**TerByte Design Layout**

**Team Plasma**

**i) Provide an outline of the page components**

**(a) Overarching comment section that describes the request flow of the user experience**

The user will enter the TerByte web page. If they want to create a new bug, they may click the “Add Bug” button in the header section of the page. This button is bright green and noticeable. Once they click the button, they will look at the div on the right side of the screen labeled Bug Details. They may now add an Author, title, description, status, and priority of the bug. Once they finished filling in the bug fields, they may click the “Submit Bug button located in this div. If they want to make changes to any given bug, they may select it from the view on the left hand side of the screen below the TerByte header by clicking on it. The contents of the bug will be displayed in the view that they created the bug in. By default, they can only view the contents of the bug. If they want to edit the fields, they may click the red “Edit Bug” button in this view. The fields will become white and editable. Once the user is satisfied with their changes, they may click the green “Update Bug” button. Once the button has been pressed, the user will see that the other views that reference this bug will be automatically updated. Maybe the user wants to add a comment for a bug. Once they select a bug, they can look below the bug details view to add comments. The user can add the author of the comment and the description. Once the comment is done, they may add the comment to the bug by pressing the “Post Comment” button. The user may view any comment for a bug by selecting them from the bug list. Statistics for all of the bugs are displayed above the bug list. This is a pie chart. By default, the pie chart displays data for bug priority. They may change the target data by selecting “Status” from the drop down menu below the pie chart. They can hover over the chart to view statistics for each target.

**(b) A simple page layout wire/line diagram describing the main containers and the fragments that are displayed in each container**

Refer to TerByte Wireframes.pdf

**(c) Mockup variations of colors, grid layouts, fonts, backgrounds, borders**

Refer to TerByte Wireframes.pdf

**(d) A list of page fragments and templates**

Bug-detail.jade, buglist.jade, comments.jade, graph.jade, terbyte.jade, layout.jade

**(e) A description of the feature functionality for each fragment**

Bug-detail.jade: Div 3 in TerByte. This view shows the detail of a bug. By default this view will be empty unless the user clicks the “Add Bug” button at the top of the page or clicks a bug in the buglist view. If the user clicks the Add Bug button, a new bug will be created in this view. The user may enter the Author, title, description, status, and priority of the bug. Once these fields have been submitted, the bug will be added to the buglist view. If the user clicks an existing bug in the buglist view, the contents will be displayed in this view. By default, they cannot edit the contents of the bug. If they want to edit the bug, they can click an Edit Bug button. The fields will become editable. Once the user is satisfied with their changes, they can click the Update Bug button and the database entry will be updated. All other views referencing this bug will change accordingly. The color key for status is as follows: Assigned – green, Completed – gray, new – blue.

Buglist.jade: Div 2 in Terbyte. Lists the bug name, the status, and its priority. The priority is displayed as a small circle to the left of the bug name with the color key: red for high, orange for medium, and yellow for low. The user may click on the bug to modify its fields. The content for the bug will display in other views such as bug-detail.jade and comments.jade.

Comments.jade: Div 4 in Terbyte. This view is displayed below the bug-details view. Once the user selects a bug from the buglist view, its contents will be displayed in this view. The user may enter the author of a comment and its description. Once the comment has been completed, they may submit the comment for the corresponding bug by clicking on a submit button located in this form.

Graph.jade: Div 1 in TerByte. This view displays a pie chart for the bug database. The user may select the chart content from a combo box in this view. They may select between Priority and Status. When they select the type, the chart updates automatically.

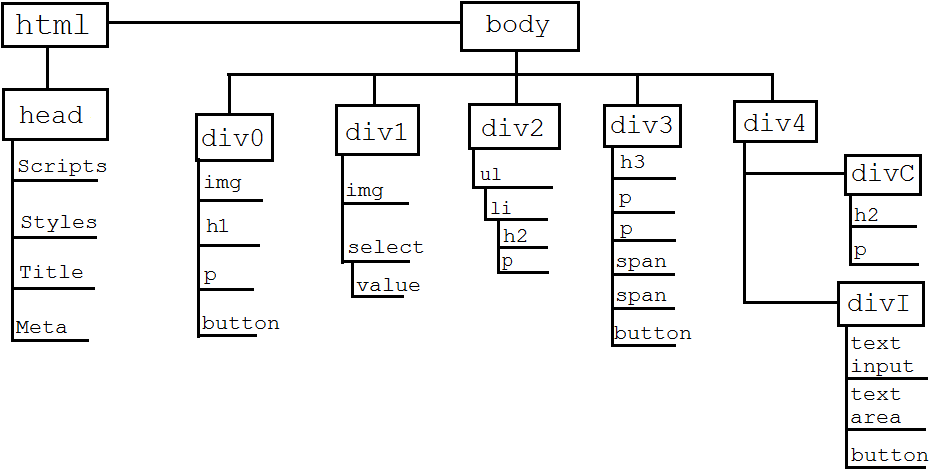
Terbyte.jade: The main view for TerByte. This view contains the page header with the name of the application and a Team Plasma quip. On the right side of the header, there is a button that adds a bug to the database when clicked. After the button is clicked, the user may edit the content of the new bug in the bug-detail view, which is div 3.

Layout.jade: The head of the index page that includes javascript, styles, and other external references. Uses block content to get all other views for the application.

**ii) Provide an outline of the express implementation**

1. **A list of URL mappings and the functions they implement**
   1. **/ - Redirects to index.html**
   2. **/index.html – Returns the main html document and scripts**
   3. **/buglist.html – Returns the html fragment for displaying the list of all bugs**
   4. **/bug/all – Returns a list of all bugs in the database in json format**
   5. **/view/create-bug – Returns the html fragment that is used to create bugs**
   6. **/view/bug/:id – Returns the html fragment to view the bug with the specified id**
   7. **/user/all – Returns a list of all users in the database formatted in json**
   8. **/user/id/:id – Returns the user info with the specified id in json**
   9. **/bug/create – Using POST creates a new bug**
   10. **/bug/update – Using POST updates an existing bug**
   11. **/comment/create – Using POST creates a new comment**
   12. **/comment/pull/:id – Returns the html fragment for viewing all the comments for a specified bug**
2. **A description of sub-applications and how they are called from the main application**
   1. **DAO.js – The “super class” of all the DOA objects, provides helper functions for connecting to a database, fetching data, and inserting / updating data.**
   2. **BugDAO.js – The Data Access Object to fetching, creating, and updating bugs in the database**
   3. **UserDAO.js – The Data Access Object for fetching, creating, and updating users in the database**
   4. **CommetnDAO.js – The Data Access Object for fetching, creating, and updating comments in the database.**
   5. **Each of these sub applications are used by the main controller / router for fetching data. As well as each other for fetching users, comments, and bugs as necesary**

**iii) A diagram of DOM layout and hierarchy**



**iv) Wireframe diagrams that describe the 1 top-level HTML document**

Refer to TerByte Wireframes.pdf

**v) Wireframe sub-diagrams that describe each of the 4 top-level DIV content boxes and the variations of information that will be displayed**

Refer to TerByte Wireframes.pdf

**vi) Navigation buttons, mouse click steps and sequences flows**

All buttons and clicks are described in the view descriptions from i-(e).