

|  |
| --- |
| **SyneMobiSafe – Advanced Tech Support with Live Chat**  Vishal Bhardwaj, Assistant Manager – Software  May 2017 |
|  |

Table of Contents

[Introduction 3](#_Toc478485824)

[Document Purpose 3](#_Toc478485825)

[Current State 3](#_Toc478485826)

[Future State 4](#_Toc478485827)

[Development Environment Setup 4](#_Toc478485828)

[ Configuration: New DB Schema 4](#_Toc478485829)

[ Configuration: SyneMobiSafe.xmlFile](#_Toc478485830) 4

[Code Changes 5](#_Toc478485831)

[ DAO (Implementation) Class 5](#_Toc478485834)

[ Module(Chatbot Implementation) Class 5](#_Toc478485835)

[Enhancement Impact 7](#_Toc478485836)

[ Conclusion 7](#_Toc478485837)

# Introduction

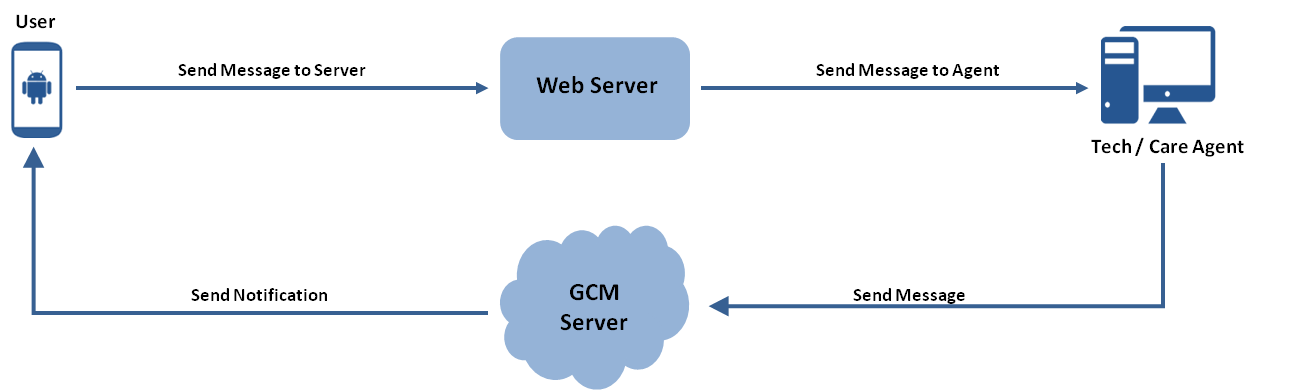
The Advanced Tech support is one of the core module of this tool which provides flexibility to the users to opt for instant support or live chat for the issues they are facing. As of now, the system routes all the queries from the customer directly to the tech agents, who assist and provide desired solutions; however, this requires a lot of manual interactions and tech agents needs to be deployed 24x7. The module needs to be enhanced in such a way that it should be able to resolve the queries with minimum manual interactions by integrating some extra process.

# Document Purpose

The purpose of this document is to introduce a new mechanism which needs to be integrated to the system in order to minimize manual efforts in providing efficient solutions to the users seeking assistance. The document provides the details of the enhancements and implementation techniques analyzed for the SyneMobiSafe Advanced Tech Support with Live Chat Module. The document also provides the impact analysis, details of code and configuration.

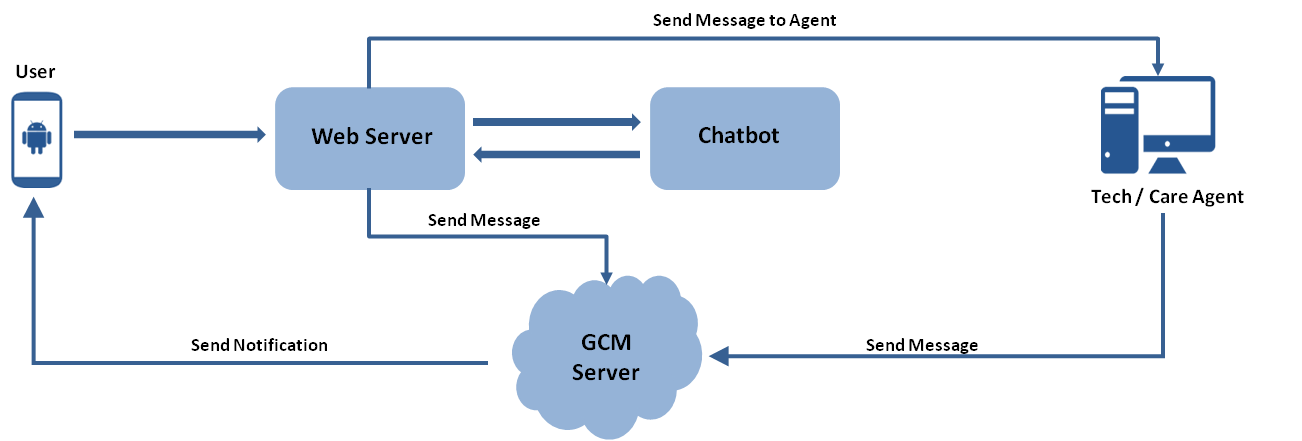
# Current State

As per current enhancement and implementation, the messages coming from the users are directly routed to the agents (Please refer below current design).



