Phase 5: Operation and Final Submission

***Note:** It is expected that all members of the group will make a reasonable contribution in writing parts of this report, that way one group member does not get overstretched. Furthermore, it ensures all members get practice of writing technical reports.*

Operation is a phase where the developed software system will be used by the customer and end users. Therefore, a manual must be written to describe the necessary procedures to install, run and demonstrate the working system.

The final deliverable includes phases 1-5, entire source code and two appendices, i.e., scrum meeting reports and presentation slides.
Your phases 1-5 deliverable should be updated to reflect any new user stories added, changes to the classes, changes to your database, and testing.
So, I expect four deliverables:

(1) ***Phases 1-5***
(2) ***Entire source code***
(3) Scrum Meeting Reports
(4) Presentation Slides

Organize your deliverables into folders, name them so that it is easy for me to locate, like phase 1, 2, 3, 4, 5, code, scrum report, etc.
Zip all of them before submission and upload on Canvas.

Format for the manual is as follows.

User Manual

1. Introduction

1.1***Purpose:***

Our group consists of Elliot Fayman, Christina Mourad, and Ryan Kao. We are writing this document to provide information on what hardware is required to run our software, hardware/software parameters needed to set in order to run the software, procedure on how to get the software running, and showing a demo of our application.

Elliot: Worked on configuring Twilio account to be able to send SMS messages, configuring Twilio to post to a specific Flask endpoint, and Adding a Flask endpoint that processes Twilio Post. Elliot also contributed equally in working on this report.

Ryan: Worked on adding a method to Flask api that sends sms messages to a specific phone number, and using send SMS method to send an SMS message upon receiving a user sent SMS. Ryan also contributed equally in working on this report.
Christina: Worked on improving home screen styling, creating a new view for messaging, and creating a new Flutter UI that allows users to type and send text. Christina also contributed equally in working on this report.

1.2 Definitions, Acronyms or Abbreviations – Any definition, acronyms or abbreviations used in this document.

Python: A high level, general purpose programming language known for its simplicity and readability.

Flask: A lightweight and flexible web framework for Python, designed to make it easy to build web applications quickly and with minimal boilerplate code.

Flutter: An open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop platforms from a single codebase.

Dart: A programming language developed by Google that is optimized for building user interfaces, particularly for creating fast and efficient applications using the Flutter framework.

LocalHost: Default loopback network address (127.0.0.1) on a computer, which allows programs and services to communicate with each other on the same machine.

SMS: A text messaging service that allows the exchange of short text messages between mobile devices.

Terminal: A text-based interface on a computer that allows users to interact with the operating system and execute commands.

Ngrok: A tool that creates secure tunnels to expose local servers to the internet, allowing easy access to local web services for testing and development purposes.

Twilio: A cloud communications platform that provides APIs and tools for developers to integrate messaging, voice, and video capabilities into their applications.

Toll-Free number: A telephone number that allows callers to reach a business or individual without incurring any charges, with the recipient bearing the cost of the calls instead.

1.3 References – Any references used in this document

Jira:
https://webcraft.atlassian.net/jira/software/projects/TEX/boards/1/backlog

GitHub:
https://github.com/elliotfayman/TextPy

2. Hardware Configuration

The hardware that is required to operate the application is a smart device with a sim card with a registered phone number and a windows computer that can be the host for the server. In addition, it is required that your phone has access to SMS messaging and the computer be connected to the internet. It is also recommended that you use a mouse, keyboard and monitor for the computer that you will use.

3. System Parameters

In order to run the application you must first ensure that your computer has Flask 2.2.2 and python 3.10.11. You must also create an ngrok account and install the ngrok executable. Additionally, you must also have two terminal windows ready to be used on the host computer and an additional ide like android studio or xcode downloaded. To run the Flutter application on the phone, you need to download Flutter 3.7.3 and Dart 2.19.2 and install the mobile application onto your phone using your ide of choice. Lastly, you must have the WebCraft Twilio account on hand with the webhook for one of the Twilio toll free numbers open and ready to be updated.

4. Operation Procedure

In order to use the application, you must first launch the Flask application in the local host from the computer terminal. From there, you must execute ngrok.exe in another terminal window and point the generated link to the open port connected to the Flask application. From there, you must go to the Twilio webhook and update the url to be of the ngrok link generated in the console after running the executable. Lastly, the user can then launch the mobile app and text the toll free number provided by twilio in order to send and receive interpreted python code via SMS messaging.

5. Demonstration

Show your working system.

Select any one requirement (feature) and execute it from start to the end.

