Customer Support Tool

Francisco J. Castillo Rivera
Master in Computer Science
Advisor: Othoniel Rodríguez, Ph.D.
Electrical and Computer Engineering & Computer Science Department
Polytechnic University of Puerto Rico

Abstract *— Small sales companies rely on their customers' satisfaction after sale has been made. A big part of a customer satisfaction relies on the customer support given to them after the product has been sold. The problem with small companies is that must of the time they produce enough money to have all people needed to maintain a good customer service. It gets even worst when a company sales technology to other business owner. Having a customer support tool is a must for the company making the sales in order to keep track of day to day equipment problems. A tool that handles available personnel in order to assign tasks, send notifications of new customer calls, keep track of unresolved problems, and even reminding the technicians when a preventive maintenance is needed, will help a small company keep their reputation with the customer they already have and make an entrance for new customers.*

Key Terms *— Customer Support, Data Handling, Service, Small business.*

INTRODUCTION

The Customer Support Tool is a web-based system with the capability of storing pending or resolved day-to-day customer calls, notifying the available technician via email or text message when a new call appears, notifying personnel when a preventive maintenance is needed for a specific equipment and guide them through the process, help sales persons to submit new quotations inquiries, notify sales persons when the quotation is ready for the customer, storing lead customers for future follow-up, and generating reports for customers leads and day-to-day customer calls.

The system main components are the SQL database, and the content management software. The content management software allows the development of graphical user interface using languages like JavaScript, PHP, HTML, and CSS and provides easier way to handle users.

The content management software is responsible for storing the user input in the SQL database component. It is also responsible for handling user authentication and authorization in order to keep the Customer Support Tool data safe. Both components are in the same web server.

Having a customer support tool is a must for the company making the sales in order to keep track of day to day equipment problems. A tool that handles available personnel in order to assign tasks, send notifications of new customer calls, keep track of unresolved problems, and even reminding the technicians when a preventive maintenance is needed, will help a small company keep their reputation with the customer they already have and make an entrance for new customers. There is a popular saying "The best marketing strategy is the word of mouth".

REQUIREMENTS SPECIFICATION

Based on the business needs, stakeholders' discussion and current customers' feedback we have outline functional and nonfunctional requirements to make sure we meet all business needs in order to provide a better service.

Functional Requirements

- The system stores and allows access to customer calls in a database.
- The system keeps customer calls as pending or closed.
- The system notifies the available technician via email or text message of a new entry using REST API
- The system generates a report of stored customer calls.

- The system allows sales user to request new equipment quotes.
- The system provides a way to notify the sales user when a quote is ready for the customer using REST API.
- The system stores and allows access to customer leads in an organized matter.
- The system stores and allows access to customer visits in order to keep customer follow-up history.
- The system generates a report of customer visits stored in the database.

Nonfunctional Requirements

- The system must be available 24/7.
- The system REST API must be accessed using HTTPS.
- The system must process a high volume of messages.
- The system must recover from failures.

Use Cases

This section describes how the Customer Support Tool should work and describes the most important system's features. It also describes interaction between system components: SQL Database and content management software (CMS). Each use case focuses on a specific scenario and describes the steps that are necessary to bring it to successful completion [1]. We have used a tabular approach to easily provide sufficient details about these interactions and also provide consistency between use cases. Here is an explanation of the template used:

- Use Case Id (required) - each use case have a use case id defined by UC-{number}
- Use Case Name (required) - each use case have a name to easily identify it. Basically is the operation's name.
- Description (optional) – column is used to describe the use case with more detailed explanation.
- Preconditions (optional) – required state or operations to be performed before use case.
- Actors (required) – entities involved in use case.
- Normal Sequence (required) – list of ordered steps to complete operation.
- Post conditions (optional) – required operations to be performed after use case. Could be some reset of state, etc.
- Exceptions (optional) – used for exceptions or errors when executing normal sequence.
- Comments (optional) – used for general or useful information.

Here are the most important use cases in the Customer Support Tool:

Table 1
UC-01 Store New User Information

Description	End user creates new account through graphical interface.
Preconditions	A SQL database has been created to store information.
Actors	CMS, SQL DB, End User
Normal Sequence	1. End user access CMS through graphical interface to store information. 2. CMS connects to SQL DB. 3. CMS performs HTTP POST method. 4. SQL DB stores information in corresponding table. 5. SQL DB sends acknowledgement to CMS. 6. Notifications is shown to End User.
Postconditions	N/A
Exceptions	1. User cannot be created due to DB connection error. It must be notified to the user.
Comments	N/A

Table 2
UC-02 Store New Customer Call

Description	User stores new entry to system.
Preconditions	User is logged in to system and has the role to store new entries. SQL DB has already been created.
Actors	CMS, SQL DB, End User
Normal Sequence	1. End user fills call information through graphical interface.

