***Abstract:***

*In our project we employ methods of machine learning from text retrieval to search and build a workflow to troubleshoot a problem with a network based on previous similar errors in company database and what had been done to solve them providing more efficient day-to-day and long term work environment.*

*In addition the program will analyses the errors on time and configurational scale and advice on threshold adaptation to prevent future network issues and false alarms.*

**Keywords***: machine learning, Natural Language Processing, question answering machine, knowledge based artificial intelligence*

***INTRODUCTION***

**Mellanox overview**

Mellanox Technologies is a leading supplier of end-to-end InfiniBand and

Ethernet interconnect solutions and services for servers and storage.

Mellanox offers a choice of fast interconnect products: adapters, switches, software and silicon that accelerate application runtime and maximize business results for a wide range of markets including high performance computing, enterprise data centers, Web 2.0, cloud, storage and financial services.

**Mellanox Call Center**

Mellanox Call Center provides permanent manned support services for Mellanox customers throughout the year (7/24/365) based on the appropriate support contract.

The Call Center is the customer’s point of contact (mostly for emergency cases) for initiating and advancing technical related and support administrative issues.

Call Center activities are designed to accomplish the following targets:

1. Provide systematic and efficient call handling services to Mellanox customers.
2. Maintain information flow according to developing situations.
3. React to cases according to predefined procedures.

**Mellanox Care – Proactive Care Platform:**

24/7 fabric management services provided by Mellanox networking experts

Mellanox Care services use a combination of advanced monitoring software with a 24/7 human expert service of Mellanox personnel.

Mellanox Care will identify – **Keep alive Events** that will alert and address hardware failures, non-optimal configuration, service degradation issues and more. Above and beyond keeping the fabric healthy, Mellanox Care identifies and addresses the more complex performance issues and bottleneck scenarios, which are impacting application performance.

The current status is as follows:

* All logs are examined manually by call center
* All work flows and troubleshooting are based on representative’s experience
* Existing, reoccurring and resolved errors are verified manually

The project objectives to day to day work are as follows:

* To analyze the Mellanox care Daily report - provide more work efficient environment.
* To timeline and analyze all previous site cases to identify reoccurring issue on port \ switch per customer.
* To synchronize and monitor any frequent alarms\ triggers to locate reoccurring issues with Mellanox products on all Mellanox care Sites
* To provide a day \ week statistics - threshold adaptation advise for each site.

The aim of this project is upon receiving a network error from a customer to analyze and build a multi-stage work flow to troubleshoot a certain network error using a machine learning algorithms (knowledge based AI).

Machine learning algorithm implementation:

1. Finding all correspondences with this error, retrieve comments in the correspondences between support engineers and customers from the company's database.
2. Identify helpful and unhelpful advices using Natural Language Processing (NLP) algorithms.
3. Parse the input by the customer and “score” the engineer’s reply.
4. Upon using the machine learning algorithms the project will suggest a multi stage work-flow based on high scoring answers for every stage and determining what high scoring directions should not be implemented, due to prior actions (fault isolation)

***Background and Related Work***

**Customer Resource Management system:**

Mellanox's online Customer Resource Management (CRM) system – SalesForce, provides a comprehensive online tool to manage all of the customer's support issues in one place.

1. **Complete case management** including reporting support issues and tracking their progress

2. A searchable **knowledge database** to find solutions, best practices and worthy information

3. Access to **documentation** and drivers/firmware/software downloads

4. Built-in RMA request and tracking system.

**Case management and workflow:**

Mellanox support engineers handle each customer request according to suitable procedure designed according to the following parameters:

* Case type: there are two types of cases -
  + **Support cases** are handled according to case priority.
  + **RMA requests** are handled by first line support accompanied by the relevant departments.
* Case priority:
  + **Priority 1-** Procedure for urgent\ fatal issues set to establish communication and urgency for customer’s issue. Meant to ensure that proper attention of Mellanox functions, SLA target is met and to raise the awareness of related account teams and other professional groups (OPS, PM, AE, OEM account team, etc.) when dealing with a fatal network or production down situation (showstopper).
  + **Priority 2-** Procedure for medium severity issues.
  + **Priority 3**- Procedure for low severity issues.

