From: ***<Team 1, Member 1>***
Sent: Friday, January 4, 2019 02:18 PM
To: ***Fayyazi, Morteza <…@mentor.com>***
Subject: Re: FW: ***<Feature 1>*** Library document

Hi ***Morteza***,

Here are few APIs that I think are going to be helpful:

1. bool ***<Retrieve Objects API>***(***<Feature 1, Class 1>***&); // retrieves all UIDs starting at ***<Feature 1, Object 1>***
2. bool ***<Retrieve Objects API>*** (***<Feature 1, Class 1>***&, Uint32); // Retrieve all UIDs up to certain ***<Feature 1, Criterion 1>***, starting at ***<Feature 1, Object 1>***
3. ***<Feature 1, Class 1 Iterator>*** Find***<Name Specifier>***(***<Feature 1, Class 1>***&, ***<Feature 1, Class 1 Iterator>***, ***<Feature 1, Class 1 Iterator>***, Pattern, Compare) // using provided binary predicate comp functor, or
<Feature 1, Class 1 Iterator> Find***<Name Specifier>*** (***<Feature 1, Class 1>***&, ***<Feature 1, Class 1 Iterator>***, ***<Feature 1, Class 1 Iterator>***, Compare) // using provided unary predicate comp functor,

It would be helpful to add to the ***<Feature 1, Class 1>*** the following attributes:
<Feature 1, Attribute 1>, ***<Feature 1, Attribute 2>***, ***<Feature 1, Attribute 3>***

Did we consider subclassing instead of having ***<Feature 1, Class 1>*** and ***<Feature 1, Class 2>***? However, this could be considered implementation detail level.

Cheers,
<Team 1, Member 1>

On Thu, 2019-01-03 at 13:21 -0500, ***Fayyazi, Morteza <…@mentor.com>*** wrote:
<Team 1, Member 1>,

The attached is the new API for ***<Feature 1>*** API.

Regards,
- ***Morteza***

From: ***<Team 1, Member 2>***
Sent: Wednesday, January 2, 2019 10:20 PM
To: ***<Team 2, Level 1 Manager>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** Library document

Hi ***<Team 2, Level 1 Manager>***,

Thanks for the ***<Feature 1>*** Library API doc, below are my listed comments. If you have additional questions regarding my feedback, we can have an offline conversation as necessary.

With regards,

<Team 1, Member 2>

1. I believe the API can be simplified and made more intuitive for the end user. Subsequent items describe details.

2. When providing library functionality, one should be careful about not making any impositions on a client's application. ***<Redacted text 1>***

3. I don't think that the library should enforce or assume singleton behavior for class ***<Redacted text 2>***

4. I don't think there should be explicit Open () and Close () APIs, ***<Redacted text 3>***

6. The ***<Redacted text 4>*** to derive their respective.

7. Since the API is a read API, ***<Redacted text 5>***

Here we clearly see that ***<Team 1, Member 2>*** did what ***<Team 1, Member 1>*** did. In fact, it was suggested by ***<Team 1, Member 2>*** to draft an email with how modification to API would look like. ***Mr. Fayyazi, Morteza*** discretionary consider the very same act as violation when it was conducted by ***<Team 1, Member 1>***!!!!

8. Contemporary C++ library APIs utilize iterators to deploy generic interfaces to their clients. I have attached a prototype interface that demonstrates what this implies. Please feel free to use this as an example in finalizing your proposal.

On 12/12/18 11:41 AM, ***<Team 2, Level 1 Manager>*** wrote:

Hi All,

Here is the current draft of the ***<Feature 1>*** Library document. Please, review. Let me know if you have any questions, comments or corrections.

Regards,

-- ***<Team 2, Level 1 Manager>***