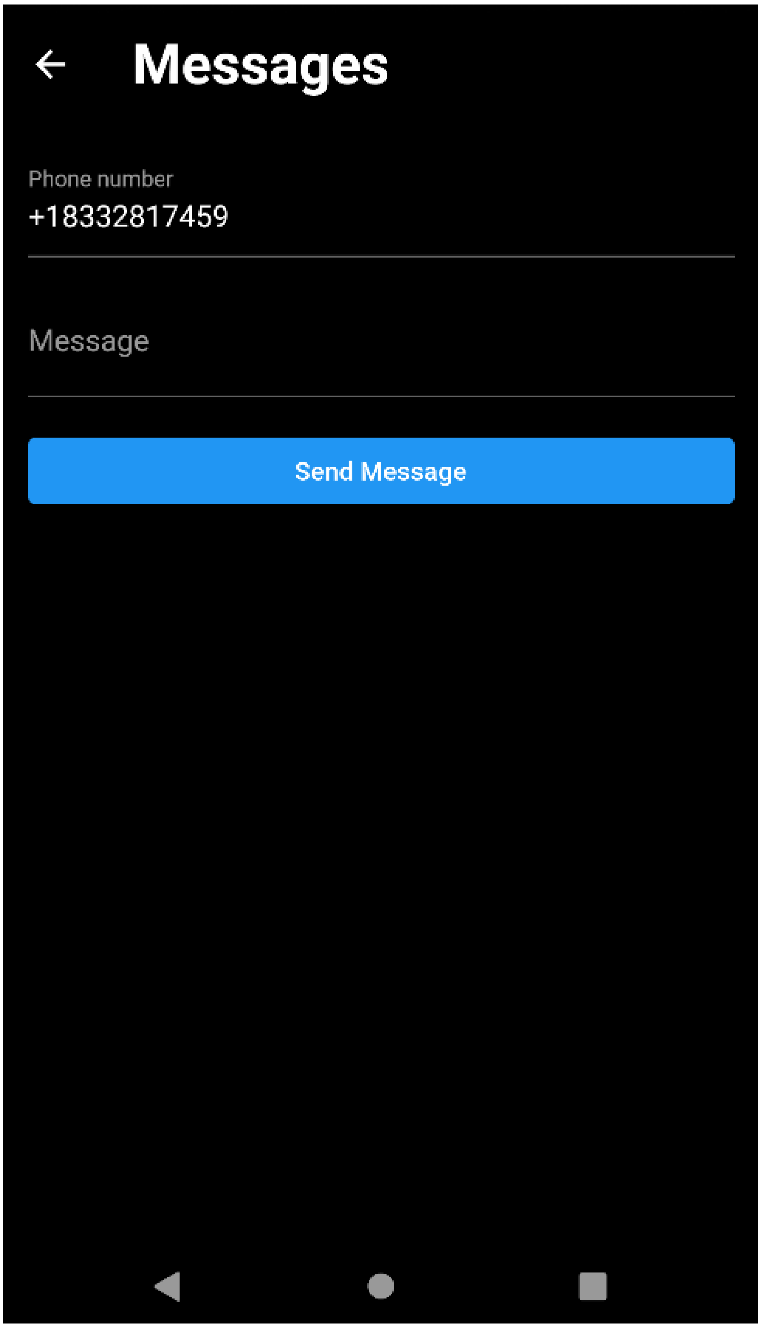
Attach screen dumps for that requirement.

5.1 **ChatPage**

a) Description.

In our app, the user is able to view their sent and received SMS messages after inputting their phone number and message.