**Priority 2** and **Priority 3** cases are handled by Pre-defined layers support system, the first line support start handling this type of cases and make the first contact with the customer and according to urgency and the complexity of the issue the case can be escalated to higher level.

* Customer support coverage: According to Mellanox policy each customer should get the support he needed therefor five types of coverage were defined :
  + **No support contract** - Cases from account with no support contract should be handled by the system administrator only.
  + **Extended hardware warranty**- Support engineers provide the customer with troubleshooting steps to insure that customer’s hardware are working properly with the latest available software.
  + **Silver support coverage & Stratigic customers** – First line support handles this cases according to priority 2 and priority 3 procedures.
  + **Gold customer**- Cases from gold customers escalated automatically to second support tier, in order to insure proper attention of Mellanox functions.

**Unified Fabric Manager (UFM®) Software for Data Center Management:**

Mellanox’s Unified Fabric Manager (UFM®) is a powerful platform for managing scale-out computing environments. UFM enables data center operators to monitor, efficiently provision, and operate the modern data center fabric. UFM eliminates the complexity of fabric management, provides deep visibility into traffic and optimizes fabric performance.

UFM covers many areas, such as fabric monitoring, provisioning, device management, Subnet Management and more.

**Mellanox Daily Report:**

Mellanox Daily Report is a snapshot of the environment it is running on that collects all the relevant information on Mellanox products installed.

It contains the following reports:

* **UFM Health Report**
* **Fabric Health report**

The content of the reports is regarded as trade secret to intellectual property of Mellanox Technologies.

**Mellanox Care Policy Booklet:**

Mellanox Care Booklet list all possible events captured by Mellanox Care application

For each event you will find the following attributes:

* Description - General problem description
* Code – Internal parameter that is related to the case description
* Category (Device management, Fabric management, etc.) – This indicate which layer is effected by this error.
* Related items/Potential root cause – Potential hardware/software failure indicator.
* Description of error – Detailed problem that can be provided to the customer.
* First Line Support corrective actions – Troubleshooting steps to solve the issue.
* Support type/process (Standard/Fatal).

**Introduction to Machine learning - Case-Based Reasoning**

The ability to automatically suggest an answer for a question given a database off possible previous occurrences of the question given to others is a focus of many studies in many fields.

The work of Roger Schank2, is widely held to be the origin of CBR. He proposed a different view on model-based reasoning inspired by human reasoning and memory organization: If a “memory packets” contains a situation where some problem was successfully solved and the person finds himself in a similar situation, the previous experience is recollected and the person can try to follow the same steps in order to reach a solution. Thus, rather than following a general set of rules, reapplying previously successful solution schemes in a new but similar context solves the newly encountered problems.

In the problem solving algorithm of case based reasoning the following steps are taken:

* Describing the current problem.
* Searching for a similar previously solved problem.
* Retrieving the solution to it.
* Adapting the solution to the current problem.
* Verifying the solution.
* Storing the newly solved problem.

In turn, since the newly found solution may be used for solving future problems, the process denotes the CBR working cycle.

**Resource limitations:**

* Data from SalesForce cannot be stored locally as this information is classified.
* Loading and analyzing all case comments can take a long time – the purpose is to be as efficient as possible

***Machine learning***

Machine Learning is the wish to program computers so that they can “learn” from input available to them1. The input to a learning algorithm is training data from our company’s database in which stored all technical issues that was dealt over the years, and the correspondences are representing experience which can the program learn in order to advise to a novice support engineer.

To find the best way to do this we’ll have to be more explicit about what data our programs will access, how they parse it, how and when the learning process will be done and will it be evaluated3.

