

## **The Design And Development Of Live Chat Support System For Prossimo IT Solution Using Meteor**

**Nor Hanani Mohd Yusoff<sup>1\*</sup>, DaliniPriya A/P Selliah<sup>1</sup>**

<sup>1</sup>Department of Information and Communication Technology, Politeknik Ungku Omar, 31500 Ipoh, Perak

**\*Corresponding author E-mail:** norhanani@puo.edu.my

### **Abstract**

Customer Service is an integral part of any business. Small business owners mostly manage it themselves while medium and large scale businesses either hire in house or outsource to specialized companies. In today's fast paced environment and increased use of social media, users in general expect a business to act fast and offer instant support. To cope up with latest trends, it is important for a business to offer support to their customers through Live Chat. This paper presents the design and development of a live chat support system based on a case study at information technology (IT) company, Prossimo™ IT Solution. The live chat system was used to provide their customers immediate response for their most basic questions. This live chat system was developed using Meteor v1.1.2.1. The main programming languages used to develop this system are Javascript, J-Query, HTML5, and CSS3. Other software used to support the development system are MongoDB and Yeoman generator. Node JS is an event-driven I/O server-side JavaScript environment to enable the server side configurations throughout the development of the live chat system. Thus, this software's help in development of the live support system which will be stored in the git repository in order to guarantee the data is not lost. As a conclusion, the live chat support system is the first step in providing an alternative for Prossimo™ IT Solution to improve their first contact resolution with their respective customers.

**Key Words:** customer service; live chat; meteor; MongoDB

### **1.0 INTRODUCTION**

Customer service is the provision of service to customers before, during and after a purchase whereas customer support is a range of customer services to assist customers in making cost effective and correct use of a product. It includes assistance in planning, installation, training, trouble shooting, maintenance, upgrading, and disposal of a product. A good customer service is very important for businesses to maintain relationship with their clients (Meng, Cheng, Shi, & Martineau, 2017). Live Customer Support Chat represents a newly implemented type of customer service in E-commerce websites. Other than e-commerce websites, there are businesses that implement a live customer support as an added value to their business operation. Live Chat Support System allows online agents through company's websites to answer customer questions, complaints, and concerns on the spot through a live chat session, which bypasses the traditional e-mail and web forms. According to (Peters, V., & Singh, A., 2015), customer support can be provided for free or with additional charge and the services are delivered over phone or various kinds of method such as email, website or even chat window. Live chat can be used for prospective customers in order to help them with their buying decision or for current customers who need support with

a product they have already bought. According to a report “Why is Live Chat Preferred” (Graham Charlton, 2013), found that over 79% of respondents agree that they would get their questions answered immediately. Furthermore 46% of them agreed that having customer support through live chat system is the most efficient as compared to using application, email, mailing, social media, phone call and short messaging system.

There are a number of benefits that businesses could gain from having a customer live chat support system. Reported by (Ahmed Elmorshidy, 2011), the benefits of live chat support system can be viewed from three perspectives; multitasking, instant leads and better first impression. Multitasking means the customer support representative can entertain more than one client at the same time. Instant leads mean customer support representative are able to ask much details from customer with regard to their problems. Meanwhile better impression means customers satisfied with the live chat support system will stay and buy more products from the company. Live chat system has been study from various context. For example, (McLean, G., & Osei-Frimpong, K., 2017), conducted a study about the important elements that influence a customer’s satisfactions with live chat system. They conducted a web based survey with 302 respondents of real-live chat service experiences with mobile phone network providers in United Kingdom. According to the study, the important elements are service quality, information quality and system quality.

Hence, Prossimo™ IT Solutions is one of the companies that are interested in having a live chat system as one of the customer support services. Prossimo™ IT Solutions was established and incorporated in 2012 with its core business as a web and application developer. Main branch, Prossimo™ Technologies was created with the objective to provide its customer with easier access to daily IT needs, with service brought to customers’ doorstep. The first branch was opened in Bukit Jelutong, Shah Alam. Based on interview session with the Head of IT of Prossimo™ IT Solutions, currently they are using phone call as a method to entertain any complaint or question by their customers. They would like to embed live support system in the form of chatting window to their customers. It is hope that their customers will be able to get immediate response from customer support representative. From an interview session, some decision about live chat support system user scope has been define. Function in the live chat support system will be divided into two. One is for customer side and the other one is for customer support representative (CSR) of Prossimo™ IT Solutions.

