IT Connect

CRM Proposal

Aspire CRM: AVGC

Anthony Coots

7-29-2023

Version 1.1

WESTERN GOVERNORS UNIVERSITY

CONTENTS

A. INTRODUCTION

‘IT Connect’ (Vendor) proposes a customer relationship management (CRM) software solution for ‘American Video Game Company’ (Client). The CRM software solution, ‘Aspire’, is a solution designed to replace the client’s previous CRM. ‘Aspire’ proposes integration to the company’s current systems along with offering future systems compatibility to welcome more growth for the client.

A.1. PURPOSE STATEMENT

This document proposes a customer relationship management (CRM) software solution for the client. The solution provided by the vendor will meet the objectives and expectations in the ‘CRM_Requirements’ document detailed by the client to replace the client’s current CRM solution.

A.2. OVERVIEW OF THE PROBLEM

The client is currently using several disconnected manual and automated processes, while outgrowing all aspects of the current CRM. The client has seen growth of 42% causing strain on their current CRM. With strained use of the CRM, the company may be subject to negative effects such as but not limited to morale, business processes, cyber vulnerability, and customer interaction

A.3. GOALS AND OBJECTIVES

GOALS & OBJECTIVES:

- Adaptive
  - Consolidates all (current) contact and business information.
  - Integrates with other systems to allow for sharing of data.
  - Can be enhanced and scaled.
- Manageable
  - Manages activities and tracks sales.
  - Reports the company's activities and interactions with contacts.
- Security
  - Controls access to features based on roles and permissions for the company's users, both internal and remote.
  - Enables access to the system by 3rd party marketing companies under contract.
  - Has robust security.
- Intuitive and user friendly.
- Maintainable / supportable.

A.4. PREREQUISITES

Number	Prerequisite	Description	Completion Date
1	Domain Accounts (Active Directory)	Active accounts within the client's domain will be used to automatically create accounts for the software solution. Users will be later sorted into roles via the software application by client software admins. Users are stored within the client directory as the infrastructure is provided by the client. NOTE: Client software admins must also be formally listed upon SOW.	Upon Statement of Work (SOW).
2	Company Data	The solution will require a collection of current data in use. All data should be translated into either .txt or .csv format for data integration.	1 – 2 weeks after SOW.

A.5. SCOPE

In-bounds:

- Functionality to archive information, without deletion, to maintain records.
- Maintaining versions of records with auditing, workflow, and roll-back.
  - Version control tools.
- Record activity against individual users for auditing and process.
  - Version control tools.
- Soft delete/hard delete. Soft removing the data without full deletion. Hard delete restricted to certain roles/permissions.
- Two-way communication between the solution and Microsoft Exchange and Outlook.
  - Server capable of SMTP handling, server on client infrastructure.
- Creation of accounts via supplied domain accounts, the design of access control the system has is negotiable.
  - UAC / DAC / RBAC / IBAC. Controls data access, workflow, etc. control based on roles/permissions.
- The solution must maintain data for reporting and sharing, along with generating dashboards for overview reports.
- Ticketing system functionality with graphical workflow capabilities and logging for audit trail.
- Data movement to, by choice of schedule, routinely re-import data along with exporting data.

Out of bounds:

- Client has listed preference to work with the internal hosting infrastructure. Vendor will hold no role in setting up or enhancing the client infrastructure any more than what currently resides. The vendor will only provide the software solution while nesting into the client infrastructure.
- All data must be housed within the United States. Vendor is not responsible for data laws and regulations violations that happen directly to the clients' infrastructure, external to the software solution, as the software only nests itself upon the current infrastructure. Alternatively, vendor is not responsible for the clients' infrastructure.

A.6. ENVIRONMENT

‘Aspire’ will use Node.js to create the logic for the website with Express.js acting as the back-end web application framework. With Node.js and Express.js, users should be able to run the latest (stable) releases of all browsers. The servers will use Windows Server 2019, as security is a high priority with new security releases via Microsoft. Windows Server 2019 is currently not nearing the end of support and is a cloud-ready operating system. Windows Server 2019 will also relay to active domain accounts as described in A.4. Additionally, a server will consist of tools such as SQL 2017, for database purposes. Current processes the client has in place, may be used if the client has a relationship to the vendor providing that service and wishes to integrate such. The implementation will nest within the clients’ current infrastructure whether the client makes use of internal hosting and/or cloud-hosting.