	2. CMS connects to SQL DB. 3. CMS performs HTTP POST method. 4. SQL DB stores call information. 5. SQL DB sends acknowledgement to CMS. 6. Notification is shown to End User.
Postconditions	N/A
Exceptions	1. Information cannot be stored due to DB connection error. It must be notified to end user.
Comments	N/A

Table 3
UC-03 Send Text Message or Email Notification to Technician

Description	Notification is sent to user with the technician role.
Preconditions	User has created an account with complete information.
Actors	End User, CMS, REST API
Normal Sequence	1. CMS prepares information to be sent. 2. CMS sends information using REST API (text message) or SMTP (email). 3. Notification is received by end user. 4. CMS receives HTTP OK response
Postconditions	N/A
Exceptions	1. If notification cannot be sent, CMS response with a 4xx HTTP Status.
Comments	N/A

Table 4
UC-04 Send New Quote Request

Description	User request new quote.
Preconditions	User has an account created.
Actors	End User, CMS, SQL DB
Normal Sequence	1. User fills quote information through graphic interface. 2. CMS connect to SQL DB. 3. SQL DB stores quote information. 4. SQL DB sends acknowledgement to CMS. 5. CMS sends notification user.
Postconditions	N/A
Exceptions	1. Information cannot be stored due to DB connection error. It must be notified to end user.
Comments	N/A

Table 5
UC-05 Retrieve Customer Leads Using Filters

Description	User retrieves customer leads using filters.
Preconditions	User has an account created.
Actors	End User, CMS, SQL DB
Normal Sequence	1. User selects filter through graphical interface. 2. CMS connects to SQL DB. 3. CMS performs HTTP GET method with desired filters. 4. SQL DB responds with filtered information. 5. CMS shows information to user through graphical interface.
Postconditions	N/A
Exceptions	1. Information cannot be stored due to DB connection error. It must be notified to end user.
Comments	N/A

Table 6
UC-06 Generate Customer Calls Reports Using Filters

Description	User generates customer calls report using filters.
Preconditions	User has an account created.
Actors	End User, CMS, SQL DB
Normal Sequence	1. User selects filter through graphical interface. 2. CMS connects to SQL DB. 3. CMS performs HTTP GET method with desired filters. 4. SQL DB responds with filtered information. 5. CMS uses fPDF PHP class to generate report in PDF format.
Postconditions	N/A
Exceptions	1. Information cannot be stored due to DB connection error. It must be notified to end user.
Comments	N/A

Table 7
UC-07 Notify Technician of Preventive Maintenance

Description	Notification is sent to users with technician role.
Preconditions	User has created an account with complete information.
Actors	End User, CMS, REST API
Normal Sequence	1. CMS prepares information to be sent. 2. CMS sends information using REST API. 3. Notification is received by end user.

	4. CMS receives HTTP OK response
Postconditions	N/A
Exceptions	1. If notification cannot be sent, CMS response with a 4xx HTTP Status.
Comments	N/A

DESIGN SPECIFICATIONS

The Customer Support Tool is designed with extensibility and scalability in mind. We are taking great care in designing a framework which can be updated easily. Many of the anticipated changes to our system in future phases will only require adding new types of data and changing the user presentation code to make use of these new data. The design will only require "plugging in" these new types of data without refactoring the logic that passes the data over the network, retrieves and updates the database, etc. There are three basic, logical components of the system: Database Engine, Server Application, and Client Applications.

Database Engine – SQL data source used to handle data regarding users and service information.

- Existing open source software: phpMyAdmin
- Hosts the backend database which is used for central data storage.

Server Application –Application that resides in a web server.

- Implemented in HTML, PHP and Javascript
- Provides methods and procedures that can be invoked remotely by a client application.
  - Retrieve data.
  - Update data.
  - Create new data.
  - Generate reports.
- Central process which can make all decisions that arise due to the distributed nature of this application.
  - For instance, when a client wishes to update a data, there may be conflicts that need to be resolved if another client has updated the same data.
  - The server can coordinate conflict resolution with the client application.

Client Application – Existing application to interact with server application and end user.

- Existing Browsers.
- Contains all presentation logic.
- Interacts exclusively with the user.
- Communicates with the server application through HTTPS.

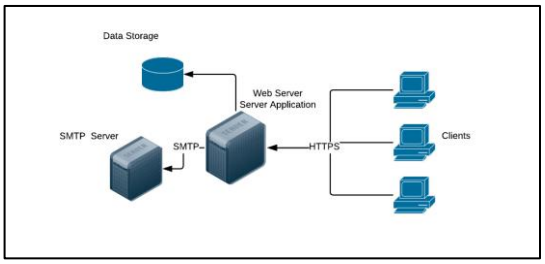

Figure 1
Overall System Architecture

DETAILED DESIGN SPECIFICATIONS

This section includes detailed design specifications. We have used small descriptive paragraph, activity diagram, class diagram and data store schema to better illustrate the functionality of the application component.

