CitiVan SMS App Review

Michelle Sit and Nahom Marie

Overview

CitiVan SMS App is designed to gather information about riders' experiences on public buses through a short and simple survey completed over SMS messages. The information is gathered in a database on Iris Couch, an online cloud database. When the CitiVan phone number receives a SMS message from the rider, it automatically sends a welcome message and sends a five question survey. The questions are:

1. "What is your bus number?"
2. "Pick a number from 1 to 5 to rate the quality of your ride. 1) Very poor. 2) Poor. 3) Average. 4) Good. 5) Excellent."
3. "Was your driver speeding? Pick 1 or 2. 1) Yes 2) No."
4. "Was your driver courteous? Pick 1 or 2. 1) Yes 2) No."
5. "Was your minibus clean? Pick 1 or 2. 1) Yes 2) No."

We were provided with the following to conduct our test: the SMS number to message to access the survey, the survey results on Iris Couch, backend code for the SMS survey on Tropo, and CitiVan Mobile App in an Android Application Package. We used Andy, an Android emulator, to run the mobile application. We were not provided with the templates for the final product of the application. The following results pertain to the SMS messaging app.

Our testing yielded mixed results. The app is able to send the survey over SMS messages, store answers in real-time, and achieve its goal of a lightweight and barebones system. However, key functions in the app do not work appropriately and there are areas of user experience that could be improved. Some work and reliability testing will need to be done before the app can be deployed to a small sample group but further work will be necessary before the app can be released on a wide scale.

Hidden and Broken Commands

"Back" functionality

We discovered by reading through the smsdatabase.rb file that there is an option for the user to return to the previous question by writing "back" to the text number. However, this was never mentioned to the user at any point which would likely mean users would not be aware of the function and less likely to use it. To fix this, we recommend that a short statement should be included in the welcome message such as "To return to the previous question, write 'back'".

Furthermore, we discovered that this function has a few errors. As seen in Figure 1, we discovered that the function is case-sensitive and yielded different results depending on whether the command was sent in lowercase or uppercase. If the user sends "Back", the app will return an

error message and resend the same question. If the user sends “back”, the app goes back two questions. This can easily be fixed if the app is written to ignore case sensitivity for this word.

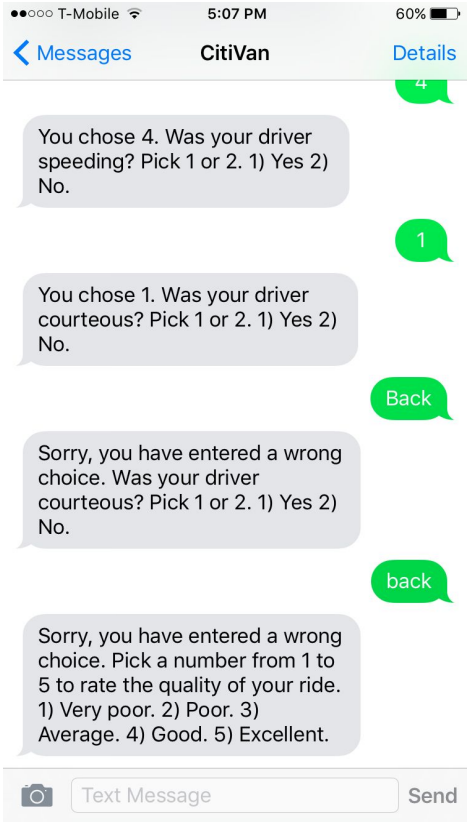

Figure 1: Survey is unable to handle case sensitivity for the “back” command.

Specifically for Question 2, when “back” is submitted, the system erases the answer provided by Question 1. In other words, the answer to the question “What is your bus number?” becomes “back” as seen in Figure 2. This error did not seem to carry into the log files. That is, even though the user is told the bus number is now “back” the previous bus number remains logged into the log files. The user can easily fix this by sending “back” again to restart the survey; however, this inconvenience should be fixed to resolve this error and improve user experience.