B. REQUIREMENTS

REQUIREMENTS:

A. Collection and storage of client data.

B. Sizable to beyond 2,000+ users with 500+ concurrently, this is subject to change from growth.

C. Ticketing system as described.

D. Data reporting (from detailed reports to dashboards).

E. Sales activity tracking including reporting (forecasting).

A. BUSINESS REQUIREMENTS

As the client outgrows the current CRM solution it uses, this is prone to put stress on the overall business of the company. The client, seeing increasing growth will require a CRM solution that can handle daily users, such as customers and employees, that the proposed solution can properly maintain growth (req A.) and client data (req B.) with no negative risk to the client and its contacts, stakeholders, and customers.

Contacts such as any user within the CRM shall be able to effectively raise concerns via a functioning ticketing system to have successful processes occur all while noting the individual who raised the concern, date and time, along with information provided to the requesting individual to ensure professional communication (req C).

B. USER REQUIREMENTS

The client has data to be reported and shared. ‘Aspire’ will cover low-level contact and sales activity data for specific business needs with high-level data dashboards for stakeholders. These dashboards will be customizable with different graphs and KPIs as needed. Data will be accessible to users defined in customizable roles granting permissions as negotiable with the client's expectations of data accessibility (req D, E).

C. FUNCTIONAL REQUIREMENTS

Though not stated in the requirements, in A.5., the option of access control is negotiable to the client as the server uses Windows Server 2019. The client's current CRM roles/permissions will require unique consideration as roles/permissions are defined in the software and may be applied to users, though access to the software may be maintained by A.5. access control.

D. NONFUNCTIONAL REQUIREMENTS

Nonfunctional requirements involve the speed, security, reliability and data integrity. The speed is dependent on the hardware of the clients' infrastructure. The development of the project will take all good business practices to ensure the most efficient software is developed. The security of the data will be protected by proper security practices for connecting to the database along with encryption of data inside the database. Security is also considered further with the negotiation of access control as access control is a Windows Security practice. SLA's will discuss the reliability of the software further detailing proper assurance of routing system functionality.

C. SOFTWARE DEVELOPMENT METHODOLOGY

We the vendor have selected to use a ‘Waterfall’ methodology approach for the software development. The development team for ‘Aspire’ consists of business analysts, solution analysts, developers, quality assurance professionals, a project manager and more.

C.1. ADVANTAGES OF THE WATERFALL METHOD

ADVANTAGES:

- Waterfall is a linear and sequential methodology that presents a solution as a whole.
- The methodology follows the order: Requirements, Analysis, Design, Implementation, Testing, Maintenance.
- The methodology determines a consensus among the client and vendor early in order to develop the solution given all requirements.
- Each of the steps in the methodology must have thorough review and sign-off before advancement to the next phase.
- Waterfall ensures good coding practice by requiring design first removing impromptu coding.

C.2. DISADVANTAGES OF THE WATERFALL METHOD

DISADVANTAGES:

- Testing does not take place until late in the development.
- There is very minimal to no room for change.
- Does not embrace client feedback during development.
- Not ideal for high-risk projects.

C.3. ADVANTAGES OF AGILE

ADVANTAGES:

- Encourages feedback from client / stakeholders / etc.
- Feedback promotes development by iteration and increments.
- “Sprints” that define expectations for development in 1 to 3-week intervals.
- All teams effectively communicate since development only works if on the same page.

C.4. DISADVANTAGES OF AGILE

DISADVANTAGES:

- Resource planning may fluctuate significantly as new developments may cause scope creep, or at least, revisiting the baseline.
- No clear picture, conversely to waterfall, on the final project.
- May cause development to exceed schedule for finish as new features are added.