The most popular formal approaches to machine learning applicable to our project are as follows:

**Supervised learning**

Algorithms are trained using labeled examples, such as an input where the desired output is known. In our project, for example, The learning algorithm receives as an input the reply from the customer to this answer and the algorithm then learns how good it was based on customer’s next reply and modifies the model (score and stage) accordingly1,4,5. It will reach this phase if a solution was not found in the "Semi-Supervised Learning" phase. If the solution will not be found in this stage, it will move ahead to the "unsupervised Learning" phase.

A major sub section of the "Supervised Learning" phase is "Ranking" approach:

**Ranking**

Ranking is the problem of ordering a set of instances according to their “relevance.” A typical application is ordering results of a search engine according to their relevance to the query5.

"0-1" Ranking loss:

F(a,b) =1 is in case that a and b has exactly the same ranking, and F(a,b) = 0 otherwise.

"a" will be the requirement and "b" is the state.

Such a loss function is rarely used in practice as it does not distinguish between the case in which "a" and "b" are almost equal, but according to our specification of the system's requirement this method is the most suitable.

To avoid cases in which a response has double meanings, the algorithm will move to the "Unsupervised Learning" phase.

**Unsupervised learning**

It is used against data that has no historical labels. And they have no scoring points for a data set. The goal is to explore the data and find some structure within 4,5 . In our project it can identify customers with similar issues who can then be treated similarly or point out a large scale problem from many costumers, for example:

1. A certain SW or FW version is causing a typical error with many customers.
2. Certain topology configuration is causing typical errors.
3. Certain HW equipment is malfunctioning (maybe due to the mentioned above)
4. Certain errors appear in certain peak-times

Popular techniques include self-organizing maps, nearest-neighbor mapping, k-means clustering and singular value decomposition. These algorithms are also used to segment text topics, recommend items and identify data outliers2

In our project, the algorithm will reach the "unsupervised learning" phase if the solution is not found in the "supervised learning" phase.

**Semi-supervised learning**

It is used for the same applications as supervised learning. But it uses both labeled and unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data .This type of learning can be used with methods such as classification, regression and prediction4,5.

In our project, the semi-supervised approach will be used by the initial phase of the problem solution by looking for the error in the "Mellanox Care Booklet" which is a pre-defined file with common errors and their initial troubleshooting steps. If a solution is not found, the algorithm will go ahead with the "supervised learning" phase.

**Reinforcement learning**

It is often used for robotics, gaming and navigation. With reinforcement learning, the algorithm discovers through trial and error which actions yield the greatest reward. It is known as a “reward hunting” because each time an agent performs an action in its environment, a trainer may provide a reward or penalty to indicate the desirability of the resulting state. The goal of the agent is to learn an action policy that maximizes the total reward it will receive from any starting state1,4,5. In our project, the goal, or “reward” is to find solution to the problem the costumer is facing in the shortest amount of replies, minimizing the time effort and cost by the customer. The faster the solution – the bigger the reward. If the decision making process reaches the goal much faster than the fastest solution so far, the score of this solutions is modified, it means a faster solution gets a bigger score.

**Decision Trees**

It is one of the most widely used and practical methods for inductive inference. It is a method for approximation of discrete-valued functions, in which a tree represents the learned function. Each node in the decision tree specifies a test of some attribute of the query instance, and each branch descending from that node corresponds to one of the possible values for this attribute 3,5 . In our project, a positive or negative answer by the customer to any of the engineer’s questions in the previous stage (that was preferably generated by this app as well to follow this tree) will eliminate many other troubleshooting steps and may focus on other, more specific, test to find the root cause of the error.

**Formal definition of AI Implantation**

**Sensors:**

The support engineer.

**Actuators:**

Suggest a multi stage work-flow based on high scoring answers for every stage and determining what high scoring directions should not be implemented, due to prior actions (fault isolation).