The paper is structured as follows. The next section is a brief description of methodology that has been followed to develop the live chat support system. Next is the report of design and development. Followed with some capture of interface design. Lastly is the conclusion and suggestion for future improvement.

## 2.0 METHODOLOGY

In order to develop the live chat support system, a series of step has been taken. We choose agile method because it suits with the objective of the system that need to be develop. Agile method is the newest software development method and the system can be develop incrementally by building a series of prototype and constantly adjusting them according to user requirements (Rosenblatt, H. J.,2013). An agile method also emphasizes continuous feedback and each incremental step is affected by what was learned in the prior steps.

Firstly, we arranged an interview session with head of IT and customer support representative of Prossimo™ IT Solutions. The goal of the interview session was to get a clear understanding on the workflow of customer support representative, acquire user requirements and defining the system scope. Further steps taken was determined the software and hardware that will be used to develop the system. Summarization and deliverables for each step is depicted in Table 2.

**Table 2:** Step and deliverables of live chat support system for Prossimo™ IT Solutions

Step	Deliverables
Interview session with Head of IT and customer support representative of Prossimo™ IT Solutions.	Understanding of workflow of customer support representative System Scope
Determine software and hardware specification	Software and hardware specification
Create database	Database of live chat customer support system
Design interface	Interface of live chat customer support system
Conduct testing	Testing result of system functionality

## 3.0 SYSTEM DESIGN AND DEVELOPMENT

### 3.1 Scope of the system

From the discussion and interview question with the Head and Customer Support Representative of Prossimo™ IT Solutions, system scope and user scope have been identified. User and system scope of live chat system are described in Table 3.

**Table 3:** User and system scope

User	Scope
Customer Support Representative (CSR) of Prossimo™ IT Solutions	<ul style="list-style-type: none"> <li>• CSR will be able to login to the chat system with their Domain ID and password.</li> <li>• CSR will be able to view the online visitors on</li> </ul>

---

Customer of Prossimo™ IT Solutions	<p>that day.</p> <ul style="list-style-type: none"> <li>• CSR will be able to chat with more than one visitor at a time and help solve their enquiries.</li> </ul> <ul style="list-style-type: none"> <li>• Users will be able to click a floating button on the company’s website.</li> <li>• Users will be able to add their information such as name and email address in the form provided.</li> <li>• Users will be able to chat with the customer service representatives after filling in the form.</li> <li>• Users will be able to return to their previous chat as long as the email address they have used previously is the same.</li> </ul>
------------------------------------	--

---

### 3.2 Software Requirements

To develop the live chat support system, some software has been identified. The list of software that has been used to develop the live chat support system is as described in table 4.

**Table 4:** List of software

Software	Function
Meteor	Allows for rapid prototyping and produces cross-platform (Android, iOS, Web) code. It integrates with MongoDB and uses the Distributed Data Protocol and publish–subscribe pattern to automatically propagate data changes to clients without requiring the developer to write any synchronization code.
Atom text editor	A text editor that acts as a framework that enables cross-platform desktop applications using Chromium and Node.js.
Yeoman Generator	A generic scaffolding system allowing the creation of any kind of app. It allows for rapidly getting started on new projects and streamlines the maintenance of existing projects.
MongoDB	A cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. Supports field, range queries, regular expression searches.
NodeJS	For developing a diverse variety of tools and applications. It allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle core functionality.
GitLab	GitLab is a web-based Git repository manager with wiki and issue tracking features, using an open source license, developed by GitLab Inc.

