Department of Computer Science and Engineering
The University of Texas at Arlington

System Test Plan

Team Real

Project: Consultant Staffing Analysis Tool, C.S.A.T.

Team Members:
Clinton Smith
Gayatri Tiwari
Mariana Flores
Karma G Gurung
Edward Kuykendall

Last Updated: 4/9/2015

Table of Contents

Document Revision History

Revision Number	Revision Date	Description	Rationale
0.1	3/25/15	Initial draft of STP	Began work on first draft of STP
0.2	3/29/15	Continued work on STP draft	Completed sections 1,2, and 3
0.3	3/30/15	Continued work on STP draft	Completed sections 6 and 7
1.0	3/31/15	Completed work on STP version 1.0	Added remaining sections of STP
2.0	4/9/15	STP version 2.0	Modified STP as per feedback

List of Figures

List of Tables

1. Introduction

This document describes the system test plan, or STP, of CSAT, the Consultant Staffing Analysis Tool for Sogeti USA. The following section will outline the product concept and scope of the design of CSAT to be used as a reference for the following sections of the STP.

1.1 Product Concept

CSAT has one primary function- to make the process of selecting a team for a software project easier and more intuitive. Currently Sogeti uses a very manual system to select their teams for projects. Staffing executives analyze availability tables, consultant profiles, and project documents to create the most effective team possible. While this system does work it is very labor intensive, which leads to a "Staffing by availability" problem. This is where the team will be built solely based on who is available at any given point and not necessarily their appropriateness for the project. This is what CSAT intends to address.

CSAT will make the team building process easier by removing the need for tedious tables and documents for staffing analysis and instead replace them with an easy to use graphical interface that will allow the staffing executives to see, at a glance, which consultants would best suit a particular project. Not only that, the system will also generate suggested teams by analyzing the consultant data, and comparing it to the needs of the client; thereby automating a large portion of the staffing process. The executives would then be able to look over entire teams instead of individual consultants making the staffing process much faster and more efficient.

CSAT shall also allow clients of Sogeti to connect to the system and see available consultants. They will be able to create teams and select individual consultants they would like to invite to speak at their business. Clients will only be able to see who is available and their skills but no specific personal information. A client would gain access to CSAT by receiving an invite from Sogeti in the form of a link that directs them to the system.

1.2 Product Scope

CSAT shall consist of a scalable web based application, allowing the user to view the system on both a standard PC and a mobile device, a database to hold the relevant client and consultant information, and a web server to host the application and the database. The application itself shall be written using standard web languages such as HTML, CSS, and Java and the database will be a MySQL database to allow for easy data access. These features shall increase the system's maintainability and portability by allowing it to be hosted on almost any type of web server.

The end users of CSAT, primarily the staffing executives, shall have many features available to them. Primarily, the system shall provide a graphical interface where the staffing executives can see consultant and team data in an easy to read fashion that allows them to quickly make staffing choices. For example, when a staffing executive searches for a consultant to add to a project they see graphs for each consultant that shows their appropriateness for a particular project based on certain criteria, such as experience, career goals, cost, and distance. These graphs provide a brief overview of each consultant, however the graphs are dynamic and the staffing executive can click on the icons on the graph to see more details concerning that particular consultant, for example, if they click the cost icon they will see a breakdown of that individuals pay based on their position as well as other things such as travel costs.

Using the graphical user interface the staffing executive shall also be able to create teams by adding individual consultants to a project. They can then use the same graphs to analyze different teams based on the same criteria as the individual consultants, thus allowing them to create the most effective team possible.

The system shall also provide automatically generated teams for each project. That is, the system shall create suggested teams based on certain project attributes. For example, if a project requires a lower cost team then the system shall provide the most optimal team to achieve the lowest cost or if a project is extremely difficult the system shall provide a suggested team with more experience.

The system shall also provide features for the consultants themselves. It shall allow them to set their preferences. For example, they could select the types of projects they would like to work on, their maximum commute distance and their availability for overtime.

The system shall also provide features for the clients of Sogeti. It shall allow them to see available consultants, create teams, and invite consultants to speak at their business. However clients will not be able to see all information in the system, they will only have access to the consultant's names, skills, and availability.

Lastly the administrators of the system shall have features allowing them to generate user accounts, as well as delete and edit them.

1.3 Testing Scope

This section describes the testing scope of CSAT as it pertains to the tests that are contained within later sections of this document. The testing scope consist of two primary components: test inputs and the test environment. Each of which is outlined below in further detail.

1.3.1 Test Inputs

Due to the nature of CSAT some inputs are impossible to test fully, as they have an infinite input space. As such measures must be taken to reduce the input space to a manageable level. This shall be achieved by limiting the test input scope for all user tests to a simple positive and negative input. That is for example, if the system expects a string in email format then CSAT shall be tested with one valid and one invalid string. The test shall be said to pass if the positive test case succeeds and the negative one fails. This will help to reduce the necessary tested input space to a manageable level.

1.3.2 Test Environment

Due to the fact that CSAT is a web application, and not a physical product, there are some components of its runtime environment that cannot be tested as they are under the control of the end user. Because of this we will be defining our own runtime environment for testing purposes. However this environment will be as close to the end users environment as possible and it will include the following major items.

- Apache Tomcat 7.0 web server
- Java Runtime Environment 6
- MySQL Server on localhost
- Google Chrome/Firefox web browsers
- Stable internet connection for API modules

2. References

2.1 System Requirements Specification

The system requirements specification, or SRS, describes all requirements of CSAT derived by both the customer and the development team. The following is a condensed list of all of the requirements of CSAT to be used as a reference in the following sections of the STP. For further details please see the full SRS document.

2.1.1 Customer Requirements

Req. #	Requirement	Description	Priority
3.1	Support Multiple User Roles	The system shall allow users to login using their username and password. Each user shall have a particular role with its own access. Supported user roles include: consultant, staffing executives, administrators, and clients. Each user shall only be displayed information or given features based on their respective roles.	5 – Critical
3.2	Visual Representation of Consultant Data	The system shall allow staffing executives to view consultant's stored data in an easy to use graphical user interface. The staffing executive can then click on various icons on the graphs to see details of the consultant's data.	5 – Critical
3.3	Search	The system shall allow staffing executives to search the data stored in the system by keyword. For example, the staffing executive can search for projects by entering the project title, or they can search for consultants by entering their name. The system shall also allow the user to filter the results based on various fields.	5 – Critical
3.4	Create Projects	The system shall allow staffing executives to create new projects by entering project information into the system. Project information to be stored in the system includes: client, title, description, cost, start/end date, and team size.	5 – Critical

3.5	Edit Projects	The system shall allow staffing executives to edit projects that are stored in the system.	5 – Critical
3.6	Delete Projects	The system shall allow staffing executives to delete projects that are stored in the system. The system shall store deleted items in the database, even after deletion, to facilitate data recovery if needed. However, deleted items will not appear in search results in the system. Deleted projects can be completely removed by the administrator only.	5 – Critical
3.7	Generate Proposed Teams	The system shall automatically generate teams for a given project. The system shall create the teams based on criteria within the project such as, cost, experience, availability, etc.	5 – Critical
3.8	Manual Creation of Teams	In addition to the automatically generated teams the system shall allow staffing executives to manually create their own teams by selecting team members from the pool of available consultants.	5 – Critical
3.9	Edit Teams	The system shall allow staffing executives to edit teams. They shall be able to add additional team members and delete team members.	5 – Critical
3.10	Delete Teams	The system shall allow staffing executives to delete teams from the system.	5 – Critical
3.11	Manage Profile	The system shall allow all users to manage their profile. Users shall be able to set their preferences, such as distance to client, overtime availability, types of projects, etc. Also they will be able to edit their personal information such as their skill set and address.	5 – Critical
3.12	Manage Skills	The system shall allow users to manage their skills. Users shall be able to add skills, and their skill level. The skill shall then be approved or disapproved by an administrator of the system.	3 – Moderate
3.13	Invite Clients	The system shall allow staffing executives to invite clients to use the CSAT system. The system shall send the invite in the form of an email with a link directing the client to the CSAT system.	3 – Moderate

3.14	Invite Consultants to Speak	The system shall allow clients to invite available consultants to speak at their businesses. The client must provide a description of the topic to be spoken on. Then the system should notify the admins of the client's request.	3 – Moderate
3.15	Android Application	The system shall be provided in an Android application that will be published on the Google Play Store.	2 – Low