**Percept**:

Booklet, comments from cases, case resolution, snapshot, switch system logs

**Procedure - Algorithm and flow of events**

**Artificial intelligence Error handling:**

1. The support engineers attaches the case logs or Daily report logs into the Data-Analyzer.
2. The Data-Analyzer compares the source and failure description with previous open Mellanox care cases.
3. If the error does not appear in the opened case or in known issues, then the issue needs to be addressed.
4. The Data-Analyzer classifies the error type according to the booklet: software, hardware, and configuration. The class can be combination of types (6 types in total)
5. If error cannot be found in the booklet it will be classified as a general error (can be the 3 types) therefore the solution for all types will be checked (Unsupervised ML).
6. The Data-Analyzer receives the relevant data from the case and determines the case priority: P1 for fatal. P2 for medium. P3 for low priority.
7. The Data-Analyzer describes the error in detail for the support engineer: class, potential root cause, more detailed description, overall actions to resolve issue. (If error is found)
8. The Data-Analyzer searches within the category’s solutions bank for the highest ranking solution
9. The support engineer replies this comment and waits for customer’s feedback.
10. Upon receiving customers feedback the NLP algorithm identifies it as “positive” feedback or “negative” (alternative: the comment is being marked manually by the support engineer as such)
11. If “negative” – deducts 1 point from the solutions rating
12. Then searches for the second heights solution.
13. If solution is feed backed as “negative” gain, the process repeat itself till a solution is found or solution bank is empty.
14. A solution can be skipped by the support engineer (go to the next in line).
15. The error type decision branch can be changed.
16. If solution bank is empty, the appropriate message will be sent and the issue will be escalated.
17. If customer’s feedback is “positive” one point is added to the solutions rating.
18. If customers reply is the latest before closing the case and is lower in it’s number in the correspondence is lower than another point is being added to the rating. (if a solution is done with fewer steps)
19. A solution’s rank can be manually changed due to new\obsolete procedures, equipment, management’s decision etc.??? problematic
20. If unclassified solution is regularly being resolved by a certain type solution a message to the engineer will advise to manually classify this error.

**Artificial intelligence Statistics:**

1. On request, the Data-Analyzer provides statistical data on monthly, weekly and daily scale on case opened and alerts on unusual grouping of errors.
2. The support engineers attaches the problem description into the Data-Analyzer.
3. The Data-Analyzer gathers: timestamp, device name, source, failure description for statistical calculations.
4. The Data-Analyzer searches in SF and shows how many cases recently opened for this error, this source, and this company.
5. The Data-Analyzer describes the error in detail for the support engineer: class, potential root cause, more detailed description, overall actions to resolve issue. (If error is found)
6. If unusual grouping of errors is found, the Data-Analyzer calculates the threshold breaches on the groups (threshold adaptation advisory).

Notes:

The solutions bank will be built in a tree shaped DB. Which consist of 6 branches for all combinations of errors classes: HW, SW, configuration, HW+SW, HW+ configuration, SW+ configuration, HW+SW+ configuration (general or uncategorized). To provide better fault isolation no hop between branches will be allowed (except for override) – the next best solution must be from this decision tree and not a high ranking solution for an unrelated error.

**Determining a “positive” or “negative” response**

Determining a “positive” or “negative” response by the customer has a vital role in scoring a solution. Positive scoring means the given solution is helpful, thus it should be considered higher in the order to be reused in further cases with this kind of error. The other way around for the negative response from the customer.

Currently we are considering 2 ways to implement the classification of the nature of the response: manual and NLP based.

**Manual classification:**

Upon receiving a reply from the customer replying to the solution proposed by the support representative, the support representative will accordingly mark the proposed solution leading to this reply as positive or negative based on the reply.

The marking will be done via two checkboxes: one for positive and one for negative.

A solution can stay in a “natural” state when the classification is not determined.

**Natural Language Processing (NLP)**

The idea of computers being able to understand ordinary languages human is being a focal point of scientific studies for many years7, in our project we deal with customers from all around the world so when taking in consideration that people from around the world communicate in various ways, and have different ways of expressing themselves and not only do individuals but also people from other countries tend to make choices may be more characteristic. When add to that is the challenge of “English as a second language”, we will fully appreciate the challenges posed by understanding of natural language and even achievements from focusing on a range of specific fields which will be determining what is a positive or a negative response by a customer.