---

MangoDB is considered as a relatively new NoSQL database management and has been used by various projects. NoSQL can support a huge web-scale application as compared

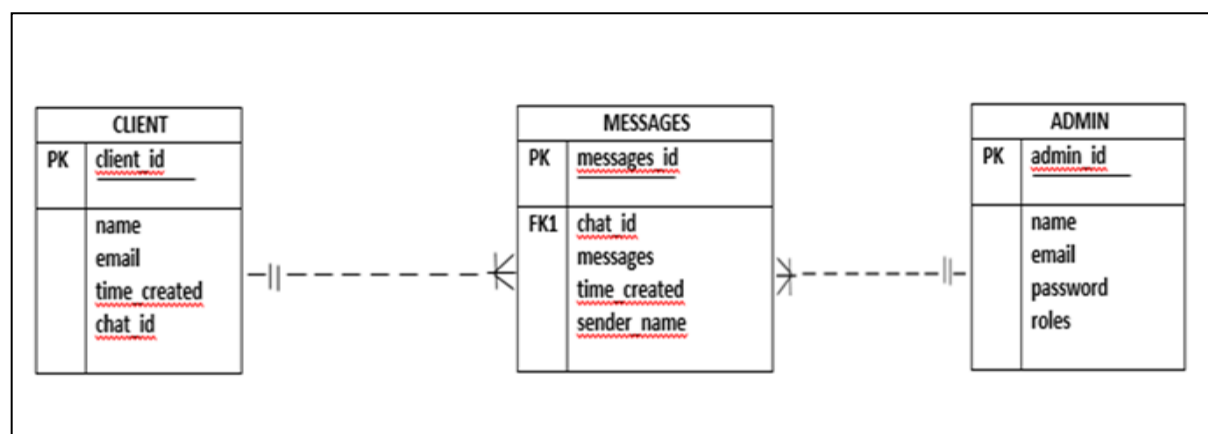
to traditional relational databases. A study by (Boicea, A., Radulescu, F., & Agapin, L. I., 2012) has summarized projects that used MangoDB as their database platform. List of the projects and functionality of MangoDB are as listed in Table 5.

**Table 5:** List of projects that used MangoDB as database platform

Name of the project	Functionality of MangoDB
Craigslist	Archiving billions of records
MTV Networks	Main repository for MTV Networks
SourceForge	Back-end storage
The Guardian	Store document
The New York Times	Form building application for photo submission
bit.ly	Store user history

### 3.3 Entity Relationship Diagram

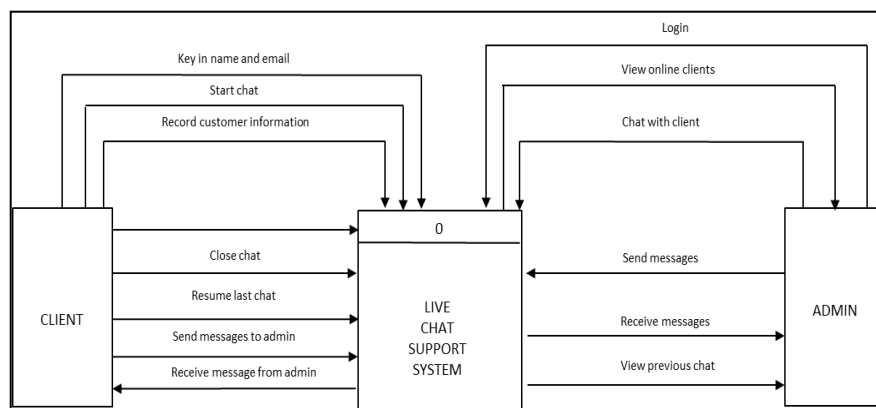
The design of the system is started with the development of Entity Relationship Diagram (ERD). ERD is a data modelling technique, which is a graphical representation of logical and detailed information on the entities and relationships between entities in an organization or business area. Relationship between one entity to another entity is necessary to demonstrate the functions of the entity by another entity. Attributes are the items that display the characteristics of an entity. As depicted in Figure 1, ERD for live chat support system involved three entities which are client, messages and admin. The client id has been assigned as a primary key of client's entity. Other attributes in client's entity are name, email, time of chat created and chat id. Second entity that has been identified is messages. In this entity, message id has been assigned as a primary key. Other attributes in messages' entity are chat id, messages, time of chat created and name of the sender. Chat id is assigned as a foreign key. Type of relationship between client and messages entity is one-to-many. The third entity involved is admin. Attributes in admin's entity are admin id, name, email, password and roles. Admin id has been assigned as primary key for admin's entity. Relationship between messages and admin entity is many to one.



