INTERFACE CONTROL DOCUMENT

Ethan J. Nephew

VALENCIA COLLEGE

Section 1.0 Purpose of Interface Control Document

The purpose of this document is to document the necessary information required to define the interfaces that serve to connect the Webpage and the Web Service of the To Do application. A goal is to clearly articulate what data is expected when the Webpage communicates to the Web Service and what data is expected when the Web Service responds to the Webpage.

Section 1.1 System Identification

Section 1.1.1 System 1

The first system is the Webpage. The Webpage is the part of the system that is displayed in a web browser. The Webpage is the client.

Section 1.1.2 System 2

The second system is the Web Service. The Web Service responds to GET, DELETE, and POST requests. The Web Service is the sever.

Section 1.2 Interface Description

The interface that I will be describing is the interface of the To Do Webpage application and the interface of the Web Service that fulfills each request.

Section 2.0 Core Operations

Both interfaces will need the ability to:

1. Receive textual information.
2. Transmit textual information.

The Webpage interface will need the ability to:

1. Send parameterized data as an integer.
2. Package and send JSON.

The Web Service interface will need the ability to:

1. Receive parameterized data as an integer.
2. Receive and process JSON.

Section 2.1 System Overview

The data that will be exchanged over the interface will be requests with textual identifiers.

Section 2.2 Functional Allocation

The user will interact with the Webpage for the purpose of viewing, submit, and deleting from a list of activities that they need to accomplish.

When the user opens the Webpage portal, a request will be made to fill the table entries. The Web Service will respond to this request by sending the necessary information to display the table.

When the user clicks the ‘submit’ button, this will send the information to the Webpage interface, the data is sent to the Web Service where that data will be stored into the database.

When deletion occurs, the Webpage will send the identification key of the entry to be deleted to the Web Service. The Web Service will use that key to identify which row in the database to delete. The row will be deleted from the database.

Section 2.3 Data Flow

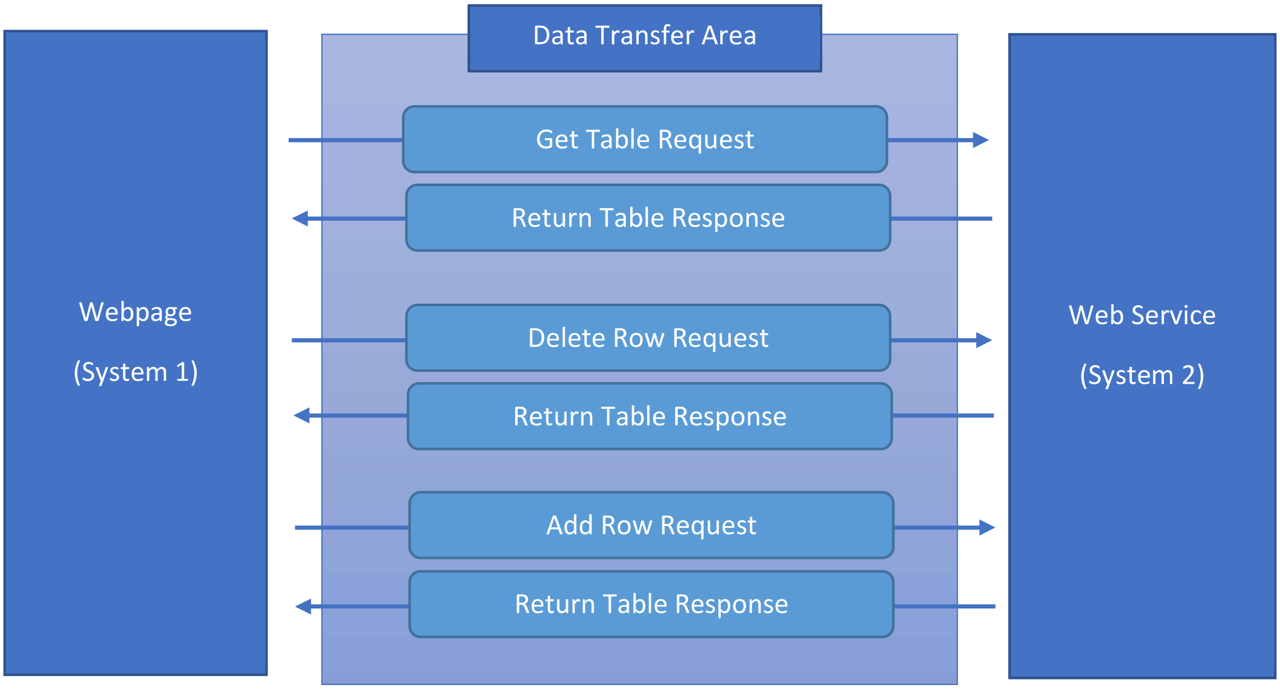

Figure 1 Depiction of the expected flow of data between System 1 and System 2 during normal operation. Through System 1, the user will issue commands, System 2 will fulfill those commands, and System 1 will request table data.

