



Student name:	Student 1: Abdulqadoos malik				
	Student 2:				
	Student 3:				
Student number:	Student 1: 3091035				
	Student 2:				
	Student 3:				
Faculty:	Computing Science				
Course:	BSCH/BSCO/EXCH	Stage/year:	2		
Subject:	Software Development 2				
Study Mode:	Full time	<input checked="" type="checkbox"/>		Part-time	
Lecturer Name:	Gemma Deery				
Assignment Title:	Review 3				
Date due:	27/04/2023				
Date submitted:					

**Plagiarism disclaimer:**

*I understand that plagiarism is a serious offence and have read and understood the college policy on plagiarism. I also understand that I may receive a mark of zero if I have not identified and properly attributed sources which have been used, referred to, or have in any way influenced the preparation of this assignment, or if I have knowingly allowed others to plagiarise my work in this way.*

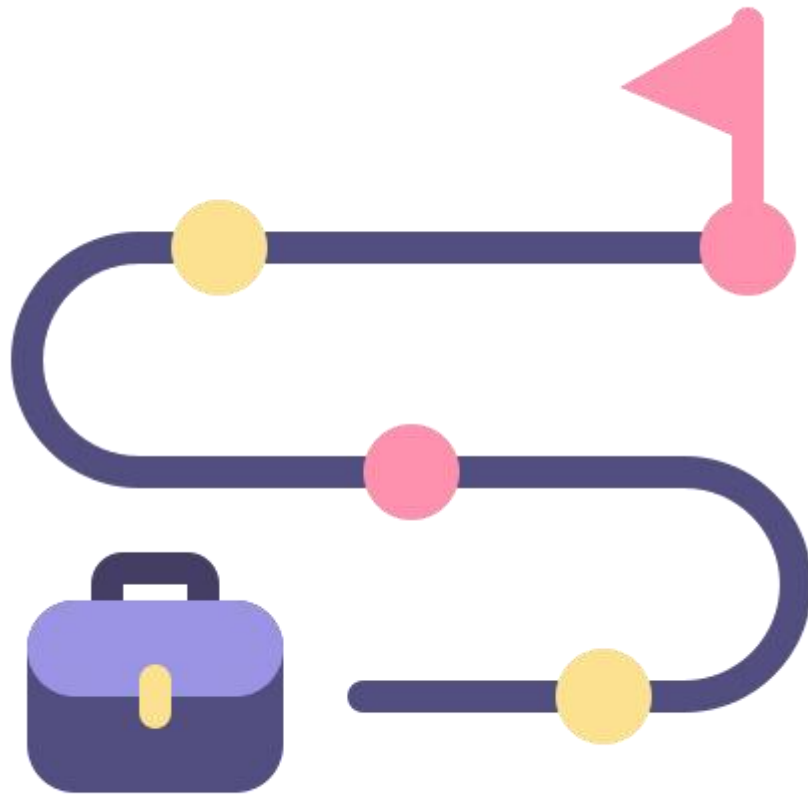
*I hereby certify that this assignment is my own work, based on my personal study and/or research, and that I have acknowledged all material and sources used in its preparation. I also certify that the assignment has not previously been submitted for assessment and that I have not copied in part or whole or otherwise plagiarised the work of anyone else, including other students.*

Signed: \_\_\_\_\_

Date: \_\_\_\_\_

**Please note:** Students **MUST** retain a hard / soft copy of **ALL** assignments as well as a receipt issued and signed by a member of Faculty as proof of submission.

# Milestone Report



Student 1 name:	Student 1 number:
Student 2 name:	Student 2 number:
Student 3 name:	Student 3 number:

<b>VERSIONING APPROACH</b>	<b>4</b>
<b>DEVELOPMENT APPROACH</b>	<b>4</b>
<b>MILESTONE 1</b>	<b>4</b>
<i>Goals</i>	4
<i>Junit Tests</i>	4
<i>Commit Logs</i>	4
<i>Full Log Details</i>	4
<b>MILESTONE 2</b>	<b>4</b>
<i>Goals</i>	4
<i>Junit Tests integration</i>	4
<i>Commit Lists and Branches Trees</i>	4
<i>Full Log Details</i>	4
<b>MILESTONE 3</b>	<b>4</b>
<i>Goals</i>	4
<i>Junit Tests integration</i>	4
<i>Commit Lists and Branches Trees</i>	4
<i>Full Log Details</i>	4
<b>COLLABORATION</b>	<b>4</b>
<b>APPENDIX</b>	<b>9</b>

## Versioning Approach

## Development Approach

### Milestone 1

Goals

JUnit Tests

Commit Logs

Full Log Details

### Milestone 2

Goals

JUnit Tests integration

Commit Lists and Branches Trees

Full Log Details

### Milestone 3

Goals

JUnit Tests integration

Commit Lists and Branches Trees

Full Log Details

## Collaboration

Please list the contributions made by each student for this iteration.