**Figure 1:** Entity Relationship Diagram of Live Chat Support System

### 3.4 Context Diagram

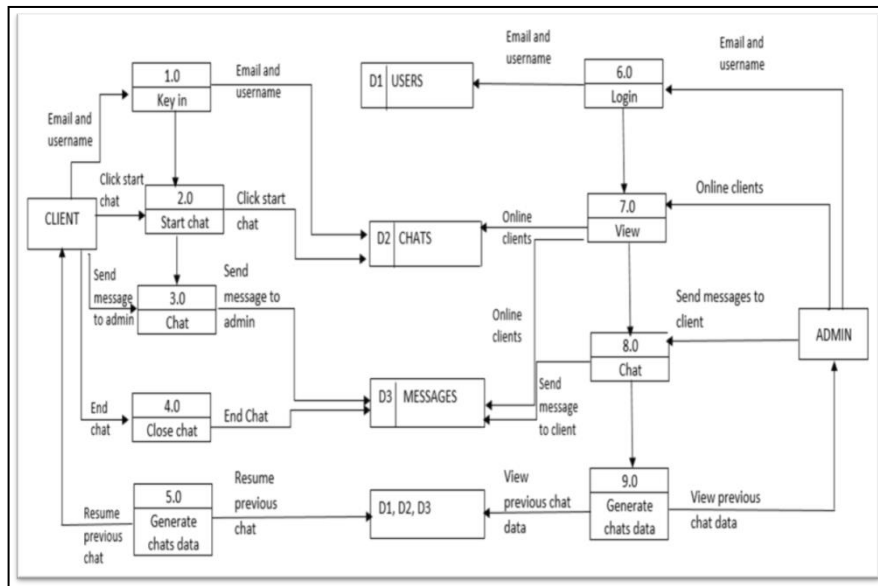
The context diagram as shows in Figure 2 describes the boundaries and scope of the system and how the system interacts with the external environment. Context diagram demonstrate high level view of a live chat support system. Data will be stored in client and admin. The processes involved between client and the live chat support system are key in name and email, start chat, record customer information, close chat, resume last chat and send messages to admin. Meanwhile live chat support system will forward a message that has been received from admin to client. The processes that will occur between admin and live chat support system are login, view online clients, chat with client and send messages. Furthermore, the live chat support system will forward messages that have been received and view previous chat to admin.



**Figure 2:** Context Diagram of Live Chat Support System

### 3.5 Data Flow Diagram

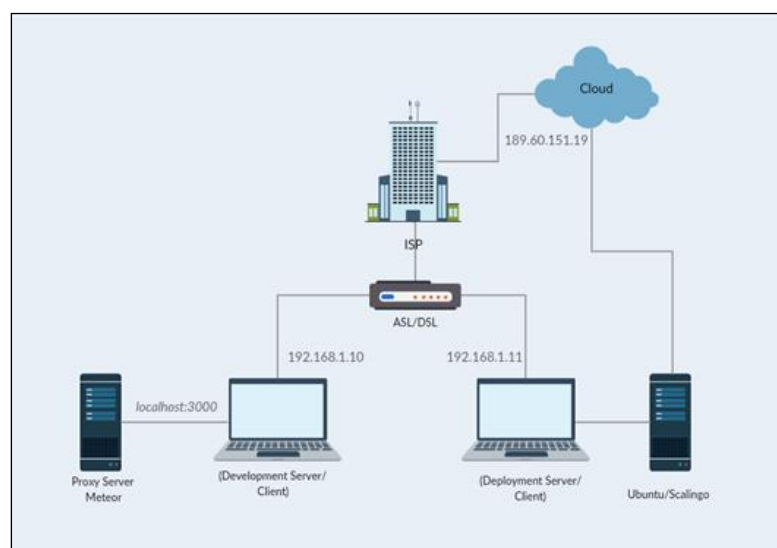
Data flow diagram describes the process that happened in live chat support system. The data flow diagram contains nine processes and the two entities, namely client and admin. Administrators can make all the processes such as log in, view online clients, send messages to client, receive messages from client and generate previous chat with clients. Clients will be able to enter their name and email in a form provided. Clients will also be able to send messages to admin and receive messages from admin too. Clients can then end the chat session and may resume their previous chat with admin by entering their name and email again. Figure 3 shows the level of a data flow from the system context.