C.5. BEST SUITED

Waterfall is the best methodology/development method for this project as IT Connect has a strong, sizable team. The expectations and objectives are clearly defined by the client which paints the finished product elegantly. As the solution is code-heavy, this methodology provides the standard of what is expected early as it requires design before the coding/implementation and testing phases. Also, each phase in the sequential methodology requires a formal sign-off verifying the work done in the phase.

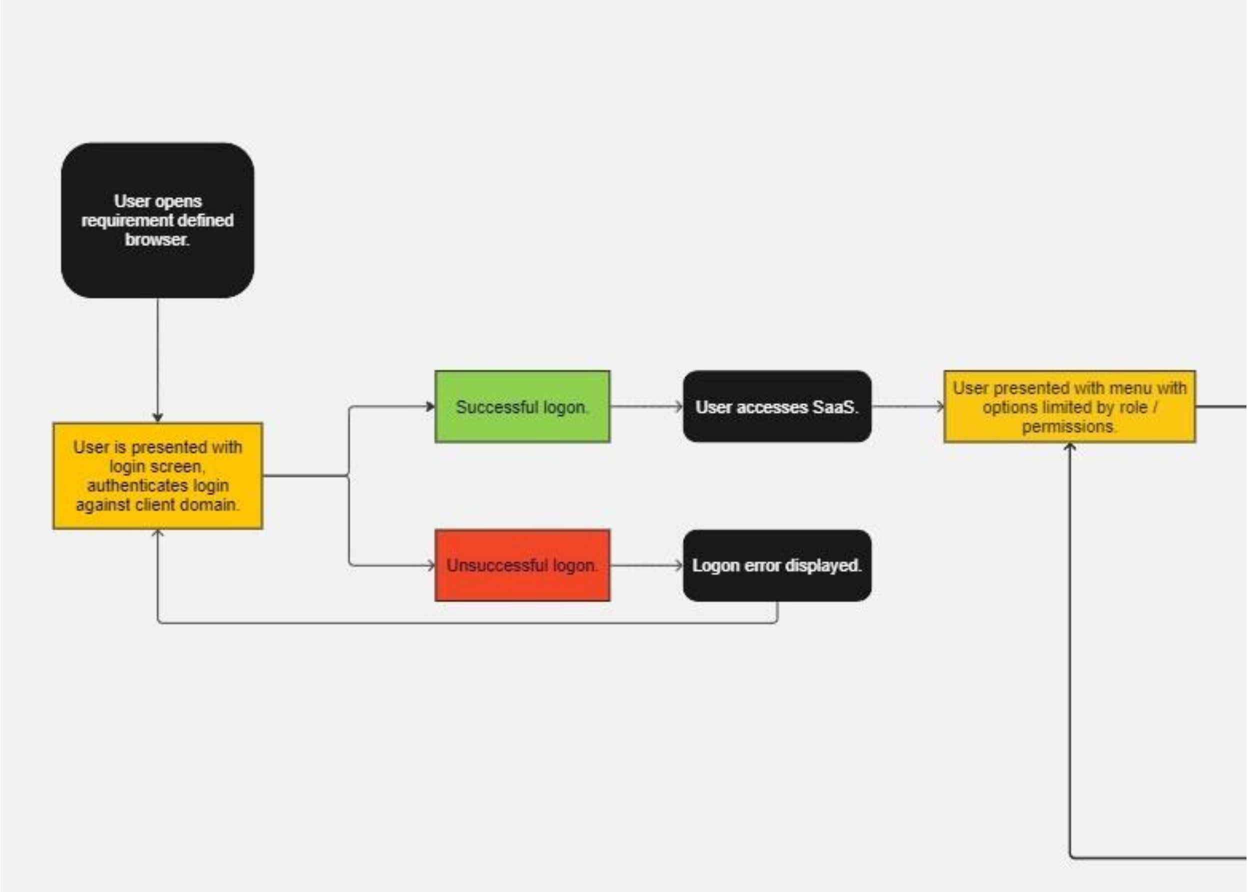
D. DESIGN

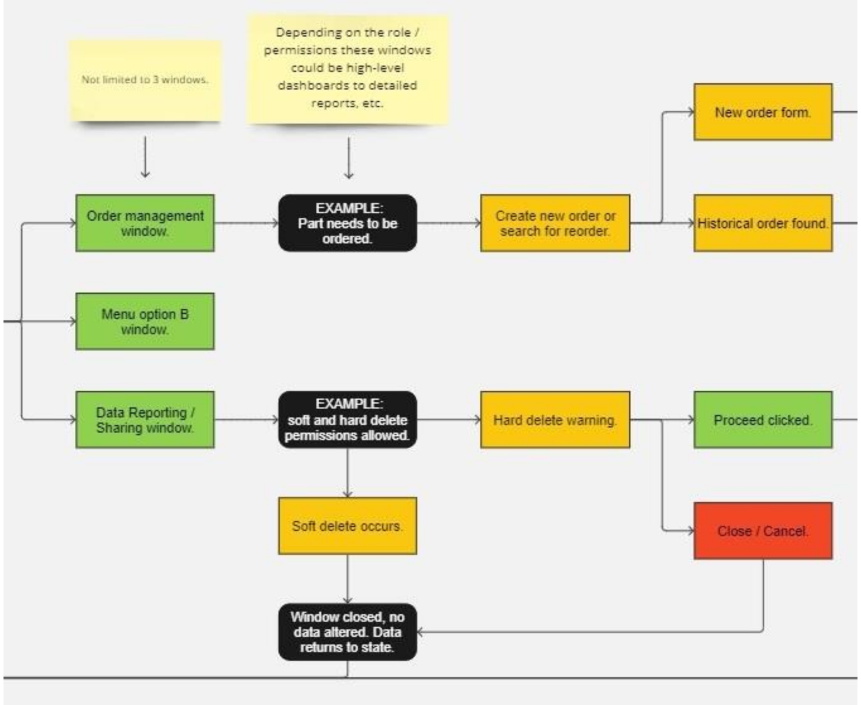
Following is overview design of the software solution:

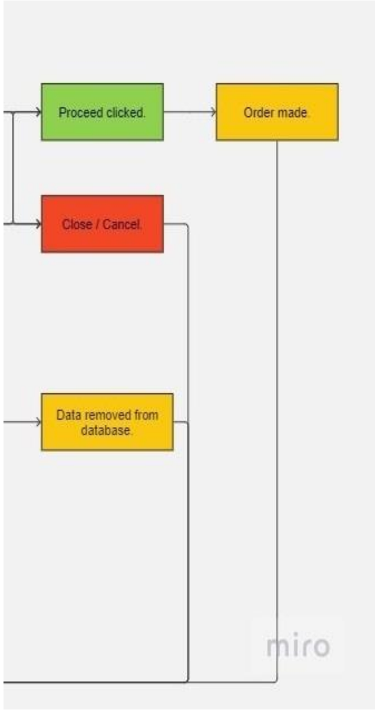
D.1. FLOWCHART

BUSINESS PROCESS RELATIONSHIP:

The flowchart below displays the process of logging on active initial setup within the client's infrastructure. With the use of a browser detailed in the requirements provided by the client, a user will be presented with a login screen when reaching the software solution. Following a logon, if successful, the user will access the software solution. Otherwise, will be redirected to attempt login again. Following a successful login, the user will be presented with a GUI menu (see D.3.) limited by user role/permissions. With general use, upon selecting the number of options limited by user role/permissions, the user will be redirected to a new window of the option they select. Different scenarios present themselves depending on the option selected, such as accessing data reporting/sharing. Upon accessing data reporting/sharing, soft and hard deletes may occur and are handled carefully by each. After following through with each action of the redirected window, the user is once again redirected to that main page (see D.3.)


User opens requirement defined browser.
User is presented with login screen, authenticates login against client domain.
Successful logon.
User accesses SaaS.
User presented with menu with options limited by role / permissions.
Unsuccessful logon.
Logon error displayed.


Not limited to 3 windows.
Depending on the role / permissions these windows could be high-level dashboards to detailed reports, etc.
Order management window.
EXAMPLE: Part needs to be ordered.
Create new order or search for reorder.
New order form.
Historical order found.
Menu option B window.
Data Reporting / Sharing window.
EXAMPLE: soft and hard delete permissions allowed.
Hard delete warning.
Proceed clicked.
Close / Cancel.
Soft delete occurs.
Window closed, no data altered. Data returns to state.


