

---

# USABILITY HEURISTICS

Kushal Singh

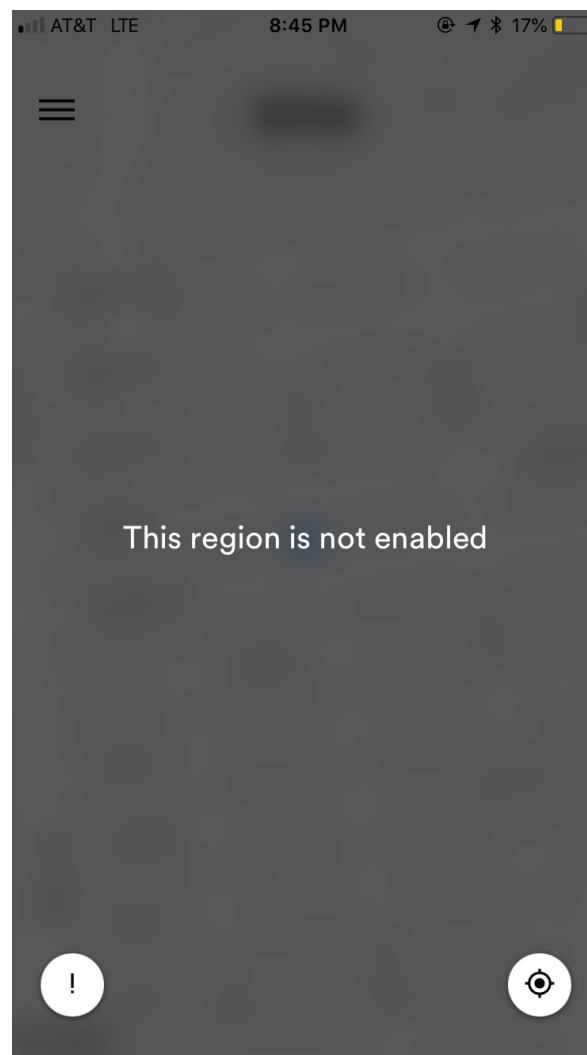
Design 02

CS160 - User Interface and Design Development

Professor Paulos

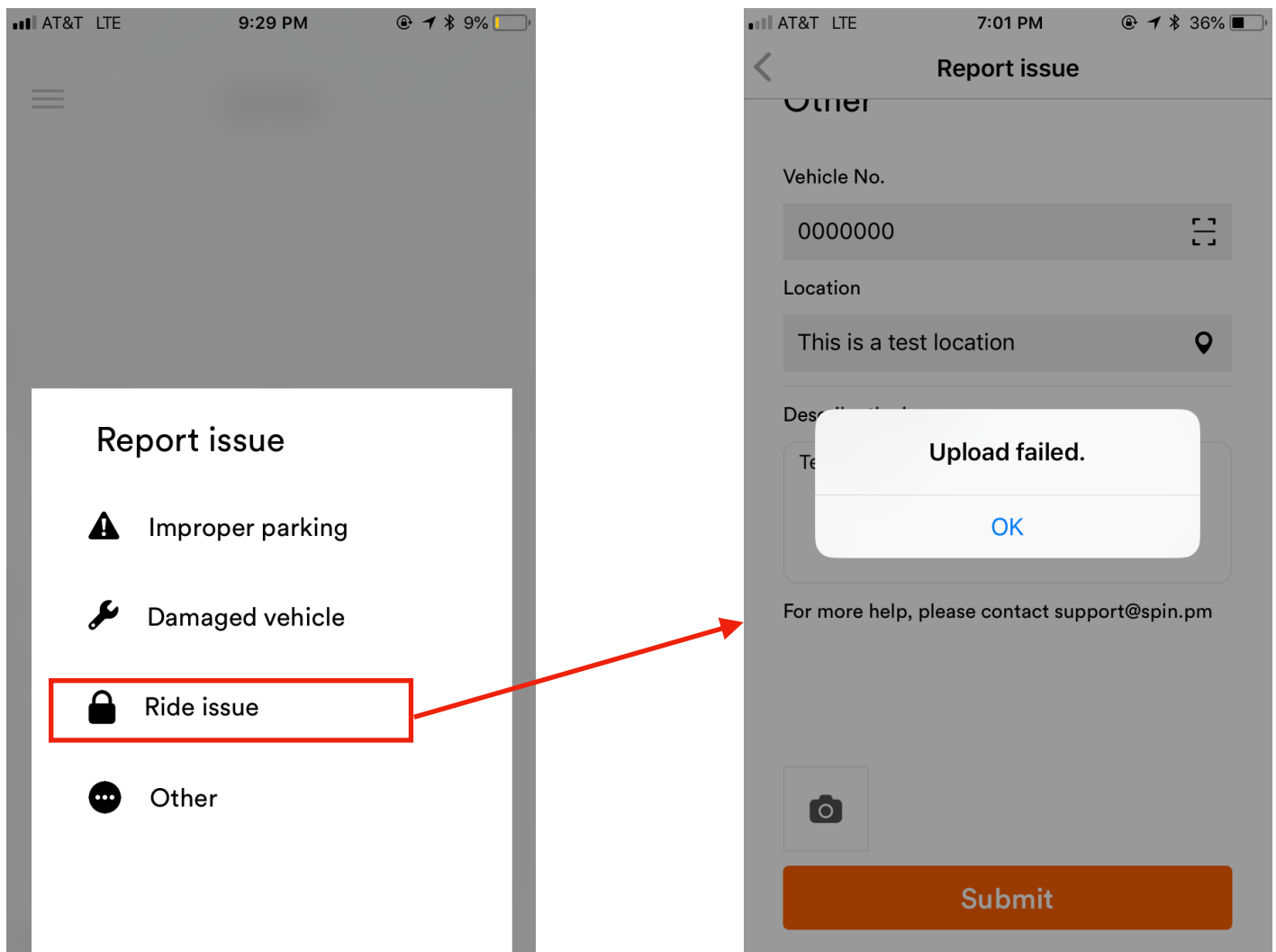
---

**HEURISTIC VIOLATION: Help users recognize, diagnose, and recover from errors**



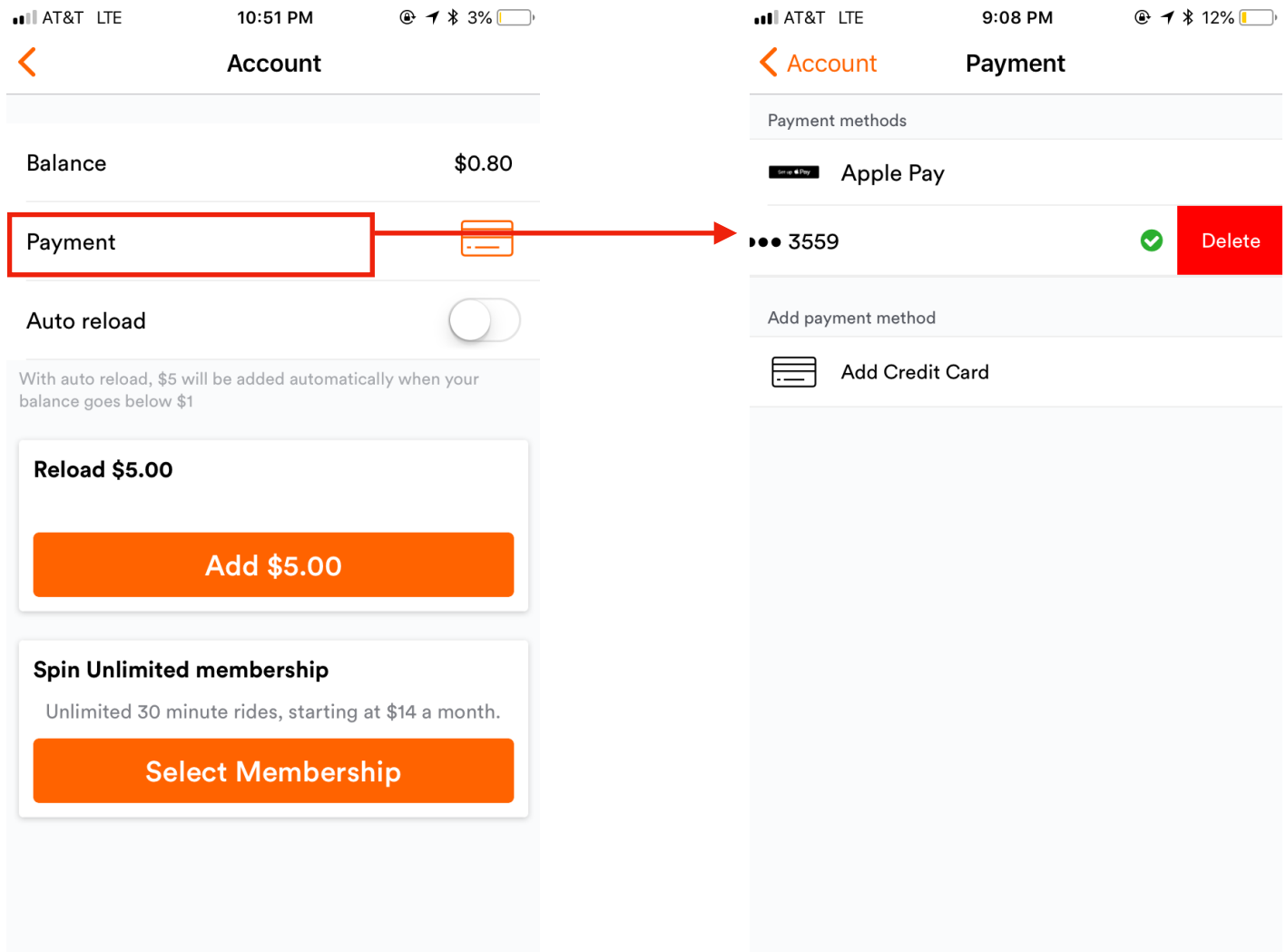
When I used this app in my current location (Berkeley, CA), the first screen that popped up was this. Firstly, this error message ([This region is not enabled](#)) is not specific and helpful, as the user is left guessing what the application means by [This region](#). Furthermore, the app fails to offer how a user can go about fixing this problem. Thus, this design violates Nielsen's [Help users recognize, diagnose, and recover from errors](#) heuristic which states that error messages should precisely indicate the problem, and constructively suggest a solution. I would rate this as a [Severity 4 \(i.e. major usability problem\)](#) issue, since although this problem only occurs on this screen, it's fairly difficult for the user to know how to overcome this issue. Also, it's the first screen that the user sees once he or she logs in, which means the user could just quit the app out of frustration. Furthermore, this problem could persist, in that there could be very many regions in which the application [is not enabled](#), and the user is unaware of what these regions are.

