specification after entering operation?	Count the number of functions described in functional specifications that had to be changed after the system is put into operation and compare with the total number of functions described in the requirement specifications.	X = 1- A / B A= Number of functions changed after entering operation starting from entering operation B= Number of functions described in requirement specifications	0<=X<= 1 The closer to 1.0 is the better.

Table 7: Functionality Metrics for Accuracy

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Accuracy to expectation	Are differences between the actual and reasonable expected results acceptable?	Do input .vs. output test cases and compare the output to reasonable expected results. Count the number of cases encountered by the users with an unacceptable difference from reasonable expected results.	X=A / T A= Number of cases encountered by the users with a difference against to reasonable expected results beyond allowable T= Operation time	0<=X The closer to 0 is the better.
Computational Accuracy	How often do the end users encounter inaccurate results?	Record the number of inaccurate computations based on specifications.	X=A / T A= Number of inaccurate computations encountered by users T= Operation time	0<=X The closer to 0 is the better.
Precision	How often do the end users encounter results with inadequate precision?	Record the number of results with inadequate precision.	X=A / T A= Number of results encountered by the users with level of precision different from required T= Operation time	0<=X The closer to 0 is the better.

Table 8: Functionality Metrics for Interoperability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Data exchangeability **(Data format based)**	How correctly have the exchange interface functions for specified data transfer been implemented?	Test each downstream interface function output record format of the system according to the data fields' specifications. Count the number of data formats that are approved to be exchanged with other software or system during testing on data exchanges in comparing with the total number.	X= A / B A= Number of data formats which are approved to be exchanged successfully with other software or system during testing on data exchanges, B= Total number of data formats to be exchanged	0<=X<= 1 The closer to 1.0 is the better.

Table 9: Functionality Metrics for Security

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Access auditability	How complete is the audit trail concerning the user access to the system and data?	Evaluate the amount of accesses that the system recorded in the access history database.	X= A / B A= Number of "user accesses to the system and data" recorded in the access history database B= Number of " user accesses to the system and data" done during evaluation	0<=X<=1 The closer to 1.0 is the better.
Access controllability	How controllable is access to the system?	Count number of detected illegal operations with comparing to number of illegal operations as in the specification.	X= A / B A= Number of detected different types of illegal operations B= Number of types of illegal operations as in the specification	0<=X<=1 The closer to 1.0 is the better.

Portability

Table 10: Portability Metrics for Adaptability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Adaptability of data structures	Can user or maintainer easily adapt software to data sets in new environment?	Observe user's or maintainer's behaviour when user is trying to adapt software to operation environment.	X = A / B A = The number of data which are operable and but are not observed due to incomplete operations caused by adaptation limitations B= The number of data which are expected to be operable in the environment to which the software is adapted	0<=X<=1 The larger and closer to 1.0 is the better.
System software environmental adaptability (adaptability to OS, network software and co-operated application software)	Can user or maintainer easily adapt software to environment? Is software system capable enough to adapt itself to operation environment?	Observe user's or maintainer's behaviour when user is trying to adapt software to operation environment.	X= 1 - A / B A= Number of operational functions of which tasks were not completed or were not enough resulted to meet adequate level during combined operating testing with operating system software or concurrent application software B= Total number of functions which were tested	0<=X<=1 The larger is the better.

Table 11: Portability Metrics for Installability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Ease of installation	Can user or maintainer easily install software to operation environment?	Observe user's or maintainer's behaviour when user is trying to install software to operation environment	X = A / B A = Number of cases which a user succeeded to in changing the install operation for his/her convenience B = Total number of cases which a user attempted to change the install operation for his/her convenience	0<=X<= 1 The closer to 1.0 is the better.

Maintainability

Table 12: Maintainability Metrics for Analyzability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Audit trail capability	Can user identify specific operation which caused failure? Can maintainer easily find specific operation which caused failure?	Observe behavior of user or maintainer who is trying to resolve failures.	X= A / B A= Number of data actually recorded during operation B= Number of data planned to be recorded enough to monitor status of software during operation	0<=X The closer to 1.0 is the better.

Table 13: Maintainability Metrics for Stability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Change success ratio	Can user operate software system without failures after maintenance? Can maintainer easily mitigate failures caused by maintenance side effects?	Observe behaviour of user or maintainer who is operating software system after maintenance. Count failures which user or maintainer encountered during operating software before and after maintenance. Otherwise, investigate problem resolution report, operation report or maintenance report.	X= Na / Ta Y = { (Na / Ta) / (Nb / Tb) } Na = Number of cases which user encounters failures during operation after software was changed Nb = Number of cases which user encounters failures during operation before software is changed Ta = Operation time during specified observation period after software is changed Tb = Operation time during specified observation period before software is changed	0<=X,Y The smaller and closer to 0 is the better.

Efficiency

Table 14: Efficiency Metrics for Time Behavior

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Response time	What is the time taken to complete a specified task? How long does it take before the system response to a specified operation?	Start a specified task. Measure the time it takes for the sample to complete its operation. Keep a record of each attempt.	T = (time of gaining the result) - (time of command entry finished)	0 < T The sooner is the better.

