**UNIVERSITY OF WATERLOO**

Faculty of Mathematics

**[Insert Title Here]**

Bank of America Merrill Lynch

Toronto, ON

Prepared by

Kushan Zaveri

1B Computer Science

ID 20721646

Month Day, Year

**\*\* Letter of Submittal Not Included**

**\*\* Table of Contents and List of Figures to be included**

Your report should have a Table of Contents that lists the report’s sections, subsections, and page numbers.

If you include figures in the body of your report, you must also include a List of Figures, indicating titles and page numbers. Figures should be numbered, titled, and mentioned in the text preceding them. Large figures may be included in an appendix.

Use tab leaders to guide the reader’s eye.

**Executive Summary**

* the purpose of the report
* the key points of the analysis
* the highlights of the conclusions
* the highlights of the recommendations

1. **INTRODUCTION**

At Bank of America Merrill Lynch, the S1 Swaps team is developing an application that is used to book trades and create swaps, two types of financial transactions that the company is responsible for conducting for their customers. There are nine developers at the Toronto Global Business and Marketing Technology (GBAM) office who work on the S1 application. The project manager of the Toronto GBAM team, Michel Chicoine, is responsible for conducting team activities and ensures that they produce software at the highest quality. A current issue is that it is difficult to find sources of error and track the quality of the team’s output. He believes it is necessary to assess metrics relating to the performance of the S1 application in order to diagnose software issues on time and react to them promptly.

A member of the team is required to develop this monitoring tool that allows developers to publish data relevant to tracking the performance of the application. This information would ultimately be presented in an accessible and clear manner to assist the S1 team.

This report outlines the importance of tracking metrics and potential beneficiaries of this information. It then explains the technology used to develop the monitoring tool, the various statistics that the team may assess, and the benefits it will provide team members.

1. **ANALYSIS**

In the Information Technology field, it is important that many aspects of software development are monitored. This may range from the productivity of the team to the quality and performance of the end product. The specific data that is monitored will vary depending on the requirements outlined by the project manager.

Due to the high failure rates in IT projects, specifically those dealing with software development, it is vital for the team to track key metrics (Marsanu, 2010). As observed at Merrill Lynch in the S1 application team and various other software development workplaces, bugs and errors come up quite regularly. By measuring certain attributes, it becomes significantly easier to track processes and thus identify and resolve problems in a timely manner.

**How the Monitoring Tool was developed:**

Whenever a task is performed in the S1 application, the software will output a message containing information relating to that task. The message will be in a format called “JSON”, a standard format used to transport data. Here is a sample JSON object:



**Value**

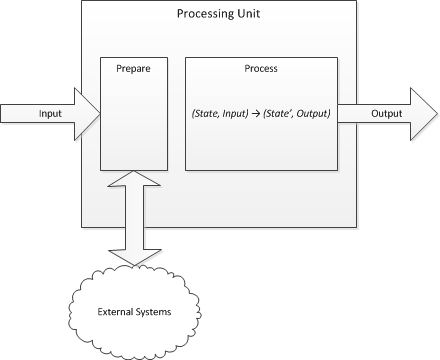
**Attribute**

As shown in the image above, a JSON message has attribute-value pairs. For example, “environment” is the attribute and “QA” is the corresponding value. This specific message would be sent by the application to indicate how much free memory the application has in the “AMRS” region at a given time. Expand on what the fields mean

These JSON objects are sent to a specific “topic” on a messaging client called AMPS, a third-party software used to send and receive data over a server. The below image is an example of a stream of JSON objects coming through an AMPS topic.

[insert diagram of JSON messages going to a topic]

Once the raw JSON messages are on an AMPS topic, the data needs to be processed to create useful metrics. In order to handle the input, processing, and output of these metrics, a processing unit (PU) model is used. This model was created by the developers of the S1 team. The following diagram describes the basic structure of a PU:



At its core, a PU accepts data in the JSON format, prepares it for use, performs the processing required to obtain useful information, and will finally output the newly created data. This approach allows the monitoring tool to break down PUs to handle smaller tasks. In the end, these PUs can be daisy-chained, passing JSON messages from one to another.