Table 1 - Customer Requirements

2.1.2 Packaging Requirements

Req. #	Requirement	Description	Priority
4.1	Software Packaging	The system shall be delivered as a single .zip file containing all source code and resources used.	5 – Critical
4.2	Installation Guide	The .zip file will contain a text file clearly describing how to set up the application on a server.	4 – High
4.3	Administrator Guide	The .zip file shall contain a text file clearly describing the features available to an administrator that will not be included in the normal user guide. For example, adding new accounts, deleting accounts, resetting passwords, etc. It shall also include the initial system administrator passwords and instructions on how to change them once the system is installed.	5 – Critical
4.4	User Guide	The .zip file shall contain a thorough user guide outlining how the various users are to use the system. This will include step by step breakdowns of use cases, list of user types and their features, as well as other pertinent information required by the end users.	5 – Critical
4.5	Google Play Publication	The Android application shall be published on the Google Play Store to allow the users to easily download the app to their Android devices.	2 – Low

Table 2 - Packaging Requirements

2.1.3 Performance Requirements

Req. #	Requirement	Description	Priority
5.1	Response Time	The system shall provide a fast and responsive user interface. A delay of more than a second will have a percent done indicator to keep the user's attention. Delays of more than ten seconds are not acceptable.	4 - High
5.2	Mobile Browser Compatibility	The application will be compatible with most mobile screen sizes and mobile web browsers. Priority shall be given to phone compatibility utilizing the Chrome and Safari mobile browsers. This shall be achieved by creating a dedicated mobile friendly sub domain with all the features of the primary site.	2 – High
5.3	Browser Compatibility	The system shall be compatible with most internet browsers. These include Google Chrome, Firefox, Safari, and Internet Explorer. Priority shall be given to Chrome and Firefox as they are the most common browsers used by Sogeti.	5 – Critical
5.4	Android Compatibility	The Android application shall be compatible with the latest Android APIs.	2 – Low
5.5	Database Scalability	The CSAT database shall be scalable to allow for growth within Sogeti. For example the system shall allow the user to add more user accounts, teams, and projects as the need arises without impacting performance. Currently the system shall support a minimum of 1000 consultant accounts and projects.	5 – Critical

Table 3 - Performance Requirements

2.1.4 Security Requirements

Req. #	Requirement	Description	Priority
6.1	Malicious Input Protection	The system shall validate all user input to ensure that the data entered is correct. For example, the system shall validate inputs to be sure they are not too long, too short, or of the wrong format.	3 – Moderate
6.2	Secure Database Access	The system shall only allow users to access information in the database that is related to their roles. For example, consultants can only see their data, staffing executives can see all consultants' data and projects, administrators can see all data in the system, and clients can only see the consultants' general profile and their availability.	3 – Moderate
6.3	Error Messages	The application will provide safe error messages that avoid displaying user and application details. That is, only enough information to adequately explain the error will be shown to the user.	3 – Moderate
6.4	Sensitive Information	The application will store sensitive information, such as passwords and encryption keys, in a secure manner. The final product shall also be hosted on a secure server protected by Sogeti's VPN. Only users who have access to the VPN shall be able to access the system.	4 – High

Table 4 - Security Requirements

2.1.4 Maintenance, Support, and Other Requirements

Maintenance, support, and other requirements sections are not directly testable and thus are not included in the STP. For further information about the maintenance, support and other requirements sections of CSAT please see section 7 and 8 of the System Requirements Specification.

2.2 Architectural Design Specification

The Architectural Design Specification, or ADS, describes the high level architecture of CSAT as it is broken down into five, distinct layers and their corresponding sub-systems. It also defines major data flows between the layers and described sub-systems. Included in the STP are the Architectural Design Diagram, Data Flow Diagram, and a brief description of each layer, these elements are to be used as references in later sections of the STP. For further details on the architecture of CSAT please see the full ADS document.

2.2.1 Layer Descriptions

Interface Layer

The purpose of the Interface Layer is to provide an easy to use user interface for the CSAT system. These interfaces will accept user requests and send the corresponding data to the Security Layer to be verified, then the data will move on to the rest of the system to be processed. This layer is also responsible for displaying the output of CSAT and formatting that output data to match the user's specific device. For example, a desktop computer or a mobile phone or tablet.

Security Layer

The purpose of the Security Layer is to provide authorization of user commands based on roles as well as authenticate users upon login. When a user makes a request of the system the request is sent to the Security Layer. At that point the Security Layer verifies that the user has sufficient permission to execute the request. If they do the request is sent on to the rest of the CSAT system to be processed, if not a message is returned to the Interface Layer to notify the user that they do not have permission to perform the request. The Security Layer is also responsible for performing output filtering. CSAT will pass in the output requested by the user and the Security Layer will filter it based on their respective role. For example, an admin can see all data while a Client cannot see personal information about Consultants but can see Consultants' skills and availability.

Processing Layer

The purpose of the Processing Layer is to process all user requests of the system and their corresponding backend functions. This layer is also responsible for creating and maintaining all data objects in the system. For example, if the user requests to add a project and has sufficient access to do so, the Processing Layer will create the Project data object, populate it with the data passed in the request, and pass it to the Database Manager Layer to be saved.

Data Layer

The purpose of the Data Layer is to process all requests for data by the system, it also handles all post database query data calculations, for example calculating distance between a Consultant's address and a Client's address. The Data Layer expects a request and a set of search parameters to search with. The Data Layer will then request the needed data from the Database Manager Layer. Once the requested data is received the Data Layer will format the data to match the corresponding request and pass that information back to the requester.

Database Manager Layer

The purpose of the Database Manager is to provide an interface to the actual database. In this way the database itself is abstracted from the system, allowing for increased modularity. The system shall pass in a request for data or a request to save data to the Database Manager Layer and the Database Manager Layer shall convert that into its corresponding database query. Once the request is processed by the database the Database Manager Layer shall format the results and send it back to the requester of the data.