## HEURISTIC VIOLATION: Error Prevention



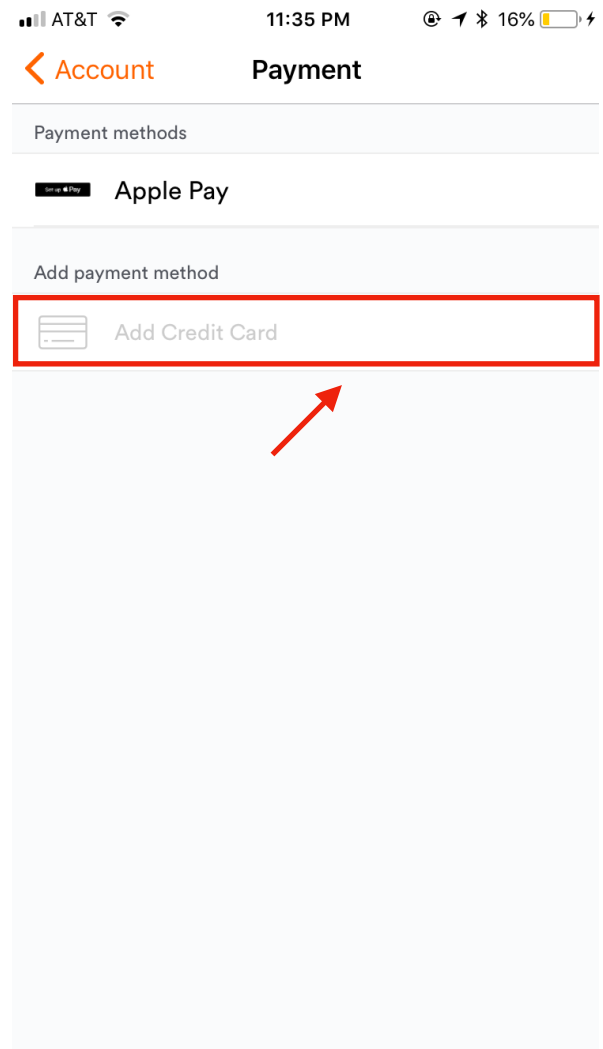
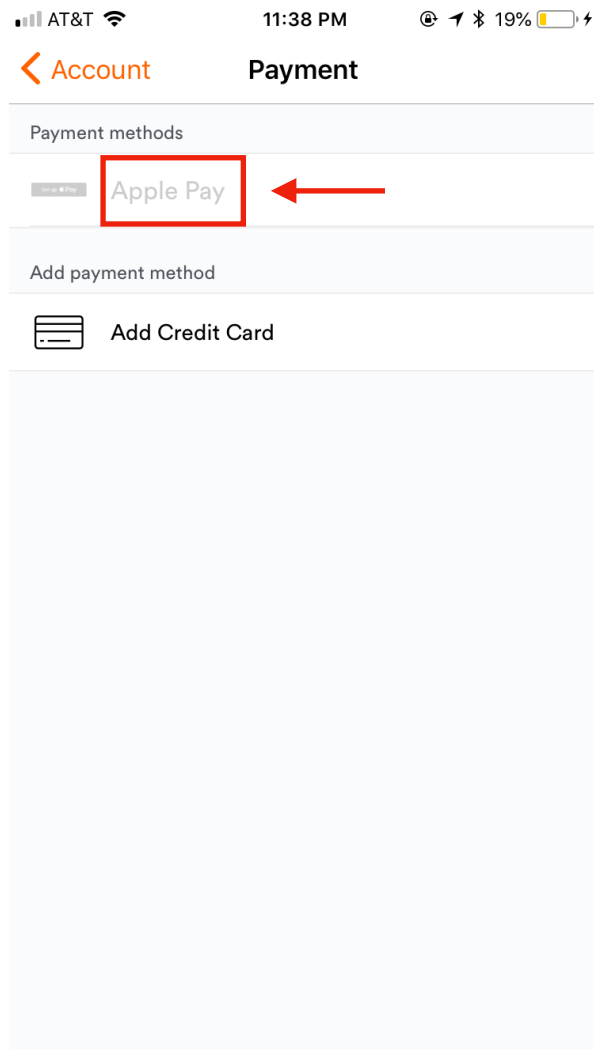
After a user clicks on [Ride Issue](#) and inputs some invalid information in the [Vehicle No.](#) text field and [Location](#) text field and hits [Submit](#), there is an Error message saying [Upload failed](#). This design violates Nielsen's [Error Prevention](#) heuristic which states that a careful design prevents a problem from occurring in the first place. In this case, the app should have highlighted an erroneous field in real-time, once a user inputs the information for that field. In this case, the user doesn't know whether the upload failed because the [Vehicle No.](#) is incorrect or whether the [Location](#) is incorrect. I would rate this as a [Severity 3 \(i.e. major usability problem\)](#) issue, since this error message is fairly frequent (it occurs on the [Improper parking](#), [Damaged vehicle](#), and [Other](#) screens as well), it is pretty cumbersome for users to identify which field is invalid (since one of the other screens has 5 input texts), and the user will repeatedly be bothered by this issue if it isn't addressed.