**Preparing the data:**

Before the code can interact with a JSON message, it must be “de-serialized”. As described in “Beginning JSON”, De-serializing is the process of taking a message from a certain point in time and converting it in to a data structure (CITATION NEEDED). Since Java is the primary language used by the S1 team, a Java de-serializer was needed. Hence, a tool called GSON was used to convert JSON to Java data structures. Once a Java object is created, code can be used to perform the processing.

[diagram]

The above image shows the corresponding Java data structure to a JSON object.  **Processing the data:**

Once the JSON message has been deserialized, all the processing can be done on the newly created Java object. As shown in the (use figure name) diagram above, the processing stage takes in a “State” and an “Input” and will produce a modified State (State’) and an “Output”. The Input is the Java object created in the preparation step. The State is used to store information throughout the processing stage.

Consider this example of how processing is performed:

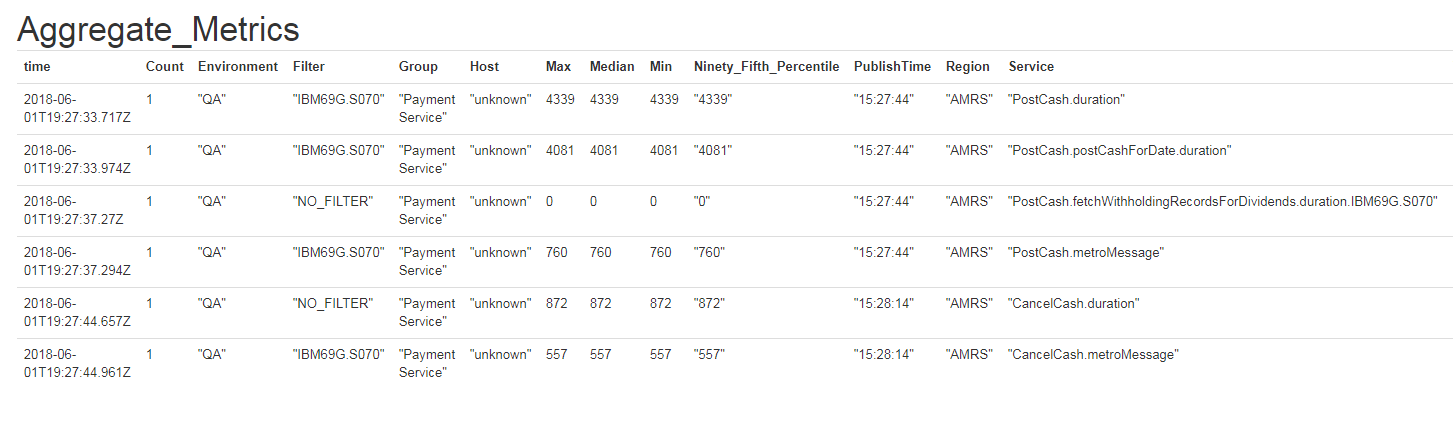
One type of metric that the S1 team requires is a sum of all the system memory used in a given minute. Since JSON messages can only come in one at a time, a State is used to store the sum so far. [MAKE THIS DIAGRAM]

Receive a JSON message with

When the processing is complete, the final Java object is the “Output” containing useful metrics.

**Outputting data:**

Once a Java object is created, there are many options for where it can go. In the final step of the monitoring tool, the Java object will typically be published to a database. The S1 team uses InfluxDB, a time-series database, to store the information. A time-series database is one that organizes data points by the time they occurred. Since InfluxDB provides support for the Java programming language, Java objects can be published directly to the database. Furthermore, InfluxDB support is provided by Grafana, a graphing tool which will be described later. The following image shows some data that is published to the InfluxDB database:



**Visualizing data:**

The previous section describes how data is monitored, processed, and stored, however it needs to be made accessible to the project manager and team members. In order to present the data, a tool called Grafana is used to generate graphs and displays to provide an overview of the information in the database. The data is taken from the database by using SQL queries which specify which data points should be graphed.

[ sample sql query]

Since InfluxDB is a time-series database, each data point has a corresponding time. This allows Grafana uses time as the independent variable for all of the graphs. This is a sample graph that tracks ….. :

In order to organize the metrics and provide multiple views for different users, Grafana provides the ability to create dashboards, which contain graphs related to a certain topic. For instance, one dashboard might contain information related to memory usage and another containing processing times for various processes. Another feature in Grafana is the ability to filter and narrow down data by a certain time/region/environment/etc. This provides more functionality to the users of the dashboards.