**Tokenised text and pattern matching**

One of the more basic operations that can be applied to a text is tokenising: breaking up a stream of characters into words, punctuation marks, numbers and other discrete items 6,7 a fair amount of information may be obtained from relatively shallow analysis of tokenised text for example negative words like “no” didn’t help” may reveal that the reply by the engineer wasn’t helpful while “expressions like “issue resolved” “thanks for the help” may indicate the it was helpful. In addition, we need to pay attention to some aspects that may reveal that the customer is not satisfied such as the text is written with all capital letters or there are numerous or more than two consecutive exclamation marks. The emotional aspects of the language may vary from language to language.

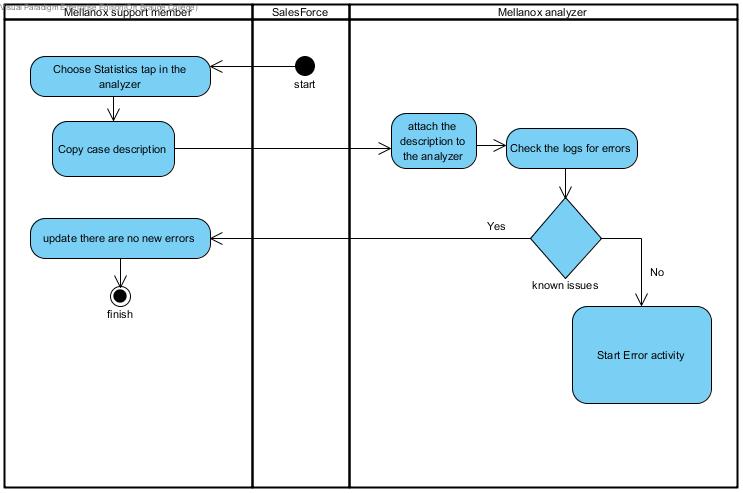
**Parts of speech and structure**

A further stage in analyzing text is to associate every token with a grammatical category or part of speech (POS) 6,7. In our project we will have to analyses the grammar of the sentence to determine on the type of the response, such as: “we solved the issue”, “the issue was solved” “your solution was helpful” etc.