Table 15: Efficiency Metrics for Resource Utilization

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
I/O related errors	How often does the user encounter problems in I/O device related operations?	Calibrate the test conditions. Emulate a condition whereby the system reaches a situation of maximum I/O load. Run the application and record number of errors due to I/O failure and warnings.	X = A / T A = number of warning messages or system failures T = User operating time during user observation	0 <= X The smaller is the better.

Reliability

Table 16: Reliability Metrics for Maturity

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Failure density against test cases	How many failures were detected during defined trial period?	Count the number of detected failures and performed test cases.	X= A1 / A2 A1 = number of detected failures A2 = number of performed test cases	0<=X It depends on stage of testing. At the later stages, smaller is better.
Failure resolution	How many failure conditions are resolved?	Count the number of failures that did not reoccur during defined trial period under the similar conditions. Maintain a problem resolution report describing status of all the failures.	X= A1 / A2 A1 = number of resolved failures A2 = total number of actually detected failures	0<=X<= 1 The closer to 1.0 is better as more failures are resolved.

Table 17: Reliability Metrics for Tolerance

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Failure avoidance	How many fault patterns were brought under control to avoid critical and serious failures?	Count the number of avoided fault patterns and compare it to the number of fault patterns to be considered	X=A / B A= Number of avoided critical and serious failure occurrences against test cases of fault pattern B= Number of executed test cases of fault pattern (almost causing failure) during testing	0<=X<= 1 The closer to 1.0 is better, as the user can more often avoid critical or serious failure.

Table 18: Reliability Metrics for Recoverability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Mean down time	What is the average time the system stays unavailable when a failure occurs before gradual start up?	Measure the down time each time the system is unavailable during a specified trial period and compute the mean time.	X= T / N T= Total down time N= Number of observed breakdowns The worst case or distribution of down time should be measured.	0<X The smaller is the better, system will be down for shorter time.
Mean recovery time	What is the average time the system takes to complete recovery from initial partial recovery?	Measure the full recovery times for each of the time the system was brought down during the specified trial period and compute the mean time.	X= Sum(T) / B T= Time to recovery downed software system at each opportunity N= Number of cases which observed software system entered into recovery	0<X The smaller is the better.

Usability

Table 19: Usability Metrics for Understandability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Completeness of description	What proportion of functions (or types of functions) is understood after reading the product description?	Conduct user test and interview user with questionnaires or observe user behaviour. Count the number of functions which are adequately understood and compare with the total number of functions in the product.	X = A / B A = Number of functions (or types of functions) understood B = Total number of functions (or types of functions)	0<=X<= 1 The closer to 1.0 is the better.
Function understand-ability	What proportion of the product functions will the user be able to understand correctly?	Conduct user test and interview user with questionnaires. Count the number of user interface functions where purposes are easily understood by the user and compare with the number of functions available for user.	X= A / B A= Number of interface functions whose purpose is correctly described by the user B= Number of functions available from the interface	0 <= X <= 1 The closer to 1.0, the better.
Understandable input and output	Can users understand what is required as input data and what is provided as output by software system?	Conduct user test and interview user with questionnaires or observe user behaviour. Count the number of input and output data items understood by the user and compare with the total number of them available for user.	X= A / B A= Number of input and output data items which user successfully understands B= Number of input and output data items available from the interface	0<=X<= 1 The closer to 1.0 is the better.

Table 20: Usability Metrics for Learnability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Ease of function learning	How long does the user take to learn to use a function?	Conduct user test and observe user behaviour.	T= Mean time taken to learn to use a function correctly	0<T The shorter is the better.
Effectiveness of the user documentation and/or help system	What proportion of tasks can be completed correctly after using the user documentation and/or help system?	Conduct user test and observe user behaviour. Count the number of tasks successfully completed after accessing online help and/or documentation and compare with the total number of tasks tested.	X= A / B A= Number of tasks successfully completed after accessing online help and/or documentation B = Total of number of tasks tested	0<=X<=1 The closer to 1.0 is the better.

Table 21: Usability Metrics for Operability

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Operational consistency in use	How consistent are the component of the user interface?	Observe the behaviour of the user and ask the opinion.	a) X = 1 - A / B A= Number of messages or functions which user found unacceptably inconsistent with the user's expectation B= Number of messages or functions	0<=X<=1 The closer to 1.0 is the better.
			b) Y = N / UOT N= Number of operations which user found unacceptably inconsistent with the user's expectation UOT= user operating time (during observation period)	0<=Y The smaller and closer to 0.0 is the better.

Table 22: Usability Metrics for Attractiveness

Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value
Attractive interaction	How attractive is the interface to the user?	Questionnaire to users	Questionnaire to assess the attractiveness of the interface to users, after experience of usage	Depend on its questionnaire scoring method.
Interface appearance customizability	What proportion of interface elements can be customized in appearance to the user's satisfaction?	Conduct user test and observe user behavior.	X= A / B A= Number of interface elements customized in appearance to user's satisfaction B= Number of interface elements that the user wishes to customize	0 <= X <= 1 The closer to 1.0 is the better.