# Software Limitations

### Twilio

- A limitation that I encountered with twilio's sms service that was mentioned by my peers in their limitation write up was that my software is only capable of sending messages to phone numbers that I have verified myself personally in my twilio account. The only way to allow the survey system to send messages to any who is signing up is for me to upgrade my twilio account to a paid version

### Servlet

- My servlet is not able to connect using HTTPS, it can only connect with HTTP.
- If you are looking to deploy this yourself on your own server, you must change the url redirects in all the servlet classes because I "hardcoded" them to use my server's ip address.
- If you are looking to use your own twilio account, you must change the acctSID and authToken in the smsVerify java class (located in the smsVerification java project). It is also very bad practice to hard code the values like I did. It is encouraged to set them up as environment variables.

### Mysql

- I believe that the Questions table, QuestionChoices table, and.QResponce table constraints need to be adjusted, the way their relationships are defined prevents the survey system from storing multiple surveys and from storing multiple answer questions.

### Survey System

- The survey system is unable to store more than one survey in the database. So if you are looking to add a new survey, the Survey table, Question table, QuestionChoices table and QResponce table must be cleared out first
- If the user enters a phone number that is registered but not verified, then they will be told that their phone number
  is not verified. (Unfortunately I forgot to allow a unverified user to comeback and verify their number, so if this case happens the only way to be able to log in is to delete the user from the table "Usr_Info" and make sure there
  is no one time passcode associated with the user in "Usr_verify". Then the user must re-register.)
- If the user attempts to add a question with a question description but no question type, then a flaw in my javascript code occurs. The question will be added

perfectly fine into the list but the question will have a type of 0 which will cause an error in storing and retrieving the survey.

- If the question is a Multiple Choice or Multiple Answer question, then there will be an additional textbox and button appear under the question which will allow the user to add choices to the question.(A flaw with this is that my code allows a user to submit a survey that contains a question(s) that have no choices. For every choice added, the user is able to adjust the choice description if need be or if they would like to delete it, there is a "delete" button
associated with each choice added.

- A major flaw with creating multiple surveys is that the database only allows for one survey to be stored. When I attempt to create another survey it only stores the survey title, description, survey id, and user id; but it does not store the questions. The reason this occurs is either because of how my code generates a question id (My approach was for each question created the question Id is the question number) Meaning that since a previous survey already used question id= 1, 2, ..., n. The question id's in the next survey that also has those question id's that were used by another survey will fail to be stored.

- When the survey page is being displayed, it is only capable of properly displaying the multiple choice and multiple answer questions properly. When the survey retrieval code encounters a reponse type question it does not provide a textbox for user input, it only prints out the question. However this flaw is not because of the survey retrival code, the reason this flaw happens is because of the Json object created by the "SurveyExtIns.jar" file. The Json object created by this file adds question-type key values to both Multiple Choice and Multiple Answer but does not add a question-type key to a question that is a response type question. Therefore causing Meri's survey retrival code to skip the step of adding a textbox area to take in a response by the user.

- The survey system in my case is only capable of storing responses from multiple choice questions only (or multiple answers, so long as only 1 choice is selected for each multiple answer question). A repsonse from a text response question cannot be stored because the user isnt able to fill in any textbox due to the problem with the Json object created in "SurveyExtIns.jar". Then the problem with selecting multiple choices I'm not really sure why that is happening but I suspect that either I made a mistake when I created the "QResponces" table or there is a flaw with the database schema particularly the relationship between the "QResponce" table and the "QuestionChoices" table.

- Another flaw is that the only responses that can be stored is by the survey creator themselves, the survey system does not store responses from anyone else.