The get table component begins with the Webpage sending a request to the Web Service. The Webservice responds with the appropriate data.

The delete row component begins with the Webpage sending the row identifier to the Web Service. The Webpage will request the table data and the Web Service responds with the amended table data that will be absent the row that was identified for deletion.

The add row component begins with the Webpage sending the data for a new entry to the Web Service. After that, the Webpage requests will request the table data and the Web Service responds with the amended table data that will include the new row entry.

Section 2.4 Transaction Types

There are three type of transactions that will take place.

1. GET for requesting and transmitting table data.
2. DELETE for deleting a row.
3. POST for adding a new row.

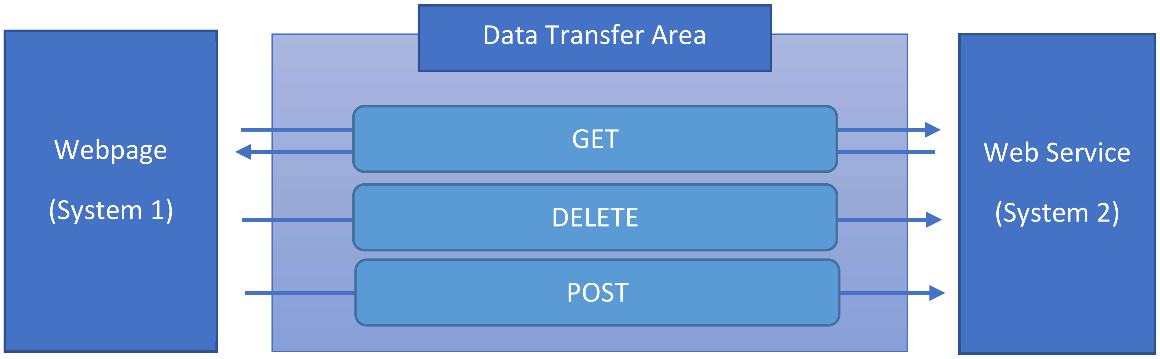

Figure 2 Depiction of the data flow between the HTTP command types. GET has a bidirectional data flow. DELETE and POST have unidirectional data flows. It can also be observed that each interface has 3 end points.

Section 2.5 Security and Integrity

Tomcat will use HTTPS encryption.

Database operations will be handled through parameterization to avoid obvious database attacks.

A filter will cover authentication.

Security context filter will handle privilege management.

Section 3.0 Interface Requirements

These are the detailed requirements for each interface. All modes of communication will be handled through HTTP.

- If a component exists on one side, then it will have a corresponding component on the other side.
- Client components are oriented for the purpose of display.
- Server components are oriented for the purpose of database operations.

Section 3.1 Webpage Component Requirements

Section 3.1.1 GET Request

Resource Access: URL

Media Type: Text Plain

Parameter: None

Request: String

Availability: Always

Initiated by: Page load function

Section 3.1.1 DELETE Request

Resource Access: URL

Media Type: Integer

Parameter: ID

Request: None

Availability: Always

Initiated by: User interaction

Section 3.1.1 POST Request

Resource Access: URL

Media Type: Application JSON

Parameter: None

Request: None

Availability: Always

Initiated by: User interaction

Section 3.2 Web Service Component Requirements

Section 3.2.1 GET Response

Resource Access: URL

Media Type: Text Plain

Parameter: None

Return: String

Availability: Always

Initiated by: Client

Section 3.2.1 DELETE Response

Resource Access: URL

Media Type: Integer

Parameter: ID

Return: None

Availability: Always

Initiated by: Client

Section 3.2.1 POST Response

Resource Access: URL

Media Type: Application JSON

Parameter: None

Return: None

Availability: Always

Initiated by: Client

Section 4.0 Quality, Testing, and Analysis

- Each Web Service component will be tested with Postman.
- The Webpage will be tested through visual inspection.
- Database entries will be visually compared to the data displayed on the Webpage.
- Database inspection will occur during and after every test.