**Figure 3: Data Flow Diagram**

### 3.6 Logical Network Diagram

A logical network is one that appears to the user as a single, separate entity although it might in fact be either an entity created from multiple networks or just a part of a larger network. A logical network is defined by its IP addressing scheme. As for this live chat support system, the system is logically connected by two ways. First is through the local host configuration that works within the development server. The development server connection is established through node JS configuration in meteor proxy server. Second connection is live connection to the deployment server. The connection is secure as the SSH key is added during the deployment of the project. Figure 4 shows the setup for logical network diagram.



**Figure 4: Logical Network Diagram**

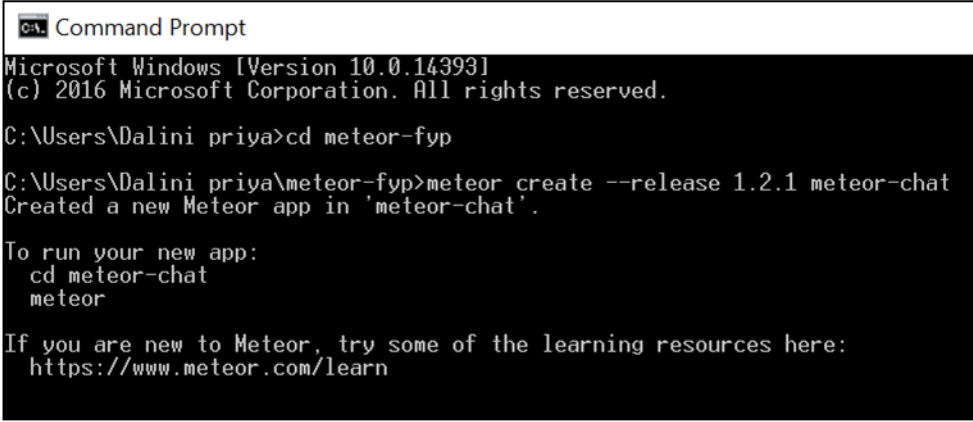
### 3.7 Development Phase

Development of the live chat support system started with the installation of earlier defined software. The sequence of software that need to be install are defined earlier in order to ensure the system working fine. The first software installed is Meteor. It contains a full stack JavaScriptsframework and hold a collection of libraries and packages. This feature help in the coding phase of live chat support system. Second software installed is MangoDB. It is an open source database that uses a document oriented model. Third software is Yeoman Generator that basically works as a plugin and can be run with the ‘yo’ commands to scaffold complete system or certain module. Fourth software is Node JS as a platform built in Chrome’s java script runtime. The advantages of node.js are it uses an event-driven, non-blocking I/O model that makes it lightweight and efficient for data-intensive real-time applications.

#### 3.7.1 Software Installation Phase

##### i. Meteor Installation

Download the meteor installer and enter command prompt. To install version 1.1.2.1 of meteor meteor create –release 1.2.1 (app name) is entered.



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Dalini priya>cd meteor-fyp

C:\Users\Dalini priya\meteor-fyp>meteor create --release 1.2.1 meteor-chat
Created a new Meteor app in 'meteor-chat'.

To run your new app:
  cd meteor-chat
  meteor

If you are new to Meteor, try some of the learning resources here:
  https://www.meteor.com/learn
```

**Figure 5:** Meteor Installation

##### ii. MongoDB Installation

MongoDB is integrated in meteor from the previous installation. To install mongol; MongoDB app editor, enter command prompt and meteor add msavin:mongol is keyed in.



```
C:\Users\Dalini priya\meteor-fyp\meteor-chat>meteor add msavin:mongol

Changes to your project's package version selections:

dburles:mongo-collection-instances  added, version 0.3.4
email                                added, version 1.0.6
blai:collection-extensions           added, version 0.1.4
meteor:toys:toykit                   added, version 1.0.4
msavin:mongol                        added, version 1.2.0

msavin:mongol: The insanely handy development package for Meteor.
```

**Figure 6: MongoDB Installation**

### iii. Yeoman Generator Installation

Yeoman generator can be installed through “npm” package or by manually enter command ‘yo’. In order to install Yeoman Generator, we need to open command prompt and change directory to the location of meteor app. After that, enter command ‘*yo meteor*’. Once yeoman is prompted, choose whether to install all the services provided by Yeoman Generator.

