1. Introduction

According to Johns Hopkins School of Medicine, medical errors are the third leading cause of death in the United States. Their research asserts that a large percentage of medical errors aren’t due to inherently bad doctors. Rather, they’re the result of poorly coordinated patient care and, as a result, unwarranted variations in medical care.\(^1\) A caregiver “no-show” is an example of an unwarranted variation in medical care and, unfortunately, the cause of many preventable deaths. A caregiver is defined as someone who cares for a person who cannot care for themselves.\(^2\) One of the most common complaints among home care clients is that their designated caregivers fail to show up for their shifts.\(^3\) Caregivers’ absence has devastating effects on the health of the care client, including the introduction of new physical injuries. In addition, caregivers often take advantage of their clients. Caregiver fraud is a common phenomenon that can take the form of forgery, embezzlement, and identity theft, among other misconducts. Recent instances of caregiver fraud in the United States have ranged from caregivers stealing their client’s credit cards to posing as medical practitioners and charging for never-provided services at an exorbitant fee.\(^4\) Considering this finding, I’d like to create a React Native application that detects caregiver “no-shows”. The motivation of application is to prevent frequent caregiver no-shows and prevent caregivers from claiming hours that they did not work.

---

1. *Johns Hopkins Medicine, Based in Baltimore, Maryland*, www.hopkinsmedicine.org/news/media/releases/study_suggests_medical_errors_now_third_leadingCause_of_death_in_the_US.
2. *Johns Hopkins Medicine, Based in Baltimore, Maryland*
1.1. Project Description

A React Native application that detects caregiver “no-shows”. The application will have four user groups: caregivers, caregiver agencies, care clients and care client family members. To detect “no-shows”, once at a care client’s home, the caregiver will be expected to scan the care client’s unique code--a QR code--from the care client’s phone or input a code (from some other device located in the caregiver’s home) to confirm their arrival. Once the correct QR code or text code is successfully scanned or inputted, all user groups except the caregiver will be notified of the caregiver’s arrival. If the caregiver does not successfully scan the care client’s QR code--presumably because they did not show up--all user groups except the caregiver will be sent a “no-show” notification. The caregiver’s “attendance” will be stored in a distributed database that can be accessed by all user groups.

1.2. Existing Systems

There are a number of existing applications in the application market that support caregivers. For instance, CareZone helps caregivers keep track of their care client’s medical information, including medication, allergies, and insurance information. CaringBridge is another example, enabling caregivers to seamlessly divide caregiving duties among other caregivers. Despite there being a litany of technologies tailored to support caregivers, no application has been created to support the care client. The dearth of such an application can be the result of care clients not having access to a smartphone; the care client having a physical ailment or a visual impairment that prevents them from operating a smartphone, and many other reasons.

2. Proposed Project

2.1. Possible Approaches

Most significantly, this project will attempt to improve caregivers’ quality of caregiving by tracking their “no-shows” and holding them accountable when it becomes an incessant problem. The application will not monitor the caregiver’s physical location--although, this may be a
feature that’ll be implemented in the near future. Rather, the application will determine if the caregiver showed up for their shift by checking if: (1) the user code (QR code) that the caregiver scanned or (2) the text code (displayed on a device other than the care client’s phone) inputted matches that of the care client that they’re expected to visit that day.

2.2. Deliverables

2.2.1. A final project enumerating implementation accomplishments, challenges, and code/test snippets.
2.2.2. A software specification
2.2.3. A system specification
2.2.4. A Github repository for front-end code
2.2.5. A Github repository for back-end code
2.2.6. Screencast of the application in use

2.3. Preliminary Timeline

2.3.1. September 27: System specification
2.3.2. October 2: Software specification
2.3.3. October 10: Create backend
2.3.4. October 24: Create front end mockups
2.3.5. November 21: Create front-end with React Native
2.3.6. December 1: User testing Results
2.3.7. December 10: Finalize fall semester final paper

2.4. Challenges:

2.4.1. Implementing Effective push notifications that remind the caregiver to scan their care client’s code. At what time should the notifications sent to the caregiver? Should they be sent to the caregiver based on their location?
2.4.2. Modelling a distributed database to store each caregiver’s attendance.
3. Learning Goals

3.1. Test-driven development (TDD)
3.2. Dual authentication
3.3. The advantages and features of a distributed database (i.e. scalability, reliability, speed, and vertically fragmented data.)
4. References


“Study Suggests Medical Errors Now Third Leading Cause of Death in the U.S. - 05/03/2016.” *Johns Hopkins Medicine, Based in Baltimore, Maryland*, www.hopkinsmedicine.org/news/media/releases/study_suggests_medical_errors_now_third_leading_cause_of_death_in_the_us.