2.2.2 ADS Diagram

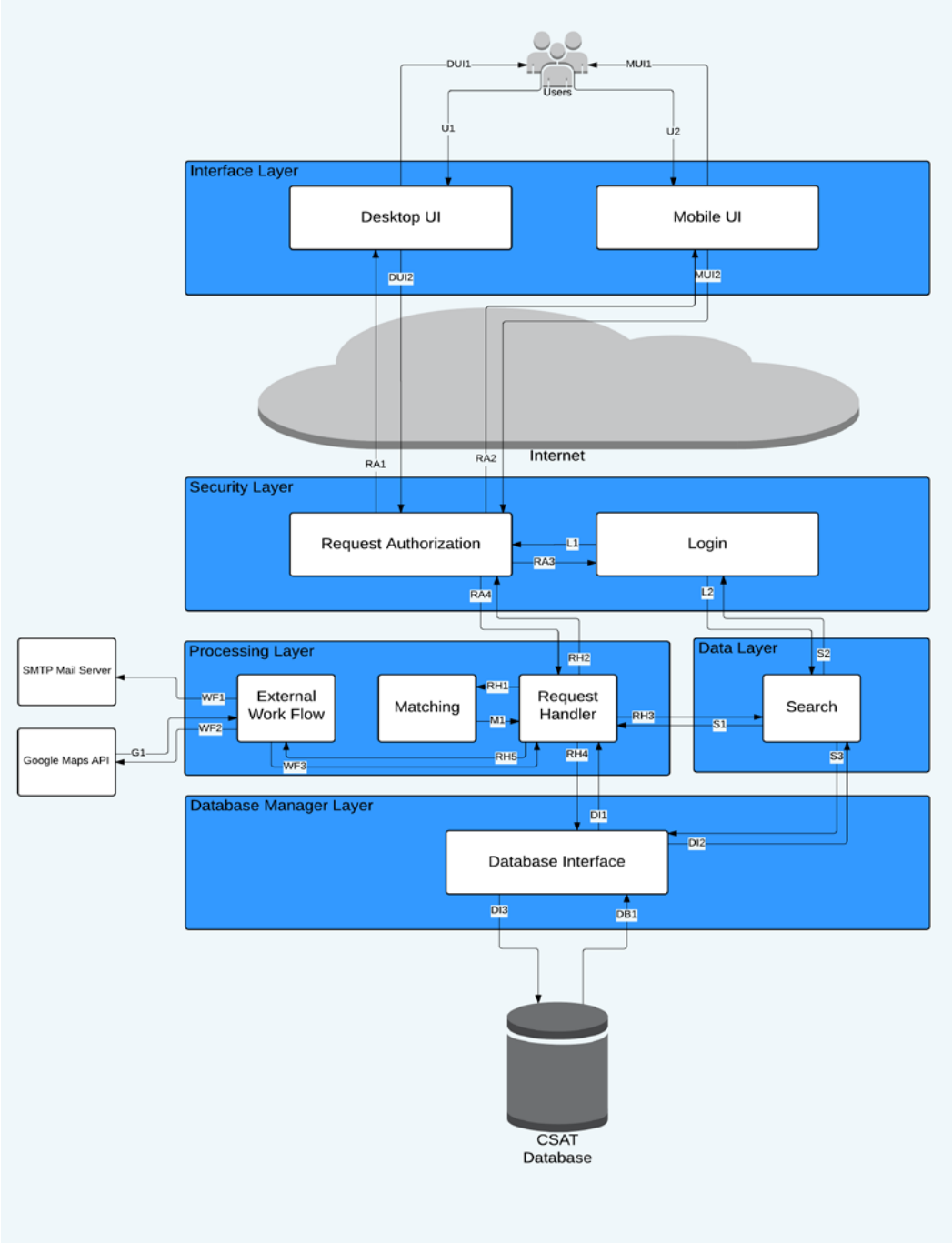

Figure 1 - ADS Diagram

2.2.3 ADS Data Flows

Data Flow ID	Description
U1	User request or action to the desktop UI.
U2	User request or action to the mobile UI.
DUI1	Desktop output to user
DUI2	Request to be authorized sent from desktop UI
MUI1	Mobile output to user
MUI2	Request to be authorized sent from mobile UI
RA1	CSAT system output formatted for a desktop UI
RA2	CSAT system output formatted for a mobile UI
RA3	Request to verify user login credentials
RA4	Authorized request to be processed by the Request Handler subsystem
L1	Users profile once verified by login subsystem
L2	Request for users profile from Search subsystem
WF1	Notification in the form of an SMTP mail message
WF2	Address to get the longitude and latitude coordinates of
WF3	Longitude and latitude coordinates of requested address
G1	Longitude and latitude coordinates of requested address from Google API
M1	Collection of teams matched to a given project
RH1	Project and a collection of consultants to be matched to that project
RH2	Raw CSAT output for a given request
RH3	Request for a collection of data and a set of filter parameters
RH4	Request to save a data object to the database
RH5	Request for external processing, either sending an email notification or getting coordinates from Google Maps API

S1	Requested collection of data based on given filters
S2	Requested user's profile for authorization
S3	Request a collection of data from the database
DI1	Result of saving a data object to the database (True or False with error message)
DI2	Result of database query for given search parameters and formatted for the Search subsystem
DI3	Database query to save or get information from the database
DB1	Raw database query results

Table 5 - ADS Data Flows

2.3 Detailed Design Specification

The Detailed Design Specification, or DDS, describes the detailed systems of CSAT broken down into individual modules. It also describes the specific data flows between those subsystems and their format. Included in the STP are the Detailed Design Diagram, Data Flow Diagram, and a brief description of each module, these elements are to be used as a reference in later sections of the STP. For further details on the detailed design of CSAT please see the full DDS document.

2.3.1 Module Descriptions

Desktop User Interface Module:

The purpose of the Desktop User Interface Module is to render all CSAT pages using a combination of HTML and CSS. The CSS for these pages shall be coded in such a way as to provide an easy to use interface on high resolution, widescreen desktop monitors. The CSS shall also be coded such that the pages match the existing Sogeti website's colors and theme.

Mobile User Interface Module:

The purpose of the Mobile User Interface Module is to render all CSAT pages using a combination of HTML and CSS. The CSS for these pages shall be coded in such a way as to provide an easy to use interface on lower resolution, mobile devices. The CSS shall also be coded such that the pages match the existing Sogeti website's colors and theme.

Graph Renderer Module:

The purpose of the Graph Renderer Module is to render the various graphs throughout CSAT using calls to the requesting pages to get data. The Graph Renderer Module will utilize the Dojo graphing utility developed by IBM to render each graph before displaying it in either the Desktop or Mobile User Interfaces.

Request Builder Module:

The purpose of the Request Builder Module is to create Request objects base on the user inputs into the HTML forms in either the Desktop or Mobile User Interfaces. The Request Builder shall be a Java Server Page that, after accessing the data from the User Interfaces, packages that data and then passes the Request on to the Security Layer for processing. The Request data type shall be described in more detail in later sections of this document.

Request Authentication Module:

The purpose of the Request Authentication Module is to ensure that all requests passed into CSAT are authentic and that the user placing the request has sufficient privileges to perform the action represented by the request. This module is a Java class that will be called by the Request

Builder Module and will be run on the server side of CSAT. Thus adding an extra layer of security between the user and the system.

Request Filter Module:

The purpose of the Request Filter Module is to filter CSAT output such that, for any given user request, the output shall only contain information they are authorized to see. For example an admin of CSAT will have no restrictions while a Client user may only be able to see very basic information. The module will take in a fully processed Request, which contains it produced output, and it will then filter it based on the user who placed the request and output the filtered Request to the proper User Interface to be displayed to the user.

User Authentication Module:

The purpose of the User Authentication Module is to check user credentials against the credentials stored in the CSAT database. Fields such as username, password, session ID and employee ID will all be checked to ensure that the user placing any given request is who they say they are. If any of these fields do not match, the request will be send back up to the User Interface with a failure flag, indicating that the user is not authorized to place the request.

Email API Module:

The purpose of the Email API Module is to provide a simple interface for CSAT to send emails over a SMTP mail server. The Email API Module will take in a Request containing the email address to send the message to, the subject of the email, and the body of the email. Then the Mail API Module shall utilize the Java Mail class to construct an SMTP mail message and send it.

Maps API Module:

The purpose of the Maps API Module is to provide a simple interface for CSAT to retrieve distance information from an outside source, in this case Google Maps API, using a stable interface. The Maps API Module will take in two addresses and return either the longitude and latitude coordinates of the addresses or the distance between the addresses, depending on the function called.

Matching Module:

The purpose of the Matching Module is to generate optimal teams based on a given projects requirements and the available consultants who can work on that project. The Matching Module will take in a collection of available consultants' accounts and a project to match them against. It will then generate teams based on that project, teams such as: most cost effective, most skill effective, most happiness effective, overall most effective and so on. Once calculated it will return a sorted collection of teams back to the caller.

Master Request Observer Module:

The purpose of the Master Request Observer Module, or MROM, is to process and route all authorized requests through CSAT. Its structure follows the Observer Pattern. That is, the Master Observer contains a collection of individual Concrete Observers that process each different type of Request individually. For example there would be a Login Observer, a Logout Observer, an Add Account Observer, and so on. When a Request is passed into the MROM it cycles through each Observer until the correct one is found and then that Observer processes the Request, either by performing the action or routing parts of the action to the subsystems needed. Once the Request is fully processed it is passed back up to the Request Filter Module to be filtered and then sent back to the user to be displayed.

Search Module:

The purpose of the Search Module is to process all requests for filtered data from CSAT. The Search Module will take in a search parameter and a collection of filters. It will then call the Database Interface Module which will return the results from the database. The Search Module will then further filter the results before passing the results back to the module that called it.

Database Interface Module:

The purpose of the Database Interface Module is to provide a stable interface for CSAT to access the physical database. The Database Interface Module and the specific database implementation follow the Bridge pattern; as such the Database Interface Module does not actually do any processing. It simply defines the interface by which all specific database implementations must be designed, thus allowing for easy database changes without changing the system as a whole.

Database SQL Implementation Module:

The purpose of the Database SQL Implementation Module is to provide all access to the physical MySQL database based on the template provided by the Database Interface Module. The Database SQL Implementation Module will take in a Request for some piece of data and it will then construct a query and submit that query to the database. It will then format the output, as needed by the specific Request and return the results to the requesting module.