```
C:\ Command Prompt

C:\Users\Dalini priya\meteor-fyp\meteor-chat>yo meteor

  Welcome to Yeoman,
  ladies and gentlemen!

  Shall we include Iron Router? Yes
  Shall we include Bootstrap? Yes
  create client\client.js
  create client\lib\subscriptions.js
  create client\views\home.js
  create client\views\home.html
  create client\views\common\loading.html
  create lib\collections.js
  create public\robots.txt
  create server\publications.js
  create server\server.js
  create server\security.js
  conflict .meteor\gitignore
  Overwrite .meteor\gitignore? overwrite
  force .meteor\gitignore
  conflict .meteor\release
  Overwrite .meteor\release? overwrite
  force .meteor\release
  create .gitignore
  create .jshintrc
  create .travis.yml
  create .editorconfig
  create LICENSE
  create README.md
  create client\routes.js
  create client\views\layout.html
  create client\styles\theme.css
  conflict .meteor\packages
  Overwrite .meteor\packages? overwrite
  force .meteor\packages
```

**Figure 7: Yeoman Generator Installation**

#### iv. NodeJs Installation

NodeJs and npm are simultaneously installed by the Node js installer. In order to update to a certain version of NodeJs, open command prompt and enter command ***npm install latest-g***. This action will automatically prompt the global configuration and installation.

#### v. Meteor Project Creation

Once meteor app is created in the directory, open command prompt and change directory to the location of meteor app. Then, enter command '***meteor***' to start-up the proxy, mongoDB and the app in the specified localhost.

```
C:\Users\Dalini priya\meteor-fyp\meteor-chat>meteor
[!!!! C:\Users\Dalini priya\meteor-fyp\meteor-chat !!!!!]

=> Started proxy.
=> Started MongoDB.
=> Started your app.

