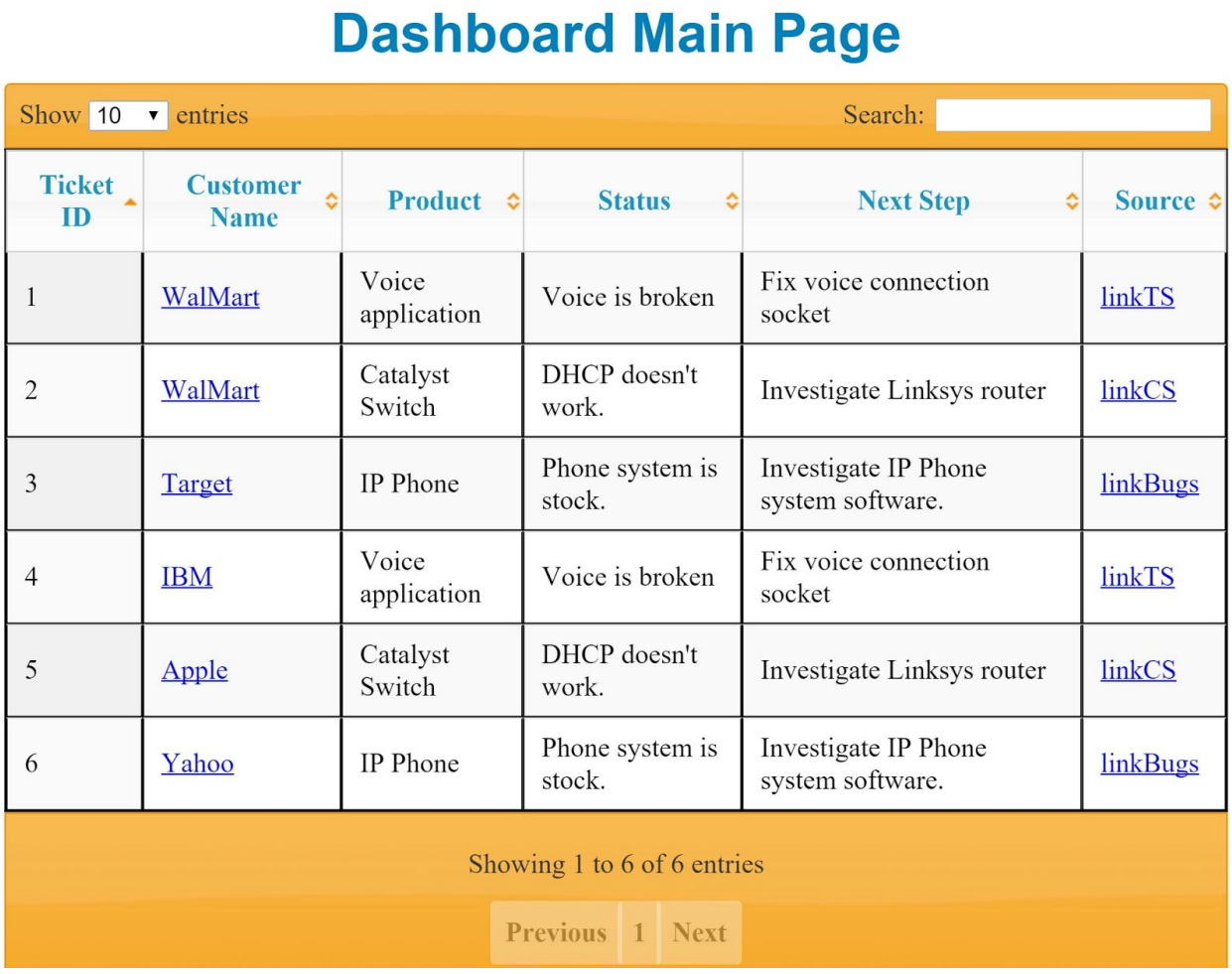
Dashboard Project Design

1. UI Design

Please look at the attached 4 html files.
Data_Table_Main_Jake.html
LinkTS.html
LinkBUGS.html
LinkCS.html

You can open the Data_Table_Main_Jake.html in a web browser and navigate to the following 3 html pages.



The image above is Dashboard Main Page. To explain the design, if a user clicks “WalMart” link under Customer Name field, the page will redirect to a page which will display all the tick info for the customer customer (WalMart).

If a user clicks “linkTS” under “Source”, the page will redirect to a Technical Support page and show details of the current issue.

I couldn’t implement all those html for page redirections. I just wanted to show and explain the concept of the plan.

2. Data retrieval from DB

When the main page redirects to pages for “Customer” or “Source”, its relevant ajax call has to be triggered and provide required data to display the details. Currently if you click links like linkTS, linkCS, linkBugs the new pages will display hard-coded info.
Ajax call will trigger backend source code with data parameter. The Java code will find data using the data parameter, retrieve the data, return it to Javascript, and display in the UI.

3. Database Plan

We can use any DB system for this project. My preference is MySQL which we can download and utilize easily. Using maven and configuration files, we can easily set up db as follows.

1) Maven setup for MySQL connection in pom.xml file

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>${mysqlconnector.version}</version>
</dependency>
```

2) database.properties file set up

```
jdbc.driverClassName=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/dashboard_prj
jdbc.username=root
jdbc.password=password
jdbc.dialect=org.hibernate.dialect.MySQL5Dialect
jdbc.showSql=true
```

3) spring-context.xml config (Spring framework) for Database and Hibernate connections

```
<context:property-placeholder location="classpath:properties/database.properties" />
<bean id="dataSource"
    class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="url" value="${jdbc.url}" />
    <property name="driverClassName" value="${jdbc.driverClassName}" />
    <property name="username" value="${jdbc.username}" />
    <property name="password" value="${jdbc.password}" />
</bean>
<bean id="transactionManager"
    class="org.springframework.orm.hibernate4.HibernateTransactionManager">
```

```
        <property name="sessionFactory" ref="sessionFactory" />
</bean>
<bean id="sessionFactory"
        class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
        <property name="packagesToScan">
                <list><value>com.cisco.dashboard.entity.impl</value> </list>
        </property>
        <property name="hibernateProperties">
                <props>
                        <prop key="hibernate.dialect">${jdbc.dialect}</prop>
                        <prop key="hibernate.show_sql">${jdbc.showSql}</prop>
                </props>
        </property>
        <property name="dataSource" ref="dataSource" />
</bean>
```