# Future State

The proposed enhancement in the live chat module needs to be updated and integrated with a new messaging system. To perform this enhancement, a new mechanism will be designed; called a chatbot, that will help minimize manual interactions and will save a lot of man hours. (Please refer below proposed design).



# Development Environment Setup

This section provides the details for setting up development environment to update the module and its upgrade mechanism. The following configurations are required to be implemented to enhance and upgrade the module:

### Configuration: New DB schema

The database needs to be configured with the new messaging bots details. The new schema needs to be created at the server will be like -

|  |  |
| --- | --- |
| Column Name | Column Type |
| UserID | Number |
| BotID | Number |
| MsgID | Number |
| Message | String |

### Configuration: SyneMobiSafe.xml File

The SyneMobiSafe.xml files needs to be created and configured as follows –

Custom Configurations are declared in this section and these values could be overridden from the values declared in class files.

<CustomConfig>

<ResponseTimeout>90000</ResponseTimeout>

<ChatSorurce>Bot</ChatSorurce >

<QueueMessages>1</QueueMessages>

<SSL>true</SSL>

<SyncFlow>true</SyncFlow>

</CustomConfig>

# Code Changes

### DAO(Implementation Class)

A background running job will fetch the new chatbot instance and update the db with the incoming message details. Following is one of the sample model classes that can provide the implementation–

**public class BotDAO {**

**private Integer userId;**

**private Integer botId;**

**private Integer msgId;**

**private String message;**

**public Integer getUserId() {**

**return userId;**

**}**

**public void setUserId(Integer userId) {**

**this.userId = userId;**

**}**

**public Integer getBotId() {**

**return botId;**

**}**

**public void setBotId(Integer botId) {**

**this.botId = botId;**

**}**

**public Integer getMsgId() {**

**return msgId;**

**}**

**public void setMsgId(Integer msgId) {**

**this.msgId = msgId;**

**}**

**public String getMessage() {**

**return message;**

**}**

**public void setMessage(String message) {**

**this.message = message;**

**}**

**}**

### Module(Chatbot Implementation) Class

Following is one of the Chat Module class that can be used as per the implementation –

public class ChatModule implements ChatDelegateAPI {

Activity context;

private static ChatModule mChatModule;

public static ChatModule getChatModuleInstance() {

if (null == mChatModule) {

mChatModule = new ChatModule();

}

return mChatModule;

}

private ChatModule() {

ChatServiceFactory.getInstance().setDelegateAPI(this);

}

public void getChatBotInstance() {

ChatServiceFactory.getInstance().getBotInstance (this);

}

public void startChat(Activity context, String lastQuestion) {

this.context = context;

Map<String, Object> customVariables = new HashMap<String, Object>();

customVariables.put(VISITOR\_ID, userID);

customVariables.put(MOBILE\_NUMBER, mobileNumber);

customVariables.put(USER\_NAME, Name);

String account\_id = preferences.getChatID();

beginChat(context, account\_id,);

setCustomVariables(customVariables);

}

@Override

public void onAgentAvailabilityChanged(boolean arg0) { }

@Override

public void onEndChat(boolean arg0) { }

@Override

public void onEvent(String arg0, Map<String, Object> arg1) { }

@Override

public void onHideChatButton() { }

@Override

public void onNewUnreadAgentMessage(int arg0) { }

@Override

public void onShowChatButton() { }

}

# Enhancement Impact

As mentioned earlier, the live chat functionality needs to be updated and integrated with a new mechanism i.e. a chatbot to provide timely and accurate responses. Following are the main advantages of the chatbot implementation.

* **Well Designed:** The computer program is designed to fit in between the server and agent portal and intercept the incoming messages from the user and if it finds a suitable match(as per the configuration), it will reply back with proper response without any manual help. In case it doesn't find any match, it will route it back to the server from where it will be re-directed to the agent portal for manual interaction.
* **More Productive:** The implementation minimizes the manual efforts and save time and man hours, hence the system is more efficient and the productivity goes up.

### Conclusion

Based on the above analysis, we can conclude that, implementing the chatbot feature to the SyneMobiSafe will help in assisting users in a more efficient manner and is not time consuming. We can also leverage the multi-lingual support feature to the chatbot and improve the performance of the SyneMobiSafe.