=> App running at: http://localhost:3000/
    Type Control-C twice to stop.
```

**Figure 8: Meteor Start App**

### 3.7.2 Chat Development Phase

The chat development's code is placed in a separate folder called partials. The chat.htm and chat.js is later called in the homepage.html with `{{>chat}}`, Thus, enabling it to function as a plug-in if necessary and makes it easier as the whole chat area is not embedded within the homepage.html. The chat.html consists the `<form >` within that enables the clients to fill up the given form. The CSS3 as well as the JavaScript inclusive j-query coding enable the functions display and animation of the chat area.

### 3.8 Interface Design of Live Chat Support System

The following are some screenshot of interface design of the system. Figure 9 depicted the homepage of live chat support system.

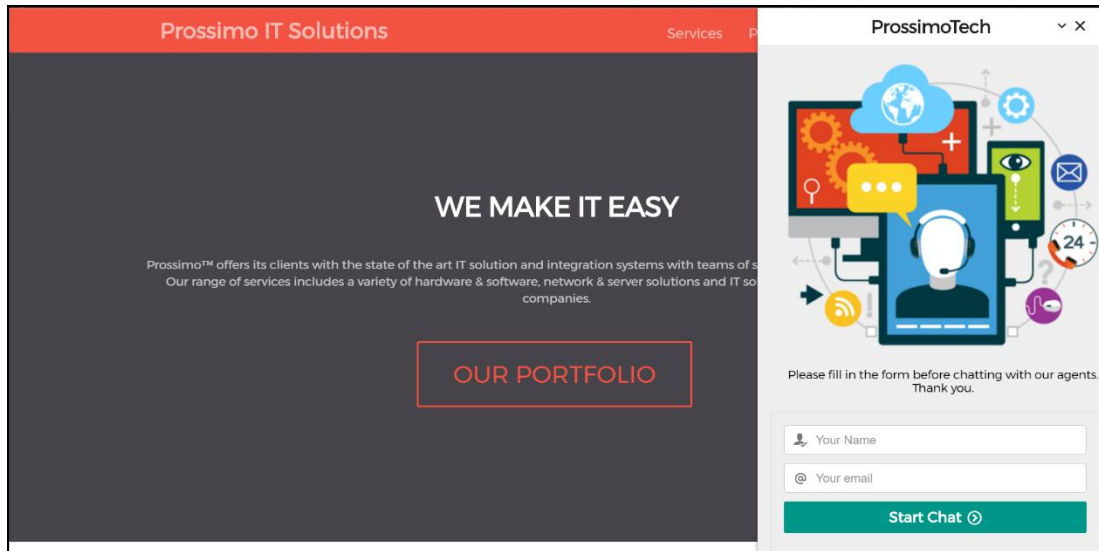


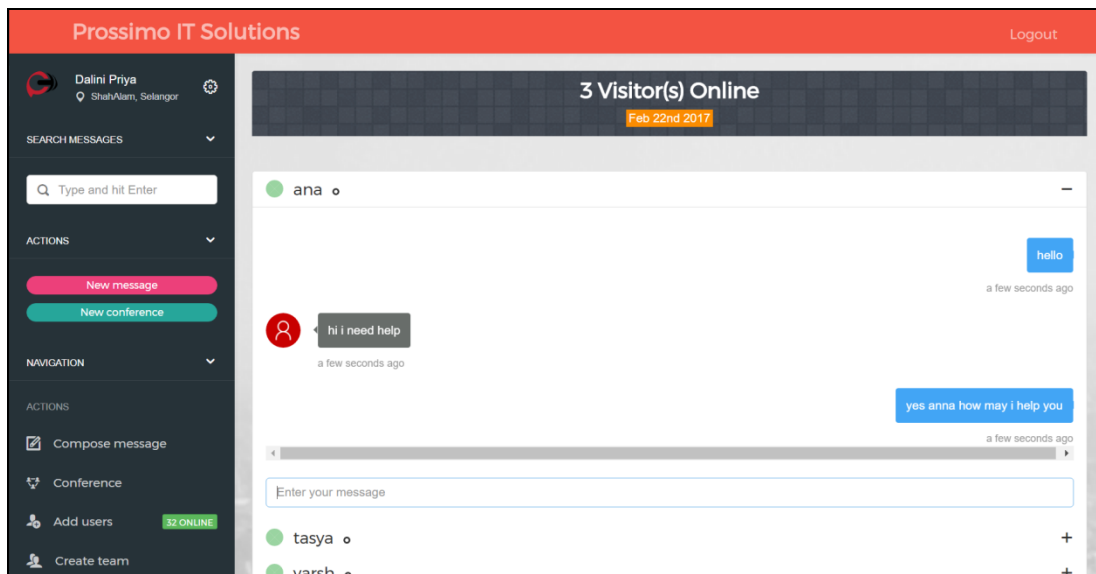
Figure 9: Homepage Chat

Figure 10 shows testing of chat message between customer and customer support representative.



Figure 10 : Chat communication testing

The interface of the live chat support system from customer representative view is depicted in Figure 11.



**Figure 11:** Customer Representative Staff page

#### 4.0 TESTING PHASE

Some testing has been carried out to test the usability and functionality of the live chat system. The testing has been done by Head of IT and Customer Service Representative. During the testing, users are given a set of task and the result has been recorded. Table 6 shows testing task for Customer Service Representative login task testing and the result.

**Table 6:** CSR login task testing

Test Case Name	Test Procedure	Pre-condition	Expected Result	Result
Domain ID	1. The CSR's login ID field is left blank and enter is pressed.  2. The valid username or email address field is fill and enter is pressed.	Incorrect Login ID or password.	1. An error message must be prompted by login form instructing the user to type in the username or email address.  2. Successful login.	Successful
Password	1. The password field is left blank and Login button is clicked.  2. The valid password field is fill and Login button is clicked.	Incorrect Login ID or password.	1. An error message must be prompted by login form instructing the user to type in the user's password.  2. Successful login.	Successful

The testing case name is given to Prossimo™ IT Solutions’s client and as well as to the admin or customer support representative team. The main goal of the testing phase is to test the functionality of live chat support system. Before the testing task started, they are given a short briefing and explanation about the task that need to be done. Three minutes are allocated for them to complete the task. Detail of the task and the corresponding result is shown in Table 7.

**Table 7:** Testing task and result for chat functionality

Test Case Name	Test Procedure	Expected Result	Result
Prossimo™ Client	1. The client click the chat button and chat box will pop out	1. Client able to click the chat button and chat box popped out.	Successful
	2. The client click on the text box and send text message.	2. Client able to send text message.	
Admin/CSR	1. The admin can see the online visitor.	1. Admin able to see online visitors.	Successful
	2. The admin can click the chat box and start to chat with client.	2. Admin able to click the chat box and send messages.	

## 5.0 DISCUSSION AND SUGGESTION FOR FUTURE WORKS

With the advent of technology as well as interactive electronic communications and the Internet, customer support system also has become advanced. Previously, customer support services are handled through email or a phone call to service centre. Nowadays, companies have an opportunity to subscribe live chat support services offered by various third party. Among the services are Pure Chat, LiveHelpNow, LiveEngage by LivePerson, My LiveChat, REVE Chat, ClickDesk, WhosOn, Olark and ZohoSalesIQ (Sara Angelas, 2017). Even though the services offered are bundled with extra functionality, the live chat system is controlled by the third party equipment's and infrastructure. This situation could lead to security breach in terms of capturing chat conversation and customer data. Thus, Prossimo™ IT Solutions has taken their initiative to design and develop the live chat support system locally.