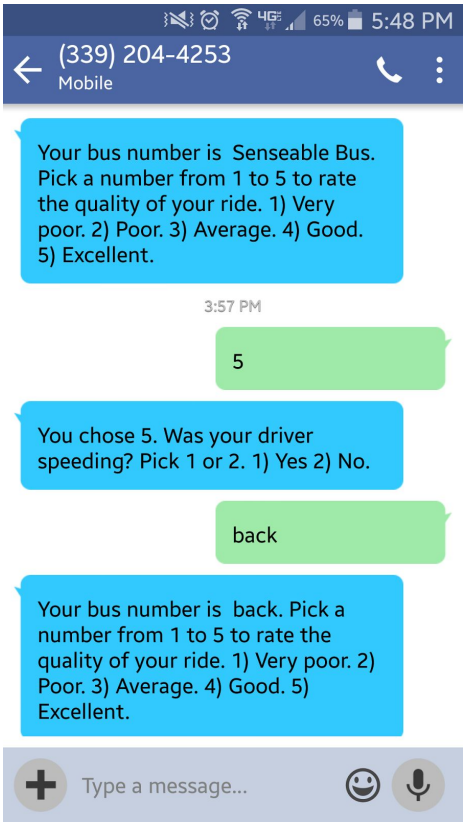

Figure 2: Survey incorrectly replaces the bus name with "back" if the user tries to return to Question 1.

"Rate" functionality

We also discovered by reading through the code that users can send the command "Rate" to view the average ratings for a particular bus. However, this option was never mentioned to users nor does it work. The code attempts to call the information from the Iris Couch database, but it fails to handle the data and calculate the averages resulting in error messages. Even bus numbers and information we submitted or from the developers' initial sample set of data produced error messages as seen in Figures 3 and 4. More work will be needed to debug this functionality.

