

CSE 490C, Programming Assignment 1, Due Wednesday, October 17, 2018, at 11:59 pm

*For this assignment, you will be building a two-way SMS application using [Twilio](#)<sup>1</sup>, where a server sends SMS messages to clients and the server processes SMS messages sent from clients. This assignment has a written portion and a programming portion, worth 50% each. This assignment has to be done individually. You will need to submit a .zip file of your code through Canvas and demo your assignment to the TAs. There are no requirements as to which programming languages you use in this assignment.*

**Write-up specifications:** In about 500 words, describe the objective of your SMS system. Examples of these types of SMS applications could be providing maternal health information, immunization tracking, or ensuring medication adherence. What does it do and who is it supposed to serve and how? Whom do you consider your end users/clients to be and what service or information is relayed through the SMS messages? Situate this description in the context of the location of deployment (e.g., the country's telecommunications infrastructure, the user needs the system supports). Describe a specific use case where the client both receives SMS from a server and sends an SMS for the server to process. Justify your design and user experience decisions.

**Programming specifications:** Develop a use case for a two-way SMS application which successfully fulfills the following requirements:

- An SMS server that sends SMS messages to clients and also processes the SMS messages sent from the clients. Implement this use case using Twilio. You should have a total of four examples. Two of those messages (160 characters each) should be messages sent by the server to the clients, and the other two messages should be ones that the client might send to the server.
- The server must meaningfully process these messages—the SMS might meaningfully respond to the client's message (this does not count towards the two messages from the server that we require), update future responses based on the information provided by clients, collect data submitted by the client, etc. Any data collected can be stored in the backend in simple data structures, and does not need to be persistent. However, there must be some evidence of this data being processed and stored, such as a confirmation message to the client or a test SMS showing updated data.

---

<sup>1</sup>Twilio is an SMS Gateway service that for commercial, scalable SMS applications. To complete this assignment a free account is probably sufficient as it supports building applications that communicate with registered phones.

Please note that you do not have to create a completely functional system:

- This assignment can be a proof of concept where someone using the system can get an idea of how it would ideally work.
- You do not have to worry about administrative logic, such as creating user IDs, user onboarding, etc. You can focus on the business logic, such as what the purpose of the system is, the content of the SMS message, and how users can and should reply to messages.
- In real deployments, these systems are often timed. For example, the server might send a reminder to take medication every 24 hours. However, since you'll be demoing these projects (as we describe below), you can build a button into the system that will send an SMS from the server at will and simply simulate the time it would be sent (e.g., via the text of the message).

**Submission:** You should submit the code through Canvas in a .zip file (firstname\_lastname.zip) by the due date. Make sure to add your name, student ID, and comments within the code to explain your implementation. If you include external APIs, header files or packages, make sure to briefly explain their need and use in the implementation.

**Demo:** Each student will then be required to give a short demo of their code to the TAs (sign-ups for demos will be announced soon). We will inspect the code to ensure that it is original and reasonably matches up to the functionality of the system as shown in the demo. Demos will be graded based on the following criteria:

- Clear and comprehensive presentation of design and implementation details during the demo.
- No serious complications or obviously missing components of the system.

*Note: We have zero tolerance for plagiarism. Plagiarism will result in either a zero or disciplinary action. Ask the TAs or the instructor if you have any questions about collaboration. In general, asking other students how they went about the assignment is appropriate, but copying code or writing code for someone is not.*