2.3.1 DDS Diagram

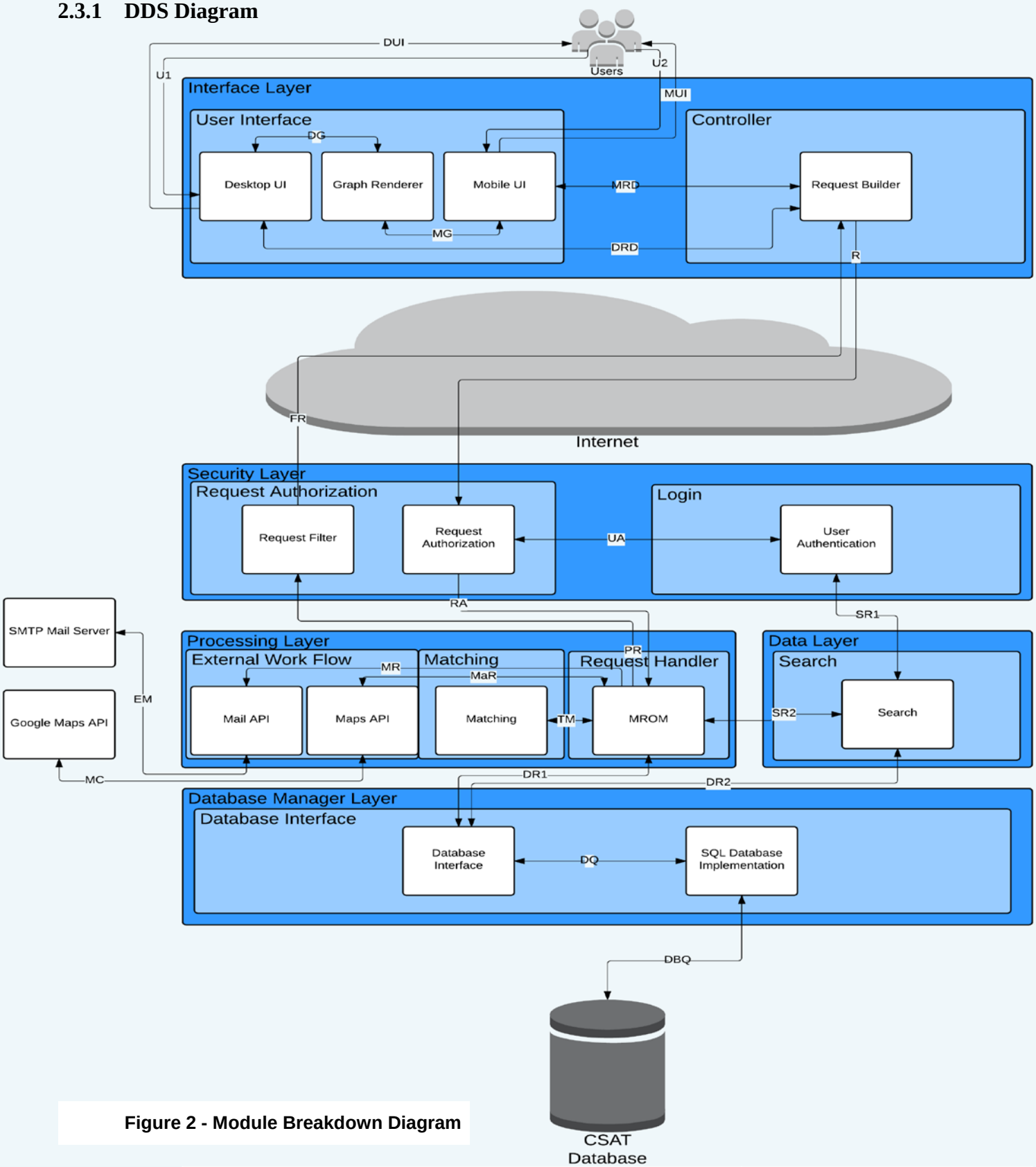

Figure 2 - Module Breakdown Diagram

2.3.1 DDS Data Flows

This table describes each dataflow between modules in CSAT. These dataflow descriptions are at a very high level. For a more detailed description of each dataflow please see each individual module's section later in this document.

Dataflow	Passes	Returns
U1	User inputs for Desktop UI	N/A
U2	User inputs for Mobile UI	N/A
DUI	Rendered HTML page for Desktops	N/A
MUI	Rendered HTML page for Mobile Devices	N/A
DG	Desktop graph data	Rendered graph
MG	Mobile graph data	Rendered graph
MRD	Mobile UI Request data	Processed Request data
DRD	Desktop UI Request data	Processed Request data
R	Request to be processed	N/A
FR	Filtered and processed Request	N/A
UA	User data from Request	T/F Boolean if user is authenticated
EM	Email data	T/F Boolean if email was sent from SMTP mail server
MC	Address or set of addresses	Float longitude and latitude of address or distance between addresses from Google Maps
MR	Mail Request	T/F Boolean if email was sent
MaR	Map Request	Float longitude and latitude of address or calculated distance between addresses
RA	Request that has been authorized	N/A
PR	Processed Request	N/A

TM	Collection of Consultants accounts and a Project	Collection of Teams matched to given Project
SR1	Search Request (From User Authentication)	Filtered search results
SR2	Search Request (From MROM)	Filtered search results
DR1	Database Request (From MROM)	Collection of requested data from the database
DR2	Database Request (From Search)	Collection of requested data from the database
DQ	Database Query	Collection of requested data from the database
DBQ	Formatted SQL Query	Collection of SQL data from the database

Table 6 - Module Data flows

3. Test Items

Figure 3-1 and the following tables show how testing will be broken down and details about each test. Tables will be grouped by testing categories Unit Tests, Component Tests, Integration Tests, and System Validation Tests. Each table entry will show an ID, the module/subsystem/layer to be tested, an expected input, the expected output, a testing method, and a priority.

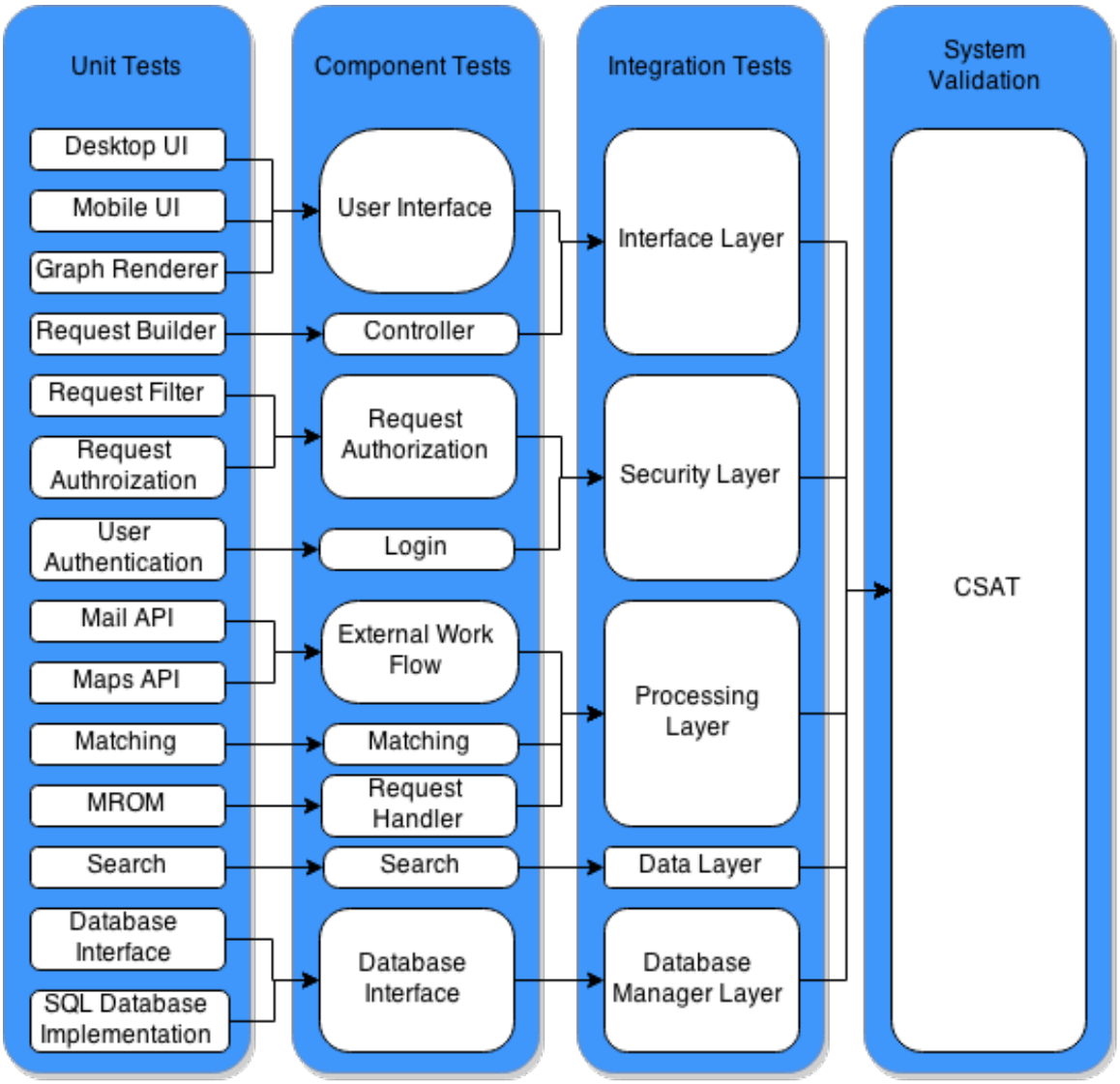

Figure 3 - Testing Category Diagram

3.1 Hardware Testing

There are not Hardware Tests associated with CSAT.

