The effects of disruptive communication on the engagement of users in the collaborative consumption model

Duri Abdurahman Duri

March 17, 2016

Abstract

Collaborative consumption is a new term that has recently gained popularity. It describes the recent trend of renting or trading of products and services through apps and common day electronics. Uber and Airbnb are famous companies that are leading this trend and have created a sense of 'sharing' within the economy. By building apps and creating a connection between people who share common interests, many companies have set out to become the 'Uber of X' with the hopes of immense monetary returns. The popularity of these services has created competition to get more people using one service opposed to another. This research came from an intuition that the way users communicate with these apps affects how engaged users are with the app. Therefore, this project entails the creation of an online app that allows people to offer money in exchange for errands. It monitored the activity of users relative to how they are notified of new errand opportunities, primarily looking into text and email messaging relative to a user not receiving notifications at all. Data was collected for a seven-week period and analyzed for patterns in usage. Findings from this project report that notification type is indeed correlated with user engagement. Finds from this research also found that users who receive notifications through text messaging were the most engaged.

Contents

1	Intro	oduction	4
2	Bacl	kground and Related Work	5
3	Des	ign	8
	3.1	The User Experience	8
	3.2	Experiment Design	13
	3.3	Software	13
	3.4	Data Collection	15
4	Ana	lysis and Evaluation	17
	4.1	Preliminary Analysis and Evaluation	19
	4.2	Analysis and Evaluation	23
5	Challenges		
6	Conclusion		
7	' Future work		

List of Figures

1	The formation of the collaborative economy.	7
2	Description of the collaborative consumption model.	7
3	The types of transactions with the collaborative model.	7
4	The main page displays all available tasks	9
5	The user can select a task to see detailed information about the task	9
6	A sample profile page	11
7	The dashboard helps keep track of the user's tasks	12
8	List of events that a user can fire. Events highlighted in red best fit my description of 'en-	
	gagement'	16
9	Preliminary Analysis	18
10	Distribution of user's notification method	19
11	Distribution of events per week, grouped by notification type	21
12	Aggregate number of events per notification type, grouped by week since app's release \ldots	21
13	Distribution of events by notification method	22
14	Distribution of 'engagement' events per week, grouped by notification type	24
15	Aggregate number of 'engagement' events per notification type, grouped by week since	
	app's release	25
16	Analysis of the distribution of 'engagement' events	25

1 Introduction

Software and the growth of technology have significantly changed the lives of many people. The ability to speed up the different processes within our daily routine has driven a lot of economic attention towards the technology industry. Companies have started taking their products online, taking advantage of the benefits associated with location free utility, and have benefited from the wide spread access to their products. Asynchronously, many companies have been created solely to sell products on the online market place. Software and its many facets of innovation have undoubtedly caused pivotal growth in the overarching global economy.

The growth in the exchange of ideas, products, and services through software and the Internet has led to the creation of collaborative consumption - an economic model based on sharing, swapping, trading or renting products and services which enables access over ownership[2]. By reinventing not just what we consume, but how we consume, a lot of companies can leverage software over the Internet in order to generate revenue. Figures 1,2 and 3 illustrate the different types of collaborative consumption transactions and how they are formed within the economy today[2].

The second decade of the Twenty-first century has seen a rise in the number of start-ups based off the collaborative consumption model: Uber, a popular online transportation network that replaces taxi drivers with an on-demand part-time driver is an example of such a company. The collaborative consumption model serves as the company's entire business framework. There have since been a surge of various other start-ups using the same business model but offering different services, whether it be an on-demand tailor, masseuse, or even a delivery-boy. The possible outlets for a company using a collaborative contribution model are immense. However, not all of these companies have done well in the market place. Why is it that not all of these ventures are successful?

TaskRabbit is a peer-to-peer start-up that allows users to find people in their neighborhood willing to carry out a task in exchange for payment. From a financial standpoint they have managed to procure \$37 million in funding, have strong investor backing, and are a strong contender within the peer-to-peer economy. College Labor is a similar start-up that competes with TaskRabbit by outsourcing labor; however, College Labor doesn't share the same success within the market place as TaskRabbit. Both ventures use the same business model where users post tasks that other users can complete to earn money. However, one is more popular than the other and thus able to earn more revenue and grasp market power. The same pattern has continued among many start-ups in the software industry and begs the question as to why one succeeds while others do not. With so much funding going into peer-to-peer companies and only a few managing to gain economic traction among a high failure rate, it becomes very important for venture capitals to understand the different attributes and characteristics that build a successful peer-to-peer startup.

This paper investigates the performance of various notification methods within software applications. Specifically how these notification methods promote collaborative consumption. By determining which features of user-software interaction generate more user engagement, entrepreneurs can use this study to find the optimal balance between preference and performance when designing software that follows the collaborative consumption model.

2 Background and Related Work

Economically developed countries have become increasingly dependent on software in the last couple of decades. From the alarm clocks that wake 9-5ers up in the morning to the social media and telecommunication outlets we use to communicate with people in our daily lives. Software is everywhere; software provides an effective interface between the user and the many electronic devices we rely on so much. Society is moving to a world that is solely run on software, even for the most menial jobs as it simplifies daily tasks to allow the consumer to achieve more and be efficient with their time.