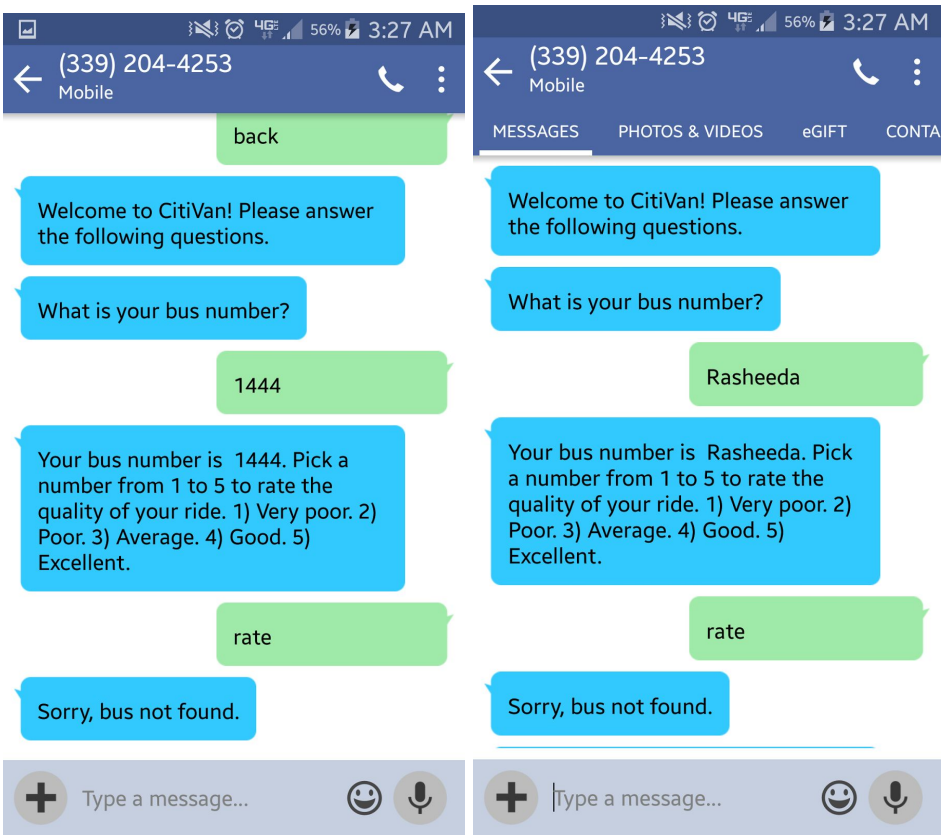

Figure 3 and 4: Both 1444 and Rasheeda are valid bus names stored in the Iris Couch database. However, sending "rate" to the app resulted in an error message.

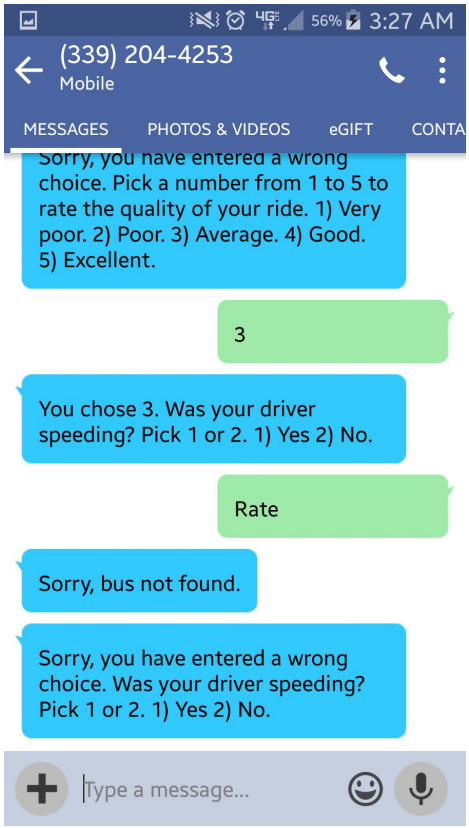

Figure 5: Changing the case of "rate" also resulted in an error message.

Missing an "Exit/Cancel" function

The option to exit or cancel the survey does not exist. The inability to exit from an incomplete or unresponsive survey because it has thrown an error, unbeknownst to the user, can also cause a number of serious problems. Users will be forced to complete surveys that will skew the received data and provide inaccurate feedback. Furthermore, if an inappropriate answer is recorded in the database and an error is thrown, the user cannot exit that survey or request for a new one (see Figure 6). Our iPhone encountered this error, but there was no user feedback to indicate that an error had been encountered in the database or instructions sent to solve the situation. Instead, our requests for the survey to send us a response were met with silence and made the system seem unresponsive. We could most likely use the "back" functionality to return to the beginning of the survey. However, without knowing that the database had encountered an error or that the "back" functionality exists, we concluded at the time that the app had failed for iPhones since our Android phone was still able to access the system. Providing better user feedback and an exit function is necessary for basic functionality.