3.2 Unit Testing

3.2.1 User Interface

ID	Module	Input	Output	Test	Priority
UT11	Desktop UI	User Interaction	HTML Form Data	We will test all input forms with data of both correct and incorrect format.	Critical
UT12	Mobile UI	User Interaction	HTML Form Data	We will test all input forms with data of both correct and incorrect format.	Medium
UT13	Graph Renderer	Client data	Graph	We will give the graph renderer various sets of client data with normal and abnormal values and verify that it displays the correct graph or produces an appropriate error message.	Critical

Table 7 - User Interface Subsystem Unit Tests

3.2.2 Controller

ID	Module	Input	Output	Test	Priority
UT21	Request Builder	HTML Form data	Request Object	We will submit form data and verify that the created object is of the correct type and contains the correct data.	Critical

Table 8 - Controller Subsystem Unit Tests

3.2.3 Request Authorization

ID	Module	Input	Output	Test	Priority
UT31	Request Filter	Completed request objects	Completed request object with filtered data	We will submit requests and security levels and verify that they are filtered correctly.	High
UT32	Request Authorization	Request objects	Request object	We will submit request with varying security levels made by accounts with varying security levels to ensure that authorized request are passed on and unauthorized request are rejected with the appropriate error messages.	High

Table 9 - Request Authorization Subsystem Unit Tests

3.2.4 Login

ID	Module	Input	Output	Test	Priority
UT41	User Authentication	Request object	Account object Request	We will submit a login request and ensure that the submitted account is logged in.	High

Table 10 - Login Subsystem Unit Tests

3.2.5 External Work Flow

ID	Module	Input	Output	Test	Priority
UT51	Mail API	Email Request	Email	We will submit a request for an email to be sent to a specific account and verify that the email was sent.	High
UT52	Maps API	Map Request	Longitude and latitude	We will submit a request object for an address get and verify that the address information it returns is correct	Critical

Table 11 - External Work Flow Subsystem Unit Tests

3.2.6 Matching

ID	Module	Input	Output	Test	Priority
UT61	Matching	Collection of account objects	Set of matched teams	We will submit sets of account objects and verify that the teams created are correct based on the given data.	High

Table 12 - Matching Subsystem Unit Tests

3.2.7 Request Handler

ID	Module	Input	Output	Test	Priority
UT71	MROM	Request Object	Specific Request	We will submit different request objects and verify that MROM sends the request to the correct module.	Critical

Table 13 - Request Handler Subsystem Unit Tests

3.2.8 Search

ID	Module	Input	Output	Test	Priority
UT81	Search	User ID	Object ID	We will submit user IDs to search and verify that it attempts to request the correct object from the database	High
UT82	Search	Sets of Objects and a filter	Filtered objects	We will give search a set of objects to filter and verify that the objects are filtered correctly.	High

Table 14 - Search Subsystem Unit Tests

3.2.9 Database Interface

ID	Module	Input	Output	Test	Priority
UT91	Database Interface	Data Request	Data Request	We will verify that data request sent to the database interface are being routed to the SQL database Implementation.	Critical
UT92	SQL Database Implementation	Data Request	SQL Query	We will verify that the SQL Query matches the data that has been requested.	Critical

Table 15 - Database Implementation Subsystem Unit Tests

3.3 Component Tests

3.3.1 Interface Layer

ID	Subsystem	Input	Output	Test	Priority
CT11	User Interface	User Input	HTML Form Data	We will perform all actions available to a user and verify that the produce the expected result.	Critical
CT12	Controller	HTML Form Data	Request Object	We will submit form data and verify that the created object is correct.	Critical

Table 16 - Interface Layer Component Tests

3.3.2 Security Layer

ID	Subsystem	Input	Output	Test	Priority
CT21	Request Authorization	Completed request objects	Completed request object with filtered data	We will submit requests and security levels and verify that they are filtered correctly	High
CT22	Request Authorization	Request objects	Request object	We will submit request with varying security levels made by accounts with varying security levels to ensure that authorized request are passed on and unauthorized request are rejected.	High
CT23	Login	Request object	Account object Request	We will submit a request for account object check that it requests the correct account	High

Table 17 - Security Layer Component Tests

3.3.3 Processing Layer

ID	Subsystem	Input	Output	Test	Priority
CT31	External Work Flow	Email Request	Email	We will submit a request for an email to be sent to a specific account and verify that the email was sent	High
CT32	External Work Flow	Map Request	Longitude and latitude	We will submit a request object and verify that the address information it returns is correct	Critical
CT33	Matching	Collection of account objects	Set of matched teams	We will submit sets of account objects and verify that the teams created make sense.	High
CT34	Request Handler	Request Object	Specific Request	We will submit different request objects and verify that MROM sends the request to the correct module	Critical

Table 18 - Processing Layer Component Tests

3.3.4 Data Layer

ID	Subsystem	Input	Output	Test	Priority
CT41	Search	User ID	Object ID	We will submit user IDs to search and verify that it attempts to request the correct object from the database	High
CT42	Search	Sets of Objects and a filter	Filtered objects	We will give search a set of objects to filter and verify that the objects are filtered correctly.	High

Table 19 - Data Layer Component Tests

3.3.5 Database Manager Layer

ID	Subsystem	Input	Output	Test	Priority
CT51	Database Interface	Data Request	SQL Query	We will verify that the SQL Query matches the data request	Critical

Table 20 - Database Manager layer Component Tests

3.4 Integration Tests

ID	Layer	Input	Output	Test	Priority
IT1	Interface Layer	User Interactions	Request Object	We will perform each action available to a user and verify that an appropriate request object is created	Critical
IT2	Security Layer	Completed request objects	Completed request object with filtered data	We will submit requests and security levels and verify that they are filtered correctly	High
IT3	Security layer	Request objects	Request object	We will submit request with varying security levels made by accounts with varying security levels to ensure that authorized request are passed on and unauthorized request are rejected.	High
IT4	Processing Layer	Request Object	Completed Request Object	We will submit all possible Request Types and verify that the correct action has been taken for that type of request	Critical
IT5	Data Layer	User ID	Object ID	We will submit user IDs to search and verify that it attempts to request the correct object from the database	Critical
IT6	Data Layer	Sets of Objects and a filter	Filtered objects	We will give search a set of objects to filter and verify that the objects are filtered correctly.	Critical
IT7	Database Manager layer	Data Request	SQL Query	We will verify that the SQL Query matches the data request	Critical

Table 21 - Integration tests

3.5 System validation

ID	Validation	Input	Output	Test	Priority
V1	Verify login	Username and password	The system logs the given user in.	We will submit Correct and incorrect data for username and password and verify that the correct action is taken in each situation	Critical
V2	Verify Add/Edit/Delete Project	Project Information Changes	Relevant data is modified in the database.	We will add, edit, and delete a project and verify that the information in the database has changed	Critical
V3	Verify Add/Edit/Delete Account	Account Information Changes	Relevant data is modified in the database.	We will add, edit, and delete an account and verify that the information in the database has changed	Critical
V4	Verify Manual Team Generation	Create a team by selecting consultants by hand	Database is updated with a saved team associated with a project	We will create a team and save it to the database	Critical
V5	Verify Auto Team Generation	A project	A set of possible teams for the given project	We will submit a project and have the system generate a set of possible teams and verify that those teams make sense for the given project	Critical
V6	Verify Search	Keywords and filter	The correct data from the database	We will submit keywords and filters into the system some that should be found and some that should not be found and verify the results are correct with what is in the database.	High
V7	Verify Security	User actions	Results of those actions	We will use an account for each security level and verify that all request made by an account of that security level are fulfilled while request that account is not authorized to submit are rejected.	High

V8	Installation Verification	User actions	Results of those actions	Once CSAT is installed in the users environment validation tests V1-V7 shall be repeated to ensure that CSAT is performing in their environment.	High

Table 22 - System validation

4. Risks

This section will list all the risks associated with the testing phase of the CSAT. Table with list of risk, impact of a risk, severity of a risk, and mitigation strategy to resolve a risk is provided below.

4.1 Risk Table

ID	Risk	Impact	Severity	Mitigation strategy
R1	Infinite Input Space	Possible missed faults due to untested inputs.	Medium	Test key features with
R2	Damaged Database Damage	A missed fault may cause data to be damaged	Medium	CSAT's database shall have at least one redundant backup
R3	Software bugs	Wrong data is sent from one layer to another	High	Dataflow testing using aforementioned tests.