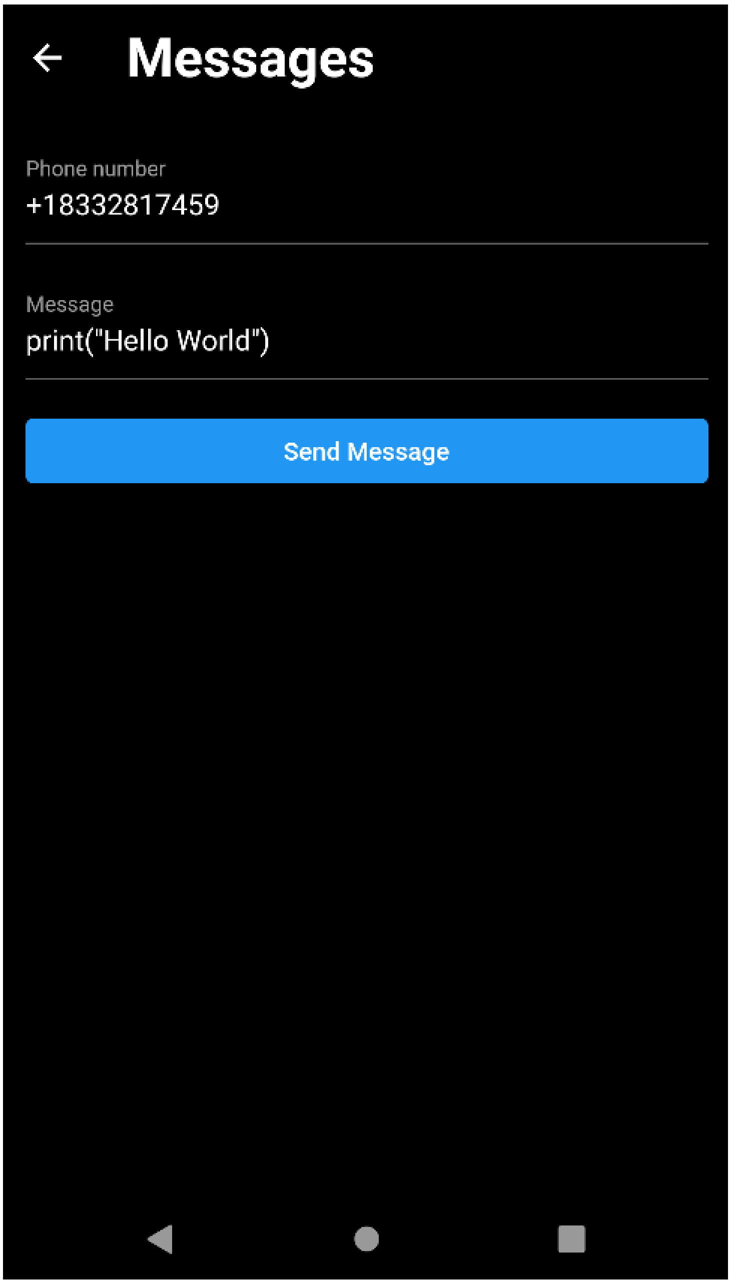
b) Screen Dumps.