Proceed clicked.
Order made.
Close / Cancel.
Data removed from database.
miro

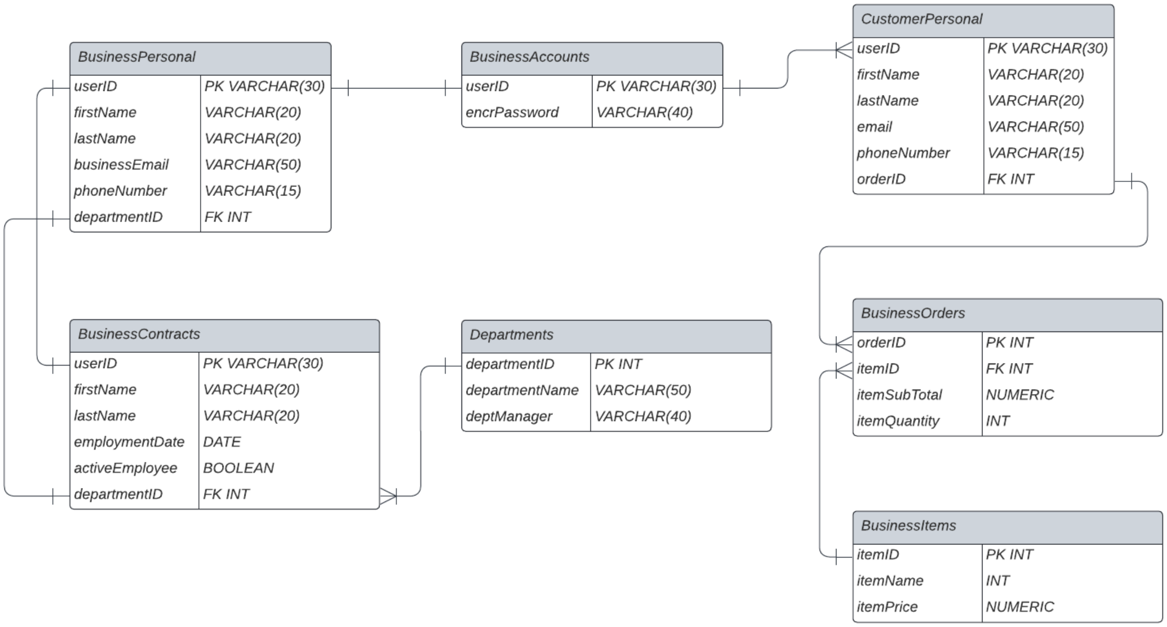
D.2. DATABASE UML DIAGRAM

BUSINESS PROCESS RELATIONSHIP:

Contacts (customers, employees, stakeholders, etc.) will have accounts within the database. Each employee will have a one-to-one entry with their information relevant to the employee along with their department identifier. The department identifier along with user identifier given to the contact will have a one-to-one entry for their contract. The department identifier will relate many entries, many-to-one in the department table. Customers will also have a user identifier with one-to-one or many entries correspondence to the customer's personal information table as this table also contains an order identifier for each order. Proper data structure is required for business efficiency and security. The data shown does contain PII and will have to be protected accordingly.

Database UML Diagram

Anthony Coots | July 28, 2023



BusinessPersonal	
userID	*PK VARCHAR(30)*
firstName	*VARCHAR(20)*
lastName	*VARCHAR(20)*
businessEmail	*VARCHAR(50)*
phoneNumber	*VARCHAR(15)*
departmentID	*FK INT*

BusinessAccounts	
userID	*PK VARCHAR(30)*
encrPassword	*VARCHAR(40)*

CustomerPersonal	
userID	*PK VARCHAR(30)*
firstName	*VARCHAR(20)*
lastName	*VARCHAR(20)*
email	*VARCHAR(50)*
phoneNumber	*VARCHAR(15)*
orderID	*FK INT*