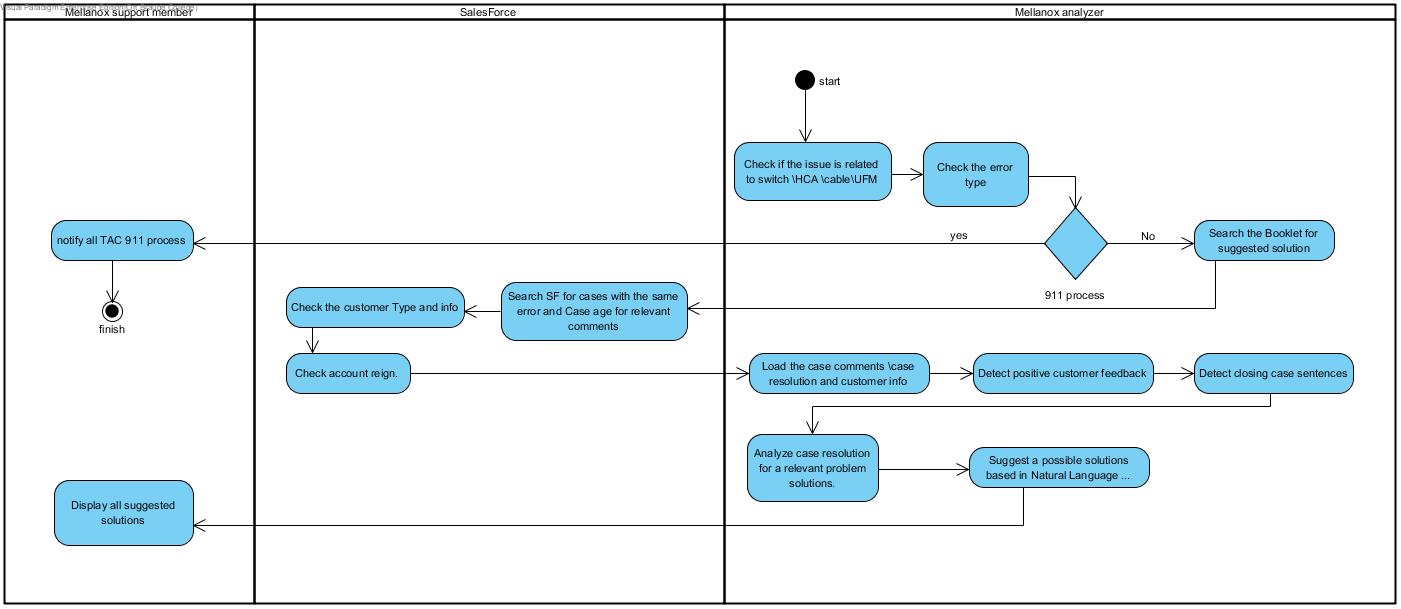
***Design (GUI, UML diagrams)***

***Activity Diagrams:***

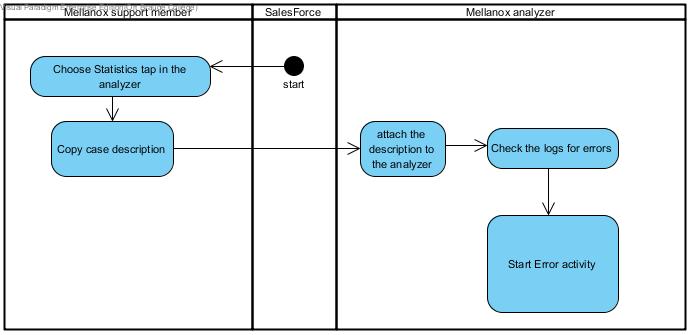
**Activity- Mellanox Care Daily report**