- Each student is expected to make equal contributions
- Each student will be marked according to his/her contribution
- You must have relevant comment on all code and messages on all commits

**Student 1:**

```
package firstpackage;

import javax.swing.*;
import javax.swing.text.BadLocationException;

import java.awt.*;
import java.awt.event.*;

public class GUIchatbot extends JFrame {

    private JPanel chatPanel;
    private JTextField userInputField;
    private JTextArea chatbotMessages;
    private JButton sendButton;
    private JButton clearButton;
    private JScrollPane scrollPane;
    private static final int MAX_MESSAGES = 100;

    public GUIchatbot() throws BadLocationException {
        // Set up the chat panel
        chatPanel = new JPanel();
        chatPanel.setLayout(new BorderLayout());
        chatbotMessages = new JTextArea();
        chatbotMessages.setEditable(false);
        scrollPane = new JScrollPane(chatbotMessages);
        userInputField = new JTextField();
        sendButton = new JButton("Send");
        clearButton = new JButton("Clear");
        JPanel inputPanel = new JPanel(new BorderLayout());
        inputPanel.add(userInputField, BorderLayout.CENTER);
        inputPanel.add(sendButton, BorderLayout.EAST);
        inputPanel.add(clearButton, BorderLayout.WEST);
        chatPanel.add(scrollPane, BorderLayout.CENTER);
        chatPanel.add(inputPanel, BorderLayout.SOUTH);

        // Set up the main window
        setTitle("ChatBot");
    }
}
```

```
chatPanel.add(inputPanel, BorderLayout.SOUTH);

// Set up the main window
setTitle("ChatBot");
setSize(500, 500);
setDefaultCloseOperation(EXIT_ON_CLOSE);
add(chatPanel);

// Add action listeners
sendButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        sendMessage();
    }
});

clearButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        clearMessages();
    }
});

userInputField.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        sendMessage();
    }
});

// Add welcome message
appendMessage("Welcome to ChatBot!");

// Set focus to user input field
userInputField.requestFocus();
}
```

```

private void sendMessage() throws BadLocationException {
    String userInput = userInputField.getText().trim();
    if (!userInput.isEmpty()) {
        // Call the chatbot's method to generate a response
        String chatbotResponse = ChatBot.generateResponse(userInput);
        // Append the user's message to the chat history
        appendMessage("You: " + userInput);
        // Append the chatbot's response to the chat history
        appendMessage("ChatBot: " + chatbotResponse);
        // Clear the user input field
        userInputField.setText("");
        // Set focus to user input field
        userInputField.requestFocus();
    }
}

private void clearMessages() throws BadLocationException {
    chatbotMessages.setText("");
    appendMessage("Chat history cleared.");
}

private void appendMessage(String message) throws BadLocationException {
    if (chatbotMessages.getLineCount() >= MAX_MESSAGES) {
        // Remove the oldest message to keep the chat history from getting too long
        int end = chatbotMessages.getLineEndOffset(0);
        chatbotMessages.replaceRange("", 0, end);
    }
    chatbotMessages.append(message + "\n");
    // Scroll to the bottom of the chat history
    JScrollBar vertical = scrollPane.getVerticalScrollBar();
    vertical.setValue(vertical.getMaximum());
}

public static void main(String[] args) throws BadLocationException {

```

```

package firstpackage;

public class ChatBot {
    public static String generateResponse(String userInput) {
        String response = "";
        // Logic to generate response based on user input
        if (userInput.toLowerCase().contains("hello") || userInput.toLowerCase().contains("hi")) {
            response = "Hi There!";
        } else if (userInput.toLowerCase().contains("how are you")) {
            response = "I'm doing well, thanks for asking.";
        } else if (userInput.toLowerCase().contains("what is your name")) {
            response = "My name is ChatBot.";
        } else if (userInput.toLowerCase().contains("who is ian")) {
            response = "he is a legend";
        }
        else {
            response = "I'm sorry, I don't understand. Can you please rephrase?";
        }
        return response;
    }
}

```

2<sup>nd</sup> try



```
package firstpackage;

import javax.swing.*;

public class GUIchatbot extends JFrame {

    private JPanel chatPanel;
    private JTextField userInputField;
    private JTextArea chatbotMessages;
    private JButton sendButton;

    public GUIchatbot() {
        // Set up the chat panel
        chatPanel = new JPanel();
        chatPanel.setLayout(new BorderLayout());
        chatbotMessages = new JTextArea();
        chatbotMessages.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(chatbotMessages);
        userInputField = new JTextField();
        sendButton = new JButton("Send");
        chatPanel.add(scrollPane, BorderLayout.CENTER);
        JPanel inputPanel = new JPanel(new BorderLayout());
        inputPanel.add(userInputField, BorderLayout.CENTER);
        inputPanel.add(sendButton, BorderLayout.EAST);
        chatPanel.add(inputPanel, BorderLayout.SOUTH);

        // Set up the main window
        setTitle("ChatBot");
        setSize(500, 500);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        add(chatPanel);

        // Add action listener to the send button
        sendButton.addActionListener(new ActionListener() {
            @Override
```



```
ot.java  module-info...  ChatBot.java  GUIChatBot.java  ChatBot.java  README.md  package-info...
import javax.swing.*;
import java.awt.*;

// Add the chat panel to the main window
inputPanel.add(sendButton, BorderLayout.EAST);
chatPanel.add(inputPanel, BorderLayout.SOUTH);

// Set up the main window
setTitle("ChatBot");
setSize(500, 500);
setDefaultCloseOperation(EXIT_ON_CLOSE);
add(chatPanel);

// Add action listener to the send button
sendButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String userInput = userInputField.getText();
        // Call the chatbot's method to generate a response
        String chatbotResponse = ChatBot.generateResponse(userInput);
        // Append the chatbot's response to the chat panel
        chatbotMessages.append("You: " + userInput + "\n");
        chatbotMessages.append("ChatBot: " + chatbotResponse + "\n");
        // Clear the user input field
        userInputField.setText("");
    }
});

public static void main(String[] args) {
    GUIChatBot chatBot = new GUIChatBot();
    chatBot.setVisible(true);
}
```

Student 2:

Student 3:

## Appendix