BusinessContracts	
userID	*PK VARCHAR(30)*
firstName	*VARCHAR(20)*
lastName	*VARCHAR(20)*
employmentDate	*DATE*
activeEmployee	*BOOLEAN*
departmentID	*FK INT*

Departments	
departmentID	*PK INT*
departmentName	*VARCHAR(50)*
deptManager	*VARCHAR(40)*

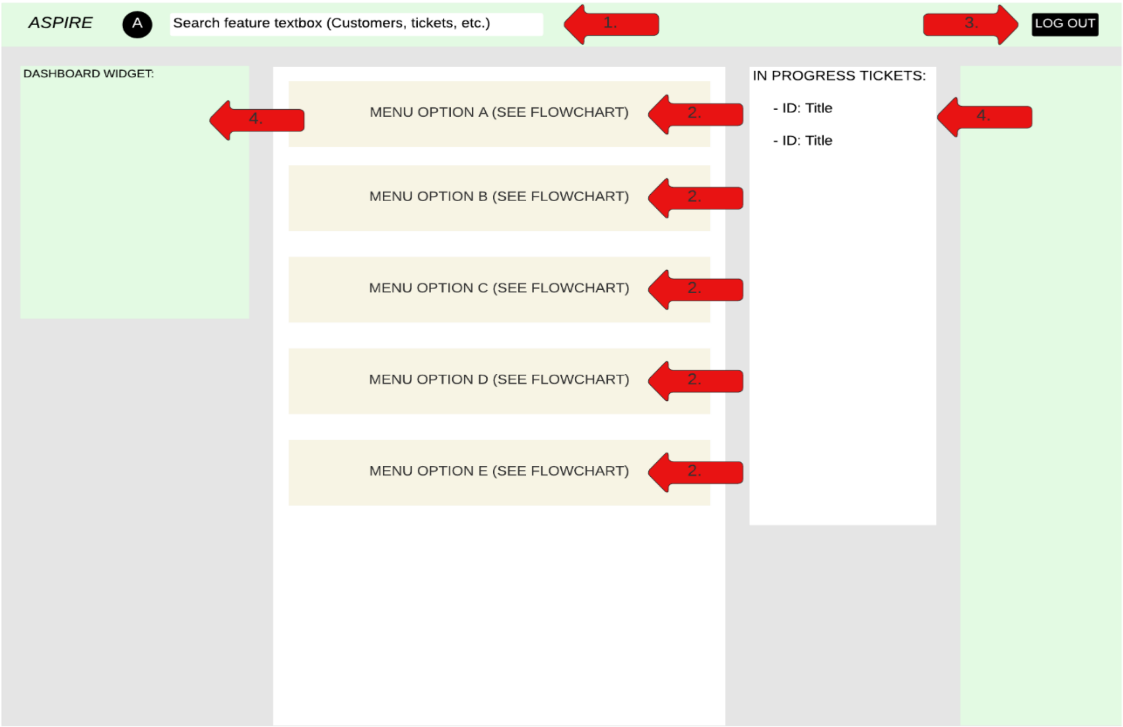
BusinessOrders	
orderID	*PK INT*
itemID	*FK INT*
itemSubTotal	*NUMERIC*
itemQuantity	*INT*

BusinessItems	
itemID	*PK INT*
itemName	*INT*
itemPrice	*NUMERIC*

D.3. GUI

BUSINESS PROCESS RELATIONSHIP:

The GUI shown demonstrates the “main” page after being redirected post-logon. A search feature textbox is provided to do immediate redirection to any business processes within the software solution. The business processes selectable in the search feature textbox are only available depending on the permissions/roles a user is allowed/grouped in. Menu options listed on the main screen will allow users to be redirected to features specific to users such as detailed data reporting, tickets, sales activity, and etcetera. Multiple may be listed depending on the user. A log-out button is demonstrated on all pages after being redirected from the logon screen. This provides a safe exit from the current session for a user. Finally, customizable widgets are provided to fill page space and provide high-level reporting for users that may also be selected to redirect to low-level demonstrations of the presented widget data.