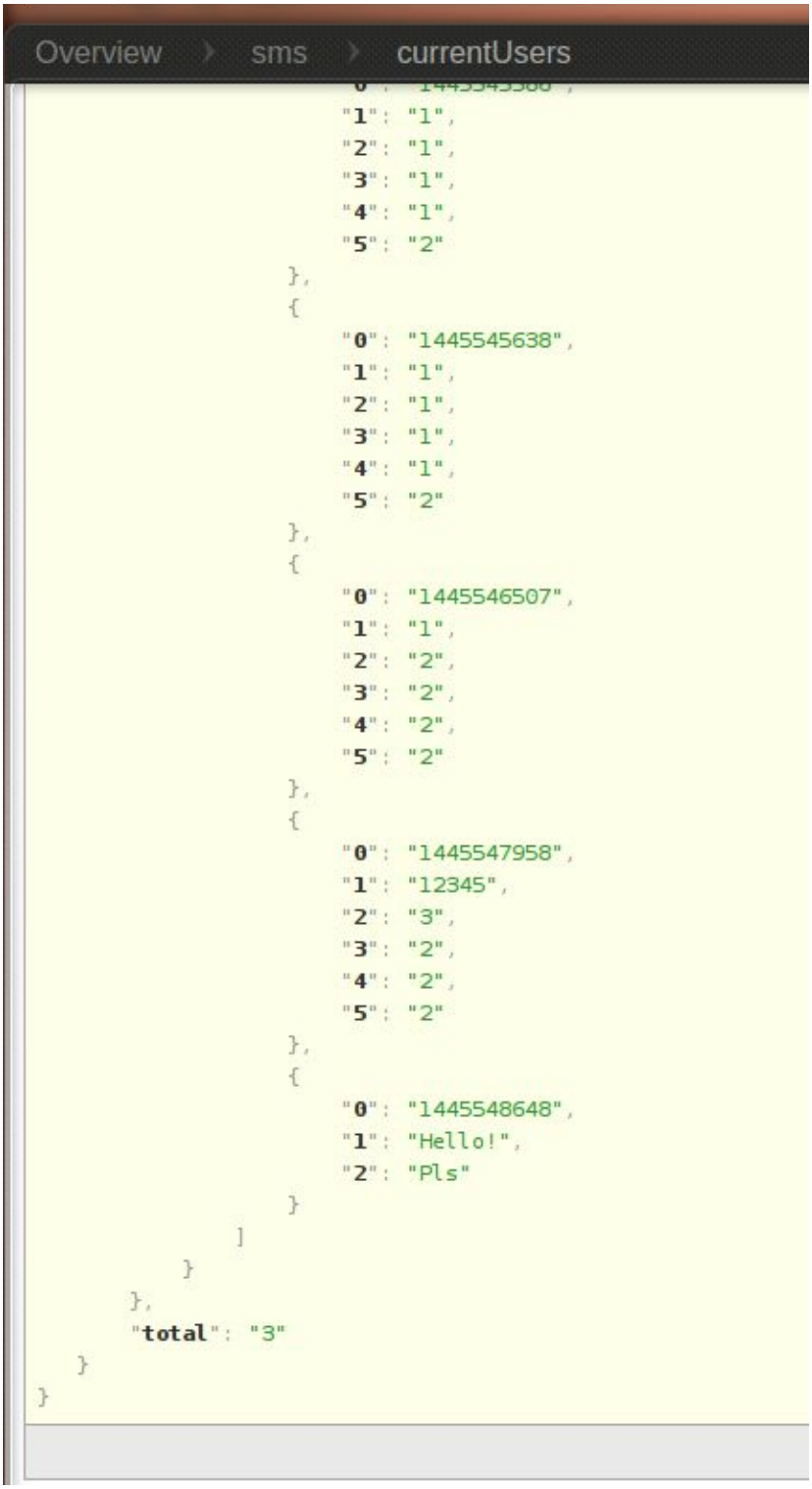

Figure 6: The last entry for this user in the Iris Conch Database shows the incorrectly recorded answer for Question 2. This unhandled error prevented our iPhone from accessing the survey or eliciting a response from the system.

Backend Review

Current state of backend

In the Tropo www folder, there are two files: citivan.php and smsdatabase.rb. smsdatabase.rb is the most up to date version of the system which interacts with users when they send SMS messages to the CitiVan number. citivan.php is an old iteration and is not used. The Tropo logs folder contains log files created when smsdatabase.rb is accessed. The log files are segmented by hour and are condensed to save space. However the information in these files is largely useless.

The Iris Couch database has a file that aggregates survey responses from users in real time and has a small sample of example submissions. It appears that the data is being stored reliably. Users accessing the system at the same time will not have their results mixed or misplaced. However, as stated above, accessing the information for analysis will need to be fixed and especially if the "Rate" function is to be useable.