Table 23 - Risks

5. Features To Be Tested

This section describes all the requirements and features in CSAT and how each of them will be tested to ensure the successful completion of the project. Brief description of the requirements and a test approach that will be used are given below.

5.1 Customer Requirements

5.1.1 Support Multiple User Roles

Description: The system shall allow users to login using their username and password. Each user shall have a particular role with its own access. Supported user roles include: consultant, staffing executives, administrators, and clients. Each user shall only be displayed information or given features based on their respective roles.

Test Approach: We will create an account for every single role and will test the login and logout for all the account. We will also query some data and see the displayed information to ensure the information is based on their respective roles.

5.1.2 Visual Representation of Consultant Data

Description: The system shall allow staffing executives to view consultant's stored data in an easy to use graphical user interface. The staffing executive can then click on various icons on the graphs to see details of the consultant's data.

Test Approach: We will login as staffing executive and click on a consultant account and see if the system is able to produce graphical view account information. We will then click on various icons on the graphs to see if the details of the consultant's data are displayed.

5.1.3 Search

Description: The system shall allow staffing executives to search the data stored in the system by keyword. For example, the staffing executive can search for projects by entering the project title, or they can search for consultants by entering their name. The system shall also allow the user to filter the results based on various fields.

Test Approach: We will login as a staffing executive and search the data by entering a few different keyword and see if the system produces the desired result. We will search using different filter using a same keyword to see if the system produces the desired result.

5.1.4 Create Projects

Description: The system shall allow staffing executives to create new projects by entering project information into the system. Project information to be stored in the system includes: client, title, description, cost, start/end date, and team size.

Test Approach: We will login as a staffing executive and create a new project by entering project information into the system. We will then go to current project tab to see if the new project we just created is listed to verify.

5.1.5 Edit Projects

Description: The system shall allow staffing executives to edit projects that are stored in the system.

Test Approach: We will login as a staffing executive and edit a project by changing the project information. We will then go to project details to see if the new changes have been saved to verify.

5.1.6 Delete Projects

Description: The system shall allow staffing executives to delete projects that are stored in the system. The system shall store deleted items in the database, even after deletion, to facilitate data recovery if needed. However, deleted items will not appear in search results in the system. Deleted projects can be completely removed by the administrator only.

Test Approach: We will login as a staffing executive and delete a project by clicking delete project button. We will then check the current project tab to see if the project we deleted is taken off of the list, and we will also go the past project tab to see if the project we deleted is listed.

5.1.7 Generate Proposed Teams

Description The system shall automatically generate teams for a given project. The system shall create the teams based on criteria within the project such as cost, experience, availability, etc

Test Approach: We will click generate team button on the project page to see if the system is able to generate the team based on criteria provided by the project.

5.1.8 Manual Creation of Teams

Description: In addition to the automatically generated teams the system shall allow staffing executives to manually create their own teams by selecting team members from the pool of available consultants.

Test Approach: We will try click on add team member button on the project page to see if it will provide us with the list of consultant with radio button or a check box. We will then click

either a radio button or a check box and then click submit to see if the manual creation of team is successfully executed.

5.1.9 Edit Teams

Description: The system shall allow staffing executives to edit teams. They shall be able to add additional team members and delete team members.

Test Approach: We will login as a staffing executive and edit team by changing the team member. We will then go to project details to see if the new changes have been saved to verify.

5.1.10 Delete Teams

Description: The system shall allow staffing executives to delete teams from the system.

Test Approach: We will login as a staffing executive and delete a team from a project. We will then go to project details to see if there are any teams associated with it.

5.1.11 Manage Profile

Description: The system shall allow all users to manage their profile. Users shall be able to set their preferences, such as distance to client, overtime availability, types of projects, etc. Also they will be able to edit their personal information such as their skill set and address.

Test Approach: We will set different preferences for a user. We will then go to user details to see if the system is able to save it. We will also edit a skill set and address, and then go to user details to verify that the system has updated the address and has sent a skill change request to the admin.

5.1.12 Manage Skills

Description: The system shall allow users to manage their skills. Users shall be able to add skills, and their skill level. The skill shall then be approved or disapproved by an administrator of the system.

Test Approach: We will add a new skill set in a user account and then go to user details to verify that the system has sent a skill update request to the admin, which will show pending next to skill added. We will also check admin email to see if the skill update email has surfaced from the user.

5.1.13 Invite Clients

Description: The system shall allow staffing executives to invite clients to use the CSAT system. The system shall send the invite in the form of an email with a link directing the client to the CSAT system.

Test Approach: After creating a project we will send a client access link along with username and password to client from staffing executive account, then we will try to access that link as client using that username and password to see if we are able to see the project details and update it.

5.1.14 Invite Consultants to Speak

Description The system shall allow clients to invite available consultants to speak at their businesses. The client must provide a description of the topic to be spoken on. Then the system should notify the admin of the client's request.

Test Approach: From client account we will invite one of the available consultants, and then we will check admin email account to see if the request email has surfaced.

5.1.15 Android Application

Description The system shall be provided in an Android application that will be published on the Google Play Store.

Test Approach: We will try to simulate the system in Android studio so that when ready it could be published on the Google Play Store.

5.2 Packaging Requirements

5.2.1 Software Packaging

Description: The system shall be delivered as a single .zip file containing all source code and resources used.

Test Approach: We will extract the .zip file and check if all the source code and resources used are included.

5.2.2 Installation Guide

Description: The .zip file will contain a text file clearly describing how to set up the application on a server.

Test Approach: We will each try to follow a text file that is included with the .zip file to see if we could set up the application on a server. We will also ask some friends from other team to volunteer to set up the application on a server given a text file and see if they are able to successfully set up the application.

5.2.3 Administrator Guide

Description: The .zip file shall contain a text file clearly describing the features available to an administrator that will not be included in the normal user guide. For example, adding new

accounts, deleting accounts, resetting passwords, etc. It shall also include the initial system administrator passwords and instructions on how to change them once the system is installed.

Test Approach: We will ask some friends from other team to volunteer to navigate as an administrator using administrator guide provided with the .zip file.

5.2.4 User Guide

Description: The .zip file shall contain a thorough user guide outlining how the various users are to use the system. This will include step by step breakdowns of use cases, list of user types and their features, as well as other pertinent information required by the end users.

Test Approach: We will ask some friends from other team to volunteer to navigate as various user using the user guide to see if they are able to successfully perform the task.

5.2.5 Google Play Publication

Description: The Android application shall be published on the Google Play Store to allow the users to easily download the app to their Android devices.

Test Approach: We will download the app into an android phone via Android studio which will ensure that the application is able to be published on the Google Play Store.

5.3 Performance Requirements

5.3.1 Response Time

Description: The system shall provide a fast and responsive user interface. A delay of more than a second will have a percent done indicator to keep the user’s attention. Delays of more than ten seconds are not acceptable.

Test Approach: We will navigate through all the features of the system to test the response time.

5.3.2 Mobile Browser Compatibility

Description: The application will be compatible with most mobile screen sizes and mobile web browsers. Priority shall be given to phone compatibility utilizing the Chrome and Safari mobile browsers. This shall be achieved by creating a dedicated mobile friendly sub domain with all the features of the primary site.

Test Approach: We will test the system on mobile web browser like Chrome, Safari, Firefox, and IE.

5.3.3 Browser Compatibility

Description: The system shall be compatible with most internet browsers. These include Google Chrome, Firefox, Safari, and Internet Explorer. Priority shall be given to Chrome and Firefox as they are the most common browsers used by Sogeti.

Test Approach: We will test the system on all major internet web browsers like Chrome, Firefox, Safari, and IE.

5.3.4 Android Compatibility

Description: The Android application shall be compatible with the latest Android APIs.

Test Approach: We will use the latest Android API's for the application to be compatible.

5.3.5 Database Scalability

Description: The CSAT database shall be scalable to allow for growth within Sogeti. For example the system shall allow the user to add more user accounts, teams, and projects as the need arises without impacting performance. Currently the system shall support a minimum of 1000 consultant accounts and projects.

Test Approach: We will test the CSAT with 1000 consultant for the time being. We will also test the system with 100000 consultants to see the impact of 100 times the growth.

5.4 Security Requirements

5.4.1 Malicious Input Protection

Description: The system shall validate all user input to ensure that the data entered is correct. For example, the system shall validate inputs to be sure they are not too long, too short, or of the wrong format.

Test Approach: We will insert incorrect data and see if the system is able to detect it and counter with correct response or notification to the user.