However, each system that has been developed locally has its own advantages and disadvantages. There are some improvements that can be done to improve the system. Among the recommendations are integrate the live chat support system with the existing ticketing system at Prossimo™ IT Solutions. The live chat support system may also be added with other modules such as visitor analytics, customization options and integrate with mobile application. In term of this study, the project can be expand with measurement of the effectiveness of live chat support system and conduct a study on user experience through live chat support system. Furthermore, to make the system more secure, the company may upgrade their network with Virtual Private Network (VPN).

## 6.0 CONCLUSION

This paper has presented the design and development of live chat support system based on a case study at Prossimo™ IT Solutions. According to result from testing phase, it can be concluded that the development of live chat support system for Prossimo™ IT Solutions has achieved the main objective. Instead of using phone call to log a report, their customers can have a live chat session with customer service representative.

## 7.0 REFERENCES

- Ahmed Elmorshidy, (2011), *Benefits Analysis of Live Customer Support Chat in E-Commerce Websites: Dimensions of a New Success Model for Live Customer Support Chat*
- Banker, Kyle (2011). *MongoDB in action*. Manning Publications Co., 2011.
- Boicea, A., Radulescu, F., & Agapin, L. I. (2012, September). MongoDB vs Oracle--database comparison. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on* (pp. 330-335). IEEE.

- Chodorow, K. (2013). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. " O'Reilly Media, Inc."
- Gao, J., Modak, M., Dornadula, S., & Shim, S. (2004, July). *Mobile jabber im: a wireless-based textchatting system*. In e-Commerce Technology, 2004. CEC 2004. Proceedings. IEEE International Conference on (pp. 337-341). IEEE.
- Graham Charlton (2013), Consumers prefer live chat for customer service: stats. Retrieved August 21, 2017 from <https://econsultancy.com/blog/63867-consumers-prefer-live-chat-for-customer-service-stats>
- Hackos, J. T., & Redish, J. (1998). *User and task analysis for interface design*.
- Livermore, J. A. (2008). *Factors that Significantly Impact the Implementation of an Agile Software Development Methodology*. *JSW*, 3(4), 31-36.
- McLean, G., & Osei-Frimpong, K. (2017). *Examining satisfaction with the experience during a live chat service encounter-implications for website providers*. *Computers in Human Behavior*, 76, 494-508.
- Meng, Yunsong, Cheng, Doreen, Shi, Yongmei, & Martineau, Justin. (2017). *Assistant redirection for customer service agent processing*: Google Patents.
- Peters, V., & Singh, A. (2015). U.S. Patent No. 8,943,145. Washington, DC: U.S. Patent and Trademark Office.
- Rosenblatt, H. J. (2013). *Systems analysis and design*. Cengage Learning.
- Sara Angelas (2017). 9 Live Chat Solutions for Small Businesses. Retrieved October 10, 2017 from <http://www.businessnewsdaily.com/7025-live-chat-solutions.html>
- Sunner, D. (2016, July). *Agile: Adapting to need of the hour: Understanding Agile methodology and Agile techniques*. In Applied and Theoretical Computing and Communication Technology (iCATccT), 2016 2nd International Conference on (pp. 130-135). IEEE.
- Tilkov, S., & Vinoski, S. (2010). *Node.js: Using JavaScript to build high-performance network programs*. *IEEE Internet Computing*, 14(6), 80-83.