The end result is a set of dashboards that provide developers and the project manager with a general overview of various metrics from the S1 application.

**Important Metrics required by the S1 Team:**

There are many different metrics that the S1 development team and project manager would like to monitor. The first general category is data related to the Payment Service application within S1.

The Payment Service is responsible for handling cash transactions and it determines when cash is paid out or to be received from a certain customer. So, the first set of metrics would be related to how long it takes for the Payment Service to perform these cash transactions. By tracking this data, the monitoring tool allows developers to track the real-time processes performed by the application and also identify any anomalies. Next, just like any other process running on the S1 application, the Payment Service takes up system memory. Since memory is a very valuable resource to the company, it would be important for developers to know when too much memory is being used. This graph below is a simple view of how much free memory is available, and a colour code to quickly convey information:

[insert image of that graph]

Similar to the Payment Service, there are many other small applications within S1 which are planned to be tracked in the future.

The other category of metrics that the monitoring tool currently handles is data related to employee progress. An example of this is a current task in which the developers are responsible for moving to a new transaction handling system. Since there are a variety of transaction types that the S1 Application handles, the developers are migrating the different types of transactions one by one. In order to track the progress of the employees, the monitoring tool tracks how many transactions are being passed through the old code, and how many are being passed through the new code. This is done by checking the “Service” field of incoming JSON messages and counting the number of messages corresponding to the old and new Service.

With the ability to measure the developer’s performance, it allows the project manager to see how employees are working and assists them in delegating tasks.

**How the monitoring tool affects specific jobs:**

There are many people that can benefit from tracking and monitoring data related to the S1 application. Internally, the project manager can use software metrics to find the root of an issue promptly after it occurs. This information can then assist them in assigning tasks or providing stakeholders with accurate explanations of why a failure might have occurred. Finally, it provides an objective way of tracking the team’s progress and performance, which is useful to both the manager and the developers (Jethani, 2012).

Apart from the management of the team, the developers in the S1 team have the responsibility of creating software of the highest quality possible. By exposing statistics that relate to application performance, developers will be able to quickly assess the overall status of the S1 software to see that it is performing well. Furthermore, the system will provide an accessible way to see the impact of a developer’s changes on the overall system. For instance, if a developer were to modify the application’s process of opening a new transaction, they could see if it has an effect on the process run times (how long it takes to execute). In turn, the S1 team will be able to make smarter decisions when it comes to developing software.

As a result of the improved management and operation of the development team, the users of the application will notice enhancements in performance and fewer bugs. It is also in Merrill Lynch’s best interest to improve the quality of their products and the productivity of their employees, and measuring key attributes are one way to reach this goal [provide a citation].

<https://link-springer-com.proxy.lib.uwaterloo.ca/article/10.1007/s13198-012-0101-1?institute=OCUL.waterloo>

Pocatilu, P. (2007). IT Project Management Metrics. *Revisita Informatica*

*Economica*, 4 (44), 122-123.

Marsanu, R. (2010). Project Management Metrics. *Oeconomics of Knowledge*, 2 (2),

10.

## [Beginning JSON](https://proxy.lib.uwaterloo.ca/login?url=http://books.scholarsportal.info.proxy.lib.uwaterloo.ca/viewdoc.html?id=/ebooks/ebooks3/springer/2015-05-26/1/9781484202029)

### Smith, Ben, author.

### Berkeley, CA : Apress 2015

3.7 Analysis

Your work report must include significant analytical content.

A description of the steps in a process is not sufficient. The following list provides examples of acceptable analytical content:

* a discussion of cause and effect
* a discussion of advantages and disadvantages
* a comparison of two or more systems or products

The following are examples of acceptable analyses:

* Why does a problem exist?
* **How does the problem affect specific jobs in the workplace?**
* **How does the new system or product solve a problem?**
* **What aspects of the problem have been improved? How?**
* What problems does the system or product not solve? Why not?
* How can the system or product be improved?

3.8 Conclusions

The Conclusions section should be brief and should contain no new information. Conclusions should not make direct reference to sources, figures, or tables.

Each conclusion should follow logically from the facts and arguments presented in the Analysis section.

3.9 Recommendations

This section is optional because recommendations are not appropriate for all reports.

Recommendations are essentially speculative but should be brief and should follow logically from the conclusions. Include comments derived from experience that may improve future activities of the company.