****

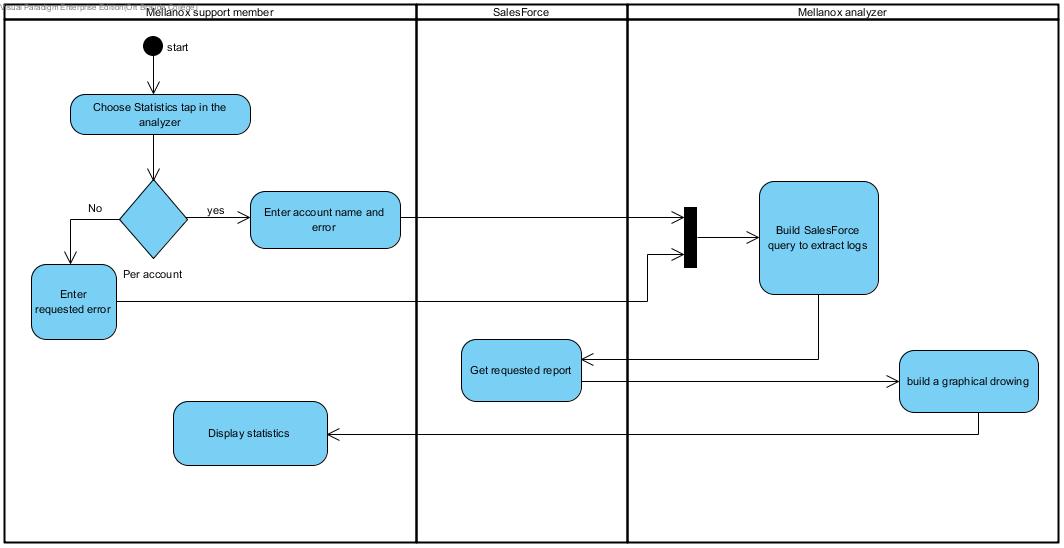
**Activity - Handle Error**

****

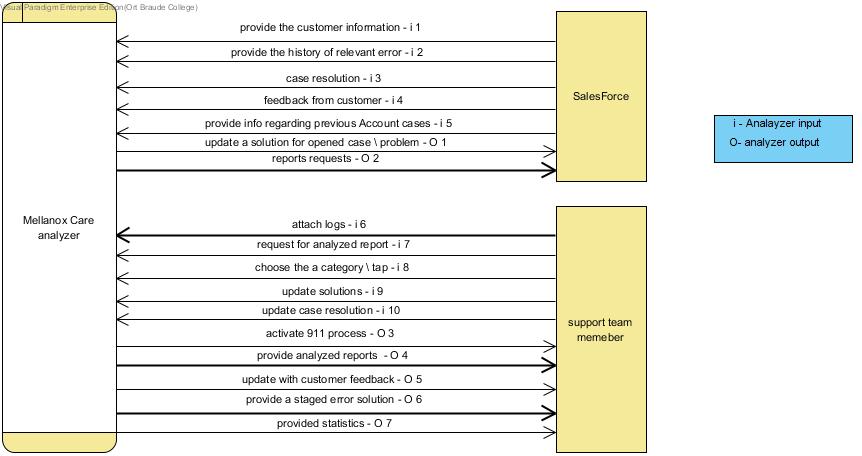
**Activity - Mellanox Care Keep alive event:**