Employee Example Post Sign-On

Anthony Coots | July 28, 2023

ASPIRE
A
Search feature textbox (Customers, tickets, etc.)
1.
3.
LOG OUT
DASHBOARD WIDGET:
4.
MENU OPTION A (SEE FLOWCHART)
2.
MENU OPTION B (SEE FLOWCHART)
2.
MENU OPTION C (SEE FLOWCHART)
2.
MENU OPTION D (SEE FLOWCHART)
2.
MENU OPTION E (SEE FLOWCHART)
2.
IN PROGRESS TICKETS:
- ID: Title
- ID: Title
4.

GUI Control Mapping			
ID	Control	Property	Data Source
1	Search box	On application open text is empty.	Will compare against database
2	Button	Will open new page, see flowchart.	Defined by roles/permissions.
3	Button	On click will log user out, returns to login screen.	N/A.
4	Widget	Will return high-level data reporting configured by user or defined in role. Ticket ID and title provide high level overview of ticket though not all details, hence, high-level.	Varies on widget, relates to database (ticketing system will be added to UML design after client negotiation as to how ticket hierarchy should be implemented.)

E. TESTING

Three different functional aspects of ‘Aspire’ are website navigation, functioning communication, and stability. Below are the testing methods for each of the aspects.

E.1. UTILITY TESTING

A finished product ensures it is easy to navigate. Any interface between logging in to detailed data reporting must demonstrate easy navigation and tools for the task. Navigating the solution with ease encourages users to use the service. The product must also convey communication for the task at hand. The communication channels may vary, such as via ticketing system, login verification, order processing, and more. The solution must also be able to adjust to growth accordingly. The client company is growing and should feel comfortable moving forward as the solution scales with them.

E.1.1. NAVIGATION TESTING

Requirement to be tested: Is the software solution navigable?
Preconditions: The solution should be developed and ready for testing in the methodology.
Steps: The steps the tester must execute to test the feature. 1. During testing, create “ghost” users for logon. 2. Grant the user full permissions.

3. Thoroughly navigate all windows accessible to ensure easy navigation with proper proceeding or returning functionality.

4. Do a once over navigation to the deepest window within the solution and be able to successfully navigate back to the home screen (post logon).

5. Select logoff feature as seen in GUI and verify the solution returns to the login window.

6. Delete “ghost” accounts.

Expected results:

The results should guarantee easy navigation through the solution without immediate concern/loss of location.

Pass/Fail: Pass

During the testing phase, user accounts with full permissions were able to navigate through the software solution without concern/loss of location. Users where then deleted to remove vulnerability upon release.

E.1.2. COMMUNICATION TESTING

Requirement to be tested:

Does the software solution appropriately communicate depending on the task at hand?

Preconditions:

The communication channels that happen with the system must be defined by the client.

The solution should be developed and ready for testing in the methodology.

Steps: The steps the tester must execute to test the feature.

1. During testing, create “ghost” users for logon.
2. Grant the user full permissions.
3. Thoroughly test all communications work as negotiated with the client. (I.E., login to database).
4. Once completed, delete “ghost” accounts.

Expected results:

The results should guarantee proper communication throughout the solution to many internal or external points without immediate concern.

Pass/Fail: Pass

During the testing phase, user accounts with full permissions were able to communicate throughout the software solution without concern (data leak, validation of accounts, etc.). Users where then deleted to remove vulnerability upon release.

E.1.3. STABILITY TESTING

Requirement to be tested:

Is the software solution stable?

Preconditions:

The solution should be developed and ready for testing in the methodology.

Ways to demonstrate use of over 2000+ users and 500+ concurrent users.

Steps: The steps the tester must execute to test the feature. 1. During testing, auto-generate 2000+ “ghost” users for logon. 2. Grant the users with varying to full permissions. 3. Thoroughly test all the users can access the software, by automation. 4. By automation, have 500+ concurrent users to guarantee stability. (Evaluate server performance). 5. Once completed, delete ALL “ghost” accounts.
Expected results: The results should guarantee stability through server and database resource observation while hosting 2000+ users and 500+ users concurrently.
Pass/Fail: Pass During the testing phase, the database was able to store more than 2000+ users and the server was able to host more than 500+ concurrent user interactions with the software.