## HEURISTIC VIOLATION: User Control and Freedom



After a user clicks on [Payment](#) and enters his or her credit card information, it gets added to the Payment system. However, if the user swipes left on the credit card information, the [Delete](#) button pops up. If the user clicks [Delete](#), the credit card information is immediately erased, without any form of confirmation (*Are you sure you want to delete?*), in the case that the user clicked this button by accident. This violates Nielsen's **User control and freedom** heuristic which states that users often choose system functions by mistake and, so, your design should support [Undo](#) and [Redo](#) operations. I would rate this as a **Severity 3 (i.e. major usability problem)** issue, since although this functionality only occurs on this screen, it has the potential to be quite difficult for the user to overcome if he or she accidentally hits delete and doesn't have access to his or her credit card. Furthermore, this problem could persist, if the user hits [Delete](#) by accident again in the future, even though the user will learn from the first time that the application will not prompt the user to verify whether he or she wants to delete.

## HEURISTIC VIOLATION: Consistency and Standards



The user has two options to add a [Payment](#) method: [Apple Pay](#) or [Add Credit Card](#). However, in order to choose [Apple Pay](#), the user must explicitly click on the text, whereas if the user wants to choose [Add Credit Card](#), he or she can click anywhere on the row. This violates Nielsen's [Consistency and Standards](#) heuristic which states that users should not have to wonder whether different actions yield the same result. I would rate this as a [Severity 2 \(i.e. minor usability problem\)](#) issue, since this problem only occurs on this screen, and it's fairly straightforward for the user to overcome—he or she can just click on the text to be safe. Furthermore, this problem would most likely not persist, in that the user will probably learn to just click on the text to avoid having to remember which payment type can be pressed where.

## HEURISTIC VIOLATION: Match Between the System and the Real World

Two side-by-side mobile app screens illustrating a heuristic violation.

**Left Screen: What's your email?**

- Status bar: AT&T, 8:32 PM, 26% battery.
- Back arrow icon.
- Title: What's your email?
- Email input field: kushal.singh@berkeley.edu
- Text: At a school or a company? Use your school or work e-mail to unlock offers.
- Next button (orange).

**Right Screen: Choose a password.**

- Status bar: AT&T, 8:33 PM, 27% battery.
- Back arrow icon.
- Title: Choose a password.
- Password input field.
- Text: At least 6 characters.
- Forgot Password? link.
- Next button (gray).

A red arrow points from the 'Next' button on the left screen to the 'Choose a password.' screen, indicating that the user is prompted to choose a password even though they are an existing user.

The application doesn't distinguish between a new user vs. an existing user. For instance, I already have a Spin account, so after I type in my email it takes me to a screen that says [Choose a password](#). Generally, this prompt applies to new users, which is why I was confused when I saw it, since I am an existing user. I was expecting a prompt like [Enter your password](#). This violates Nielsen's [Match between the System and the Real World](#) heuristic which states that designs should follow real-world conventions and make information appear in a natural and logical order. I would rate this as a [Severity 1 \(i.e. cosmetic problem only\)](#) issue, since this problem only occurs on this screen, and it's fairly straightforward for the user to overcome—he or she can just enter the password. Furthermore, this problem would most likely not persist, in that the user will probably learn to just type in the password even if he or she is an existing user.