5.4.2 Secure Database Access

Description: The system shall only allow users to access information in the database that is related to their roles. For example, consultants can only see their data, staffing executives can see all consultants' data and projects, administrators can see all data in the system, and clients can only see the consultants' general profile and their availability.

Test Approach: We will login as each role and navigate the features provided to see if they are true to their role. For example, consultant account should not have a create project button.

5.4.2 Error Messages

Description: The application will provide safe error messages that avoid displaying user and application details. That is, only enough information to adequately explain the error will be shown to the user.

Test Approach: We will input wrong information and see if the proper and safe error message is provided.

5.4.2 Sensitive Information

Description: The application will store sensitive information, such as passwords and encryption keys, in a secure manner. The final product shall also be hosted on a secure server protected by Sogeti's VPN. Only users who have access to the VPN shall be able to access the system.

Test Approach: We will try to access other consultant information from the consultant account.

5.5 Maintenance, Support, and Other Requirements

Maintenance, support, and other requirements sections are not directly testable and thus are not included in the STP. For further information about the maintenance, support and other requirements sections of CSAT please see section 7 and 8 of the System Requirements Specification. However some specific features may be tested with user input. These requirements are outlined below.

5.5.1 Troubleshooting Guide

Description: Team Real shall provide a troubleshooting guide that will provide general troubleshooting steps for the system. This guide shall also outline all built in error messages, describing their probable cause and suggested resolution.

Test Approach: The guide shall be given to the end user and they will provide feedback on its usefulness.

5.5.2 Training

Description: Brief training will be provided for users who will utilize the system. This shall be in the form of both a user manual, as stated in requirement 4.4, and a brief training demo with Sogeti.

Test Approach: The guide and training materials shall be given to the end user and they will provide feedback on its usefulness.

6. Requirements Not To Be Tested

This section describes the requirements that are not to be tested. These requirements fall under two categories, either the requirement is technically untestable or the requirement is a non-developmental requirement, such as the user manual. For more information about these requirements please see the full SRS document.

6.1 Maintenance and Support Requirements

6.1.1 Source Code

Description: All source code will be made available to Sogeti in the final product delivery. It will be thoroughly documented using comments in the code as well as in the user manual under the Source Code section. This section will contain descriptions of classes and data objects used in the code, as well as their associated attributes and functions.

6.1.2 Software Support

Description: Team Real shall not be responsible for supporting CSAT after its initial release. All known issues and assumptions will be documented in the user guide under the Source Code section.

6.1.3 Language Support

Description: All documents and user guides shall be provided in English.

6.2 Other Requirements

6.2.1 Extensibility

Description: The web application will be designed and implemented taking future growth into consideration, as more enhancements and functionality are likely to be added. This shall be achieved by developing CSAT such that it is highly modular and all modules are loosely coupled.

6.2.2 iOS Mobile Application

Description: The system shall be converted to an iOS™ app to allow the users to access the system on their mobile iOS™ device.

7. Testing Approaches

This section describes the testing approaches that shall be employed during the development and deployment of CSAT. Included in this section is the general test strategy that will be used, tools, core functions to be tested, as well as the testing metrics.

7.1 Strategy

The testing phase for CSAT will have to be done during implementation. The team will test our code and core functionality to ensure that it is working as anticipated during implementation. The team will try to test their code at the time of implementation so we do not have any code error and bug in future. If there is time towards the end of the product development then there will be extra testing done to catch any bugs or mistakes that may been unaccounted for Mobile UI . We do not have any hardware part to test it.

7.2 Tools

The development team will be using the following tools to test CSAT:

- JUnit – For unit testing of individual functions and features, as well as for backend integration testing.
- Selenium – For regression testing and feature validation at the user level.

7.3 Core Functionality

- Login/Logout
- Add/Edit/Delete Account
- Add/Edit/Delete Project
- Add/Edit/Delete Team Manually
- Auto Generate Team
- Search
- Email Notifications

7.4 Test Metrics

The below metrics will vary according to the priority levels which are listed in the following table.

Priority	Description	Success Criteria	Failure Criteria
Critical	Features that need to be completed for core requirements to function.	100 %	<100%
High	Features that are important to the system but it can still function without them	>=90%	<= 90%
Medium	Features that help polish and refine the system, the system will still fully work without these features.	>=75%	<75%
Low	Features that are listed in future requirements or are extra add on functionality to existing system.	>=50%	<50%

Table 23 - Test Metrics

8. Item Pass/Fail Criteria

This section provides a detailed description of the pass and fail criteria for all tests included in section 3: hardware tests, unit tests, component tests, integration tests, and system validation.

8.1 Hardware Testing

There are not Hardware Tests associated with CSAT.

8.2 Unit Testing

ID	Module	Pass Criteria	Fail Criteria
UT11	Desktop UI	• UI is successfully displayed on the desktop screen	• UI is not displayed on the desktop screen
UT12	Mobile UI	• UI is successfully displayed on the mobile screen	• UI is not displayed on the mobile screen
UT13	Graph Renderer	• Receives input from the UI modules • Successfully sends the correct graph or error message to UI display	• Fails to receive input from the UI modules • Unable to send graph or error message to the UI display • Sends incorrect graph or message to UI display
UT21	Request Builder	• Receives form data from the UI subsystem • Sends correct Request Object to Request Authorization subsystem	• Fails to receive data from the UI subsystem • Sends the incorrect object to the Authorization subsystem • Fails to send object to Request Authorization subsystem
UT31	Request Filter	• Receives request object from the MROM module • Sends filtered data to Request Builder module	• Fails to receive request object from the MROM module • Sends unfiltered or incorrect data to the Request Builder module • Fails to send request object to Request Builder

UT32	Request Authorization	• Receives request object from Request Builder • Sends authorized request to MROM module • Sends error message if unauthorized request is made	• Fails to receive request object from Request Builder • Fails to send authorized request to MROM module • Allows unauthorized requests
UT41	User Authentication	• Receives request object from Request Authorization module • Sends correct Boolean value to Request Authorization module	• Fails to receive request object • Sends incorrect Boolean value to Request Authorization module
UT51	Mail API	• Receives Email request from Request Handler • Sends correct email message to recipient	• Fails to receive email request • Sends incorrect email • Fails to send email to recipient
UT52	Maps API	• Receives Map request from Request Handler • Obtains correct address from Google Maps API	• Fails to receive Map request • Fails to obtain data from Google Maps API • Obtains incorrect data from Google Maps API
UT61	Matching	• Receives a collection of account objects from MROM module • Sends a collection of teams to MROM module	• Fails to receive a collection of account objects from MROM module • Sends incorrect collection of teams to MROM module
UT71	MROM	• Successfully receives Request object • Sends the request to the correct module	• Fails to receive Request object • Fails to send the request to the correct module
UT81	Search	• Receives user ID from User Authentication • Sends requested Account object to User Authentication module	• Fails to receive user ID from User Authentication module • Fails to send the requested Account object to User Authentication
UT82	Search	• Receives set of objects and filter MROM module • Sends filtered object(s) to MROM module	• Fails to receive objects and filter from the MROM module • Fails to send the requested object(s) to the MROM module
UT91	Database Interface	• Receives data request from MROM module • Sends correct data request to SQL Database Implementation module	• Fails to receive data request from MROM module • Fails to send correct data request to SQL Database Implementation module

UT92	SQL Database Implementation	• Receives data request from Database Interface module • Generates correct SQL query	• Fails to receive data request from Database Interface module • Generates incorrect SQL query

Table 8 – Unit testing pass/fail criteria

8.3 Component Testing

ID	Subsystem	Pass Criteria	Fail Criteria
CT11	User Interface	• Displays data correctly in web or mobile browsers	• Fails to display data correctly in web or mobile browsers
CT12	Controller	• Receives HTML Form Data from the User Interface • Sends correct Request object to Request Authorization subsystem	• Fails to receive HTML Form Data from the UI • Fails to send correct Request object to Request Authorization subsystem
CT21	Request Authorization	• Receives request objects from Controller subsystem • Sends filtered data to Controller subsystem in accordance to user's security level	• Fails to receive request objects from Controller subsystem • Fails to send data to Controller • Fails to filter data in accordance to user's security level
CT22	Request Authorization	• Receives Request object from Login subsystem • Sends authorized requests to Request Handler subsystem	• Fails to receive Request object from Login subsystem • Fails to send authorized requests to Request Handler • Sends unauthorized requests to request handler
CT23	Login	• Receives Request object from Request Authorization subsystem • Requests and receives correct account object from Search subsystem	• Fails to receive Request object from Request Authorization subsystem • Receives incorrect account object from the Search subsystem
CT31	External Work Flow	• Receives email request from Request Handler subsystem • Sends correct email to correct recipient	• Fails to receive email request from Request Handler subsystem • Sends incorrect email • Sends email to incorrect recipient
CT32	External Work Flow	• Receives map request from Request Handler • Receives correct address from Google Maps API	• Fails to receive map request from Request Handler • Fails to receive incorrect address from Google Maps API