CMS Design Specifications

The CMS is in charge of the operations and interaction between the user and data. It will decide the system behavior depending on the user input. The interaction with the CMS is only for registered users of the system.

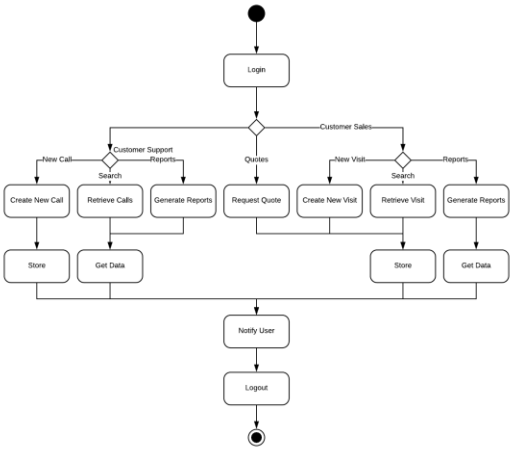

Figure 2
CMS Activity Diagram

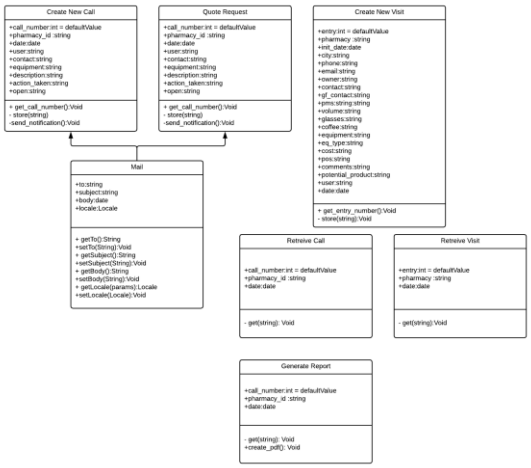

Figure 3
CMS Class Diagram

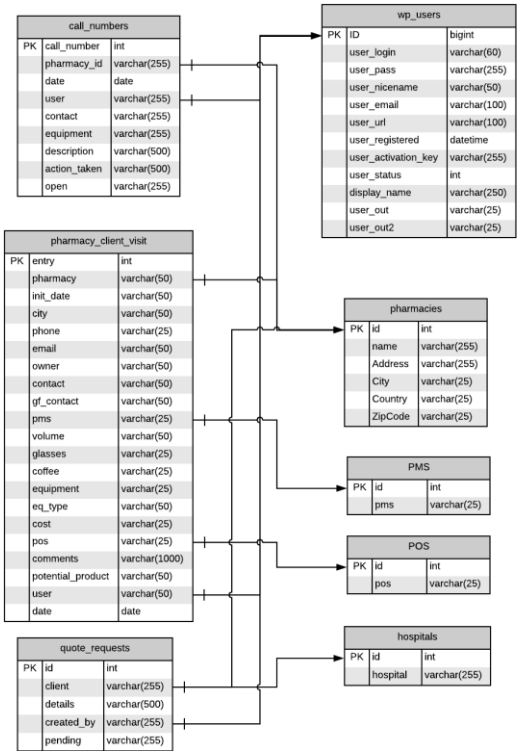

Figure 4
CMS Data Store Schema

IMPLEMENTATION PLAN

This section outlines the process to be taken in order to implement the Customer Support Tool. The application will be implemented on a web server with the requirements for hosting the server application and the data storage. Once installed, a data entry will be performed in order to have the latest company's list of clients and users in the data storage. Employee roles will be defined and access will be determined on each role.

Also, due to the complexity of maintaining a local application server and smtp server, a web hosting service will be implemented to reduce down time risk. A three-month support period will be provided to assure the application is working as expected and fix any issues that may have occur during the implementation phase.

TEST PLAN

In order to ensure the Customer Support Tool is working properly as designed, a series of tests will be performed to validate that the software meets the customer requirements. To validate the code use for our application, a series of standards have been developed for testing each time a change is made. A source code review is done to validate the code meets standards. For the functionality validation, manual tests were performed to ensure the application has been built with the customer requirements in mind.

RESULTS

The Customer Support Tool was successfully implemented and deployed. All components have passed their Unit Tests and their respective Acceptance Tests, after executing these tests we have seen no issue within the tests performed.

FUTURE WORK

The scope of the project was to develop a system that stores customer support data to facilitate day to day activities, there are no other functionalities available to help the company automate other process. In a future the following functionalities were identified to improve their system:

- Develop an employee time entry system to record their working hours.
- Create payroll based on the entered working hours and the employee rate to facilitate the payroll process.

- Develop a payroll email notification once the payment has been processed.
- Develop mobile devices applications to access the Customer Tool instead of using a Web Browser.

REFERENCES

[1] C. Horstman, "The Object-Oriented Design Process" in *Object-Oriented Design & Patterns*, 2nd ed. NJ, USA: B.S, ch. 2, 2006, pp. 48-49.