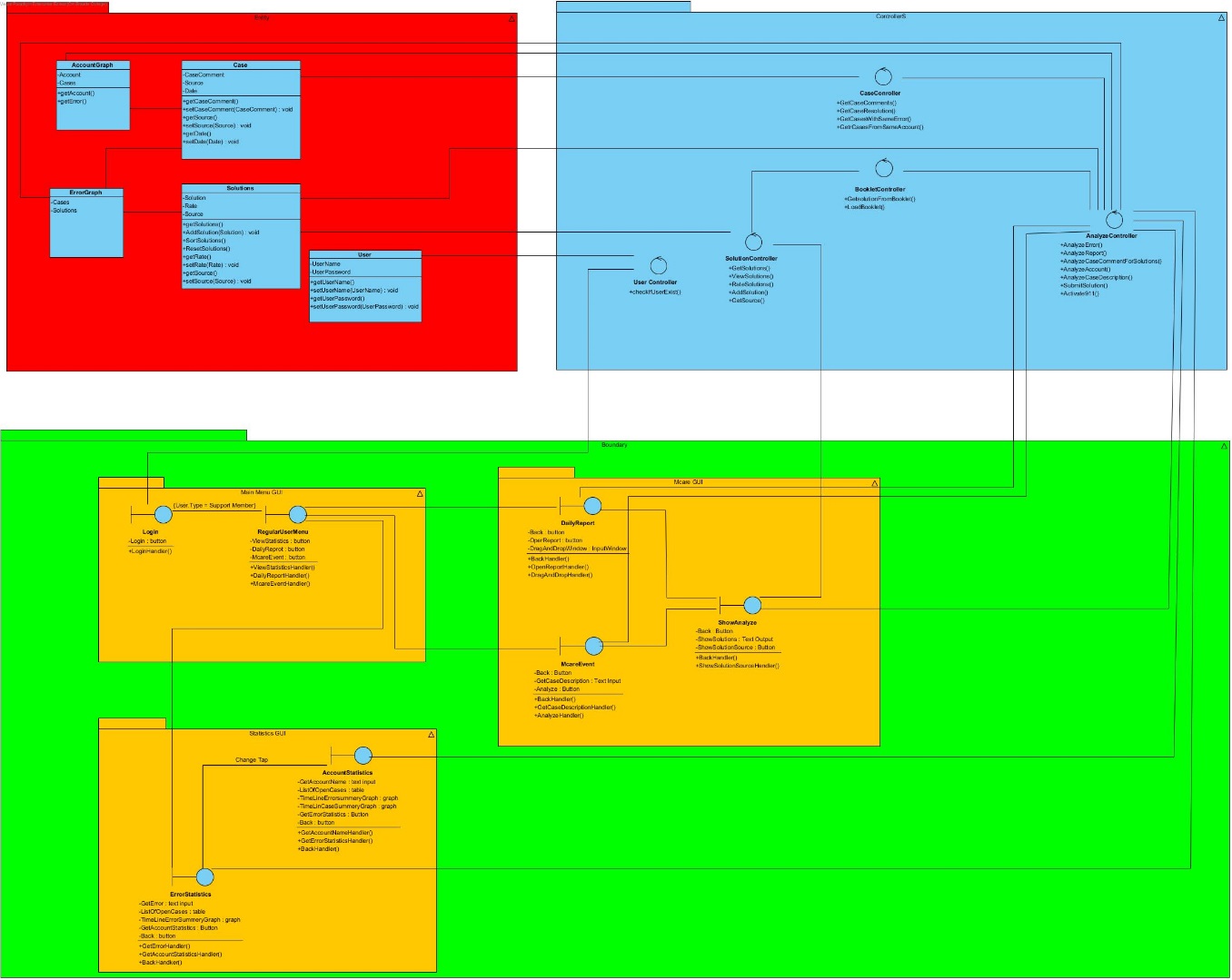
**Activity- Show statistics**

****

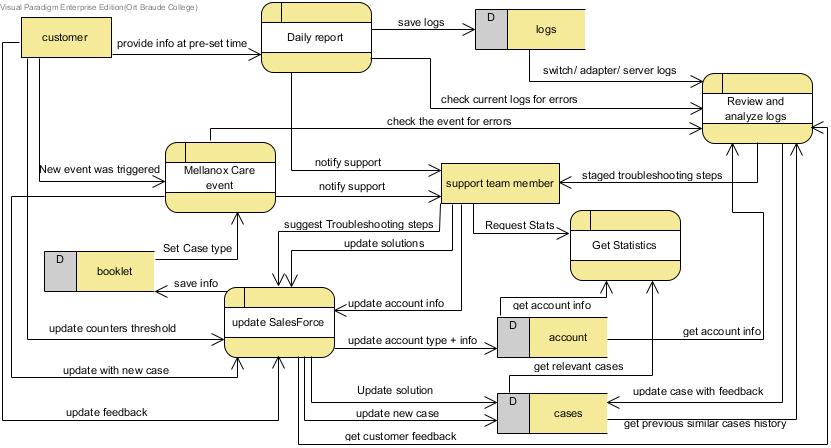
**Connection Diagram:**

****

**Case Diagram:**

****

**Data Flow Diagram:**

****

**Testing plan**

We will implement our approach in the context of the alarm triage problem submitted by a customer. Our simulation experiments with data from many previous cases in which the troubleshooting steps by the engineers are different and the responses by the customer are different. We will add new engineers replies and new customer responses to show the potential of our constantly-updating machine-learning-based approach for improving the error triage and troubleshooting process.

**Bibliography**

1. *Introduction to Machine Learning & Case-Based Reasoning - by Maja Pantic*
2. *R.C. Schank, Dynamic memory: A theory of reminding and learning in computers and*

*people. Cambridge, UK: Cambridge University Press, 1982.*

1. *T.M.. Mitchell, Machine Learning. Singapore: McGraw-Hill Companies Inc., 1997.*
2. *Machine Learning - What it is & why it matters -* *SAS®*
3. *Understanding Machine Learning: From Theory to Algorithms - by Shai Shalev-Shwartz and Shai Ben-David*
4. *Natural Language Processing (Almost) from Scratch - Ronan Collobert - by*

*Jason Weston, L´ eon Bottou, Michael Karlen MICHAEL, Koray Kavukcuoglu, Pavel Kuksa*

1. *Introduction to natural language processing - by R. Kibble*
2. *Human-Guided Machine Learning for Fast and Accurate Network Alarm Triage - by Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, Blaine Christian*
3. *Automating Datacenter Operations Using Machine Learning by Peter Bod´ık*
4. *Application of Machine Learning in Fault Diagnostics of Mechanical Systems Massieh Najafi, David M. Auslander, Peter L. Bartlett, Philip Haves*