CT33	Matching	• Receives a collection of account objects from Request Handler • Sends correct collection of teams to Request Handler	• Fails to receive a collection of account objects from Request Handler • Fails to send correct collection of teams
CT34	Request Handler	• Receives request objects • Sends request to correct subsystem	• Fails to receive request objects • Fails to send request to correct subsystem
CT41	Search	• Receives account request from Login subsystem • Sends account request to Database Interface subsystem • Receives requested account from Database Interface subsystem	• Fails to receive account request from Login subsystem • Fails to send account request to Database Interface subsystem • Receives incorrect account from Database Interface subsystem
CT42	Search	• Receives set of objects and filter Request Handler • Sends filtered object(s) to Request Handler	• Fails to receive objects and filter from the Request Handler • Fails to send the requested object(s) to the Request Handler
CT51	Database Interface	• Receives data request from Request Handler and Search subsystems • Sends correct data requested to Request Handler and Search subsystems	• Fails to receive data request from Request Handler and Search subsystems • Fails to send correct data requested to Request Handler and Search subsystems

Table 20 - Component tests pass/test criteria

8.4 Integration Tests

ID	Layer	Pass Criteria	Fail Criteria
IT1	Interface Layer	• Displays data correctly to the user	• Fails to display data correctly
IT2	Security Layer	• Receives request objects from Interface layer • Sends filtered data to Interface layer in accordance to user's security level	• Fails to receive request objects from Interface layer • Fails to send data to Interface Layer • Fails to filter data in accordance to user's security level
IT3	Security layer	• Receives Request object from Interface Layer • Sends authorized requests to Processing Layer	• Fails to receive Request object from Interface Layer • Fails to send authorized requests to Processing Layer • Sends unauthorized requests to Processing Layer
IT4	Processing Layer	• Receives request objects from Security, Database, and Data layers • Sends processed data to correct layer	• Fails to receive request object from Security, Database, or Data layer • Fails to send processed data to correct later
IT5	Data Layer	• Receives account request from Security layer • Sends account request to Database Manager Layer • Receives requested account from Database Manager Layer	• Fails to receive account request from Security layer • Fails to send account request to Database Manager Layer • Receives incorrect account from Database Manager layer

IT6	Data Layer	• Receives set of objects and filter from Processing Layer • Sends filtered object(s) to Processing Layer	• Fails to receive objects and filter from the Processing layer • Fails to send the requested object(s) to the Processing layer
IT7	Database Manager layer	• Receives data request from Processing and Data Layers • Sends correct data requested to Processing and Data Layers	• Fails to receive data request from Processing and Data Layers • Fails to send correct data requested to Processing and Data Layers

Table 24 - Integration tests pass/test criteria

8.5 System validation

ID	Validation	Pass Criteria	Fail Criteria
V1	Verify login	• User will be logged in if username and password correct • User will be notified if credentials are invalid	• User will be logged in if credentials are invalid • User not be logged in if credentials are valid
V2	Verify Add/edit/Delete Project	• UI and database will reflect changes as projects are, deleted, or new projects are created	• UI and database will not reflect changes as projects are, deleted, or new projects are created
V3	Verify Add/edit/Delete Account	• UI and database will reflect changes as accounts are, deleted, or new accounts are created	UI and database will not reflect changes as accounts are, deleted, or new accounts are created
V4	Verify Manual team Generation	• UI and database will reflect changes as teams are, deleted, or new teams are created	• UI and database will reflect changes as teams are, deleted, or new teams are created

V5	Verify Auto Team Generation	• Teams will be auto generated to match a given set of criteria	• Teams will fail to auto generate • Teams will be generated that do not match the given criteria
V6	Verify Search	• Search results will match given criteria • Search results will reflect database data	• Search results will not match the given criteria • Search results will not reflect data from the database
V7	Verify Security	• Information displayed in the UI will reflect the level of security of the user • The user will be allowed to perform authorized actions • The user will not be allowed to perform unauthorized actions	• Information displayed in the UI will not reflect the level of security of the user • The user will be allowed to perform unauthorized actions • The user will not be allowed to perform authorized actions

Table 25 - System validation pass/test criteria

9. Test Deliverables

A record of all tests performed will be kept while developing the CSAT system. This section will include what will be documented, what and how each test will be performed.

9.1 Deliverables

9.1.1 System Test Plan

Includes the entirety of this document to specify how each unit and component will be tested, as well as the system as a whole.

9.1.2 Test Cases

Each test case recorded will include the following information:

- Test ID
- Tester Name
- Description
- Pre-condition
- Post-condition
- Input
- Expected output
- Actual output
- Test Steps
- Test Result
- Notes/ Comments

9.1.3 Test Case Result

Each test case result will include the following information:

- Test ID
- Date and time
- Pass/fail/inconclusive verdict
- Notes/ Comments

9.1.4 Defects

In the case of a test case failure, a defect will be created with the following information:

- Test ID
- Particulars of error. (Error messages, line of error, etc.)
- Possible solution

10. Test Schedule

10.1 MS Project Plan – System Testing Phase

WBS	Task Name	Planned Start	Planned Finish	BCWS
3.4.3	System Testing Phase			
3.4.3.1	System Testing – Phase I (Unit Testing)	4/20/2015	4/24/2015	28hrs
3.4.3.1.1	Desktop UI	4/20/2015	4/24/2015	2hrs
3.4.3.1.2	Mobile UI	4/20/2015	4/24/2015	2hrs
3.4.3.1.3	Graph Renderer	4/20/2015	4/24/2015	2hrs
3.4.3.1.4	Request Builder	4/20/2015	4/24/2015	2hrs
3.4.3.1.5	Request Filter	4/20/2015	4/24/2015	2hrs
3.4.3.1.6	Request Authorization	4/20/2015	4/24/2015	2hrs
3.4.3.1.7	User Authentication	4/20/2015	4/24/2015	2hrs
3.4.3.1.8	Mail API	4/20/2015	4/24/2015	2hrs
3.4.3.1.9	Maps API	4/20/2015	4/24/2015	2hrs
3.4.3.1.10	Matching	4/20/2015	4/24/2015	2hrs
3.4.3.1.11	MROM	4/20/2015	4/24/2015	2hrs
3.4.3.1.12	Search	4/20/2015	4/24/2015	2hrs
3.4.3.1.13	Database Interface	4/20/2015	4/24/2015	2hrs
3.4.3.1.14	SQL Database Implementation	4/20/2015	4/24/2015	2hrs

3.4.3.2	System Testing – Phase II (Component Testing)	4/27/2015	5/1/2015	18hrs
3.4.3.2.1	User Interface	4/27/2015	5/1/2015	2hrs
3.4.3.2.2	Controller	4/27/2015	5/1/2015	2hrs
3.4.3.2.3	Request Authorization	4/27/2015	5/1/2015	2hrs
3.4.3.2.4	Login	4/27/2015	5/1/2015	2hrs
3.4.3.2.5	External Work Flow	4/27/2015	5/1/2015	2hrs
3.4.3.2.6	Matching	4/27/2015	5/1/2015	2hrs
3.4.3.2.7	Request Handler	4/27/2015	5/1/2015	2hrs
3.4.3.2.8	Search	4/27/2015	5/1/2015	2hrs
3.4.3.2.9	Database Manager	4/27/2015	5/1/2015	2hrs
3.4.3.3	System Testing – Phase III (Integration Testing)	5/4/2015	5/7/2015	10hrs
3.4.3.3.1	Interface Layer	5/4/2015	5/7/2015	2hrs
3.4.3.3.2	Security Layer	5/4/2015	5/7/2015	2hrs
3.4.3.3.3	Processing Layer	5/4/2015	5/7/2015	2hrs
3.4.3.3.4	Data Layer	5/4/2015	5/7/2015	2hrs
3.4.3.3.5	Database Manager Layer	5/4/2015	5/7/2015	2hrs
3.4.3.4	System Validation	5/7/2015	5/7/2015	3hrs

11. Approval

Name	Role	Signature	Date
Sogeti USA, Douglas Jones	Customer		
Mike O'Dell	Program Director		
Clinton Smith	Project Manager		
Gayatri Tiwari	Team Member		
Mariana Flores	Team Member		
Karma G Gurung	Team Member		
Edward Kuykendall	Team Member		