Messages

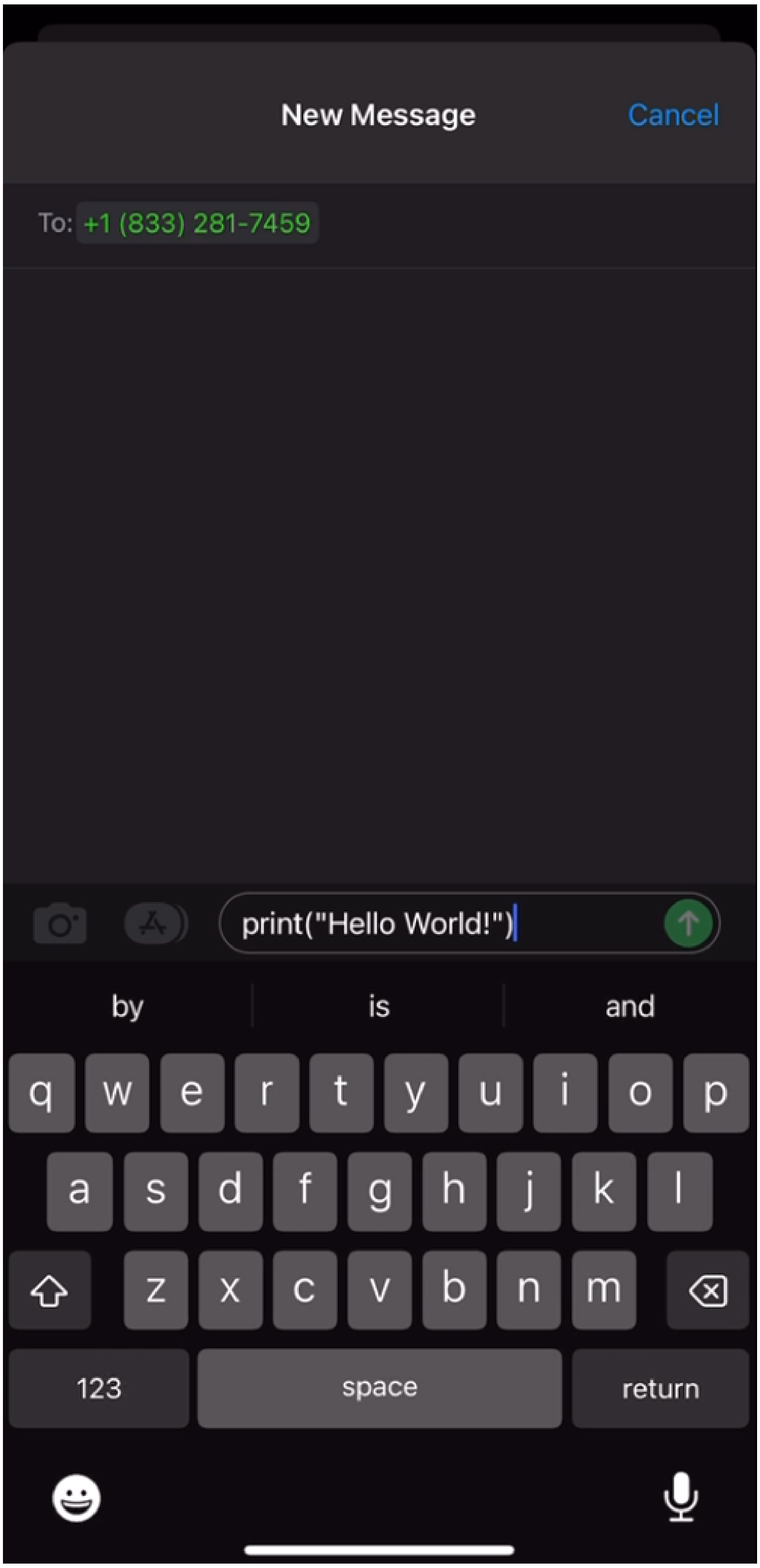
Phone number
+18332817459

Message



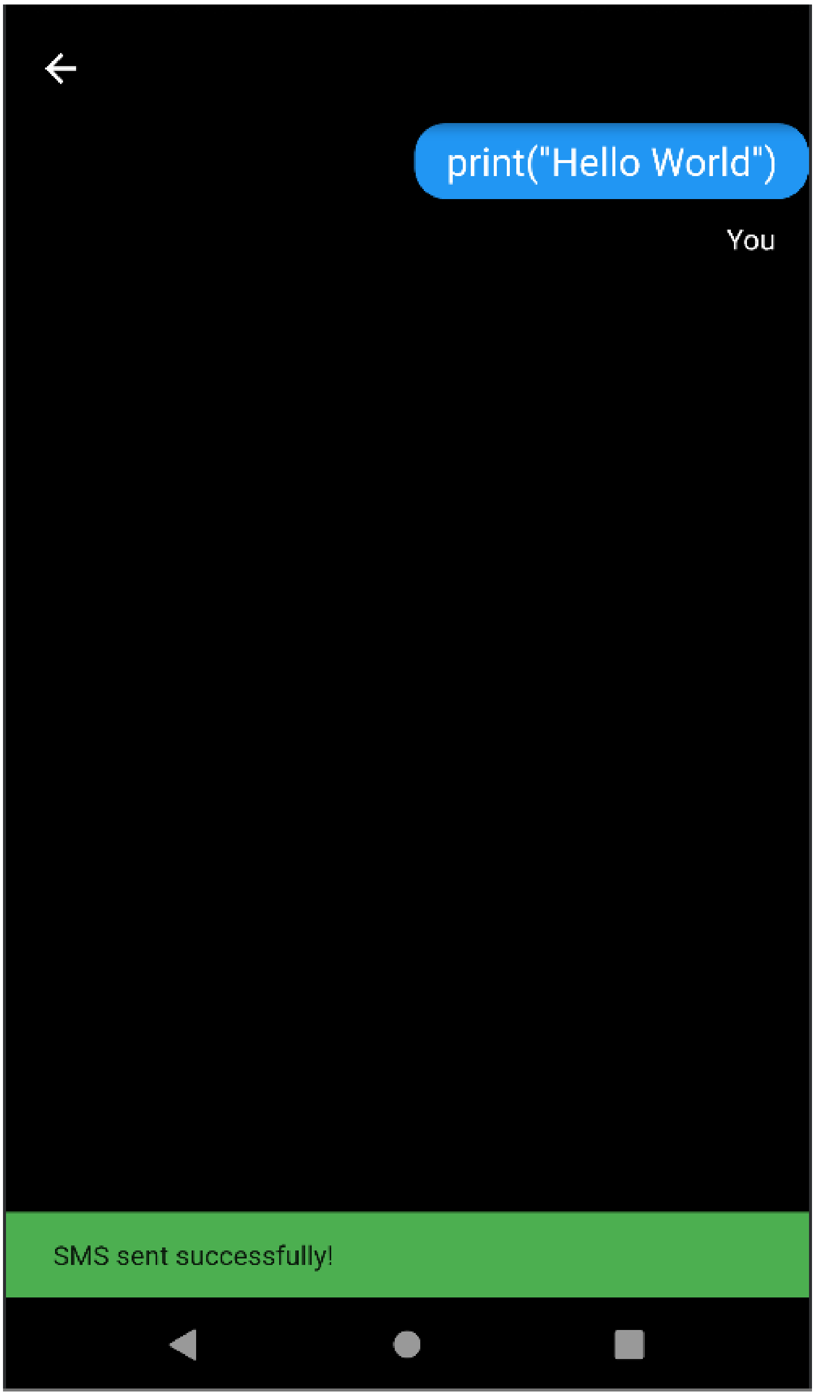
Phone number
+18332817459

Message
print("Hello World")



To: +1 (833) 281-7459

print("Hello World!")



print("Hello World")

You

SMS sent successfully!