Further recommendation for improving backend

One suggestion is to handle user submissions for Question 1 in a more robust manner for information analysis. Currently Question 1 accepts any combination of characters, numbers, and symbols except for the word "back." The exception to this rule is if there is any variation of the word "back" such as "Back" or "back123", in which case the app accepts it as a valid bus name. However, if "Rate", "rate", or "rat" are submitted or are at the start of the bus name, the system returns "Sorry, bus not found." For user interaction purposes, allowing users to submit any variation of bus names without prompting the user to correct their answers to adhere to a specific format improves their experience. However, the Ruby code does not appear to standardize cases of these answers when storing it in the online cloud database. Similarly, there is no indication that the information is standardized when it is pulled from the database for analysis. This is problematic since bus names containing characters or special characters such as a hyphen or parentheses may be recognized as different buses and result in inaccurate results. Adding this functionality is a relatively simple task but is important for accurate analysis.

User Experience Review

The overall user experience of the app is both positive and negative. Sending single digit answers is simple and easy. It works for small scale testing. There are a number of additional features that could be made to improve user experience but are not necessary for the baseline functionality of the app. However, these improvements to the system would vastly improve user experience especially in crucial cases where users encounter errors or problems. The lack of an error notification coupled with the app's silence when it encounters an error makes the system appear to be unreliable and broken. This issue will need to be fixed before the app is deployed for any type of testing. Similarly, notifying users at the beginning of the survey about key functions such as "back", "rate", and "exit" would allow them to take advantage of these resources.

Further recommendations

Having the option for the user to select “I’m not sure” as a response to Question 3, which asks users to identify if the bus driver is speeding, would be useful. Without seeing the speedometer and knowing the speed limit of the area, determining whether the bus driver is speeding becomes a relative assessment. Some users may judge that the bus is speeding when the bus is actually traveling under the speed limit. Without adding a third option, forcing user feedback for this question may not accurately represent the speed of drivers.

User submissions should not be case-sensitive. As stated above, the program only runs certain functions if the user submits the command in a specific format which results in error messages or frustration on the user’s end when the app does not perform as expected. Handling this information appropriately in the backend will greatly improve user experience.

The feedback at the beginning of each message currently is not very useful. As of now, users are simply told the number they selected for the previous question. A more substantive feedback response would allow the user to be more aware of the answers they just submitted and subsequently be better able to decide if they are satisfied with their answers as accurate reflections of their experience. If the user is not satisfied, the user then can go back and edit their response. Similarly, providing more specific feedback on why the app has encountered an error and providing an easy solution would help users use the app more intuitively.

We propose that users be prompted for additional feedback at the end of the survey. Although this information could be difficult to process given the magnitude of feedback received daily and the additional work of creating a system to accurately interpret user experience, it could ultimately provide a better lens for interpreting the true experience of users. For example, in cases where users’ answers appear to be contradictory, a feedback section would allow them to explain why they might rate their experience as excellent but state that the bus driver was courteous and drove over the speed limit.

Mobile App Review

Unfortunately we were unable to test the mobile interface. As stated above, we did not have the template for the final product of the application so we could not make any conclusions about design differences between the current product versus the end goal. We were also unable to log into the mobile application because we did not have the appropriate login credentials.

Conclusion

Overall, we have concluded that the basic functionality of the app could allow for a small sample of drivers to test the app if some work is completed beforehand. The SMS messaging app reliably captures submissions. However, the “back” and “Rate” functions must be fixed and an exit function should be implemented before the app can be released to this sample. More work on implementing better error handling and feedback sent to the user would greatly improve this small sample’s experience and is necessary to prevent fatal errors, such as the one we encountered on our iPhone, from occurring. For large scale deployment, improving user

experience will be mandatory for ensuring this system works and creating working methods to pull and handle the database data for analysis is necessary.