The growth of the Internet has helped expedite the worlds dependence on technology. Markets have shifted to a technological medium. Basic trading, accounting, and organizational logistics have all become contingent on software's ability to provide information and perform the necessary calculations. The growth in the demand and dependence of online software has foreseen the creation of the marketplace of software that is designed to provide services to its end users.

Google is an example of company that produces software that provides a service to its users. The company started off as an online search engine and has developed into an Internet mogul that provides

electronic messaging, mapping, and a plethora of other business solutions. There are companies such as Facebook and Twitter as well, that have utilized the power of software, harnessed the connectivity of the Internet to create profitable companies aiming to connect people from all over the world [4]. A world everyone can share news, ideas, and local knowledge all in the pursuit for public good.

Collaborative consumption and the notion of a "sharing economy" are very recent phenomena that have come about with the growth of the Internet and social media. Despite the recency of both collaborative consumption and the sharing economy, there is a lot of research to back their existence. Russell Belk, an economist from York University has published several papers on his research on collaborative consumption and its effects on the economy. Belk describes the growth in popularity of collaborative consumption as a consequence of the rate of return associated firms pulled. With minimal initial investment, Belk describes firms that dominate the sharing economy as being 'money printers'[1]. It therefore, isn't surprising that a lot of attention has been going into this model and that research continues in order to understand its long-run implications.

Other works have also focused on the collaborative consumption model and its implications within the sharing economy. The growth of software applications and their resultant extension of the collaborative consumption model have led to large scale accounting issues within the economy. Current regulations haven't yet adjusted to this new form of disruption. The lack of regulations within the sharing economy has created an imbalance among firms, as firms that haven't entered the sharing economy have seen a loss in consumer spending but yet still face existing regulatory impositions[3]. Research into notifications and other attributes that stimulate user engagement in software would provide substantial information to firms currently affected by this imbalance. Information which can be used in the gateway into the sharing economy and provide useful insight for firms planning to regain their market position.



Figure 1: The formation of the collaborative economy.



Figure 2: Description of the collaborative consumption model.



Figure 3: The types of transactions with the collaborative model.

3 Design

The foundation of this research project would be a website that conforms to the collaborative consumption model. The web application in itself would be a peer-to-peer application that allows members of the Union College community to list offers for tasks they need completed, such as a ride to the airport or fast food delivery. Likewise, members have the alternative option of accepting available tasks that they are willing to do. Tasks will be split up based on the category of the task, and users have the ability to select which type of category of task they want to be notified about. By modeling an application that uses the collaborative consumption model, the experiment monitored the ways users interacted with the application and studied whether different forms of notification affected the user's participation.

3.1 The User Experience

When first entering the website the user is prompted to create an account, or log in if they already have an existing account, in order to gain access to the site. Ensuring that the user has an account before giving access to the platform helps create a certain level of security as the application only allows users with the college email address to sign up. The user would also have to submit a password, phone number and their full name. Having only users from the same college not only helps minimize fraudulent offers but also prevents spam and other bot-type malicious threats.

Once the user has successfully signed in, they are redirected to the main page where they can browse tasks that are currently available. The most recently available offers will be placed at the top. Figure 4 illustrates how each offer is placed with its title, a snippet of its description, and how much the offer is worth. If interested, the user has the ability to select the task and will be routed to a page that contains detailed information about the task, as seen in Figure 5. From here the user can choose whether they want to accept the offer or ignore it and keep looking for other available tasks. If the user chooses to ignore the task they would be routed back to the tasks listing. However, if they were to accept the task, the task would be marked as 'in progress', removed from the list of currently available tasks and assigned to the user for completion. The user who has accepted the task will from then be referred to as the 'Taskee', and the user who posted the task as the 'Tasker'.

Available Tasks



Figure 4: The main page displays all available tasks



Figure 5: The user can select a task to see detailed information about the task

Tasks have four different states. Tasks are active the moment they have been submitted by the user and up until they have either expired or have been accepted by another user. Once accepted the task moves to the 'in progress' state, whereby only users who have either posted the task or accepted the task can see and engage with that specific task. When the task is 'in progress' the employee has the option of either dropping the task, if they were unable to complete it, or have the option to mark it as being complete once they have finished. If they drop the task, the task will revert back to the 'active' state, and the employer will be informed of this change. On the other hand, if the task was to be marked as being completed, the task will move to the 'task complete' state whereby the task's employer has to confirm the successful completion of the task. The task's employer now has the option to confirm or decline completion of the task. If the employer declines the completion of the task, the task is reverted to the 'in progress' state and the employee is notified of its unsatisfactory completion. If the employer was to confirm that the task was indeed completed satisfactorily, the task will be move to the 'completed' state. At this stage the employer would have money from their bank account withdrawn and routed over to the employee.

From the main page users also have the option to post a task that they need help with. By navigating to the new tasks section the user will be given a form where they could enter the task's description. They would need to supply a title, description, category, price, and expiration date for the task they plan to submit. All inputs will have to go through validation checks to ensure that the task is submitted properly.

Users also have a profile page that allows them to keep track of the settings pertaining to their account, as seen in Figure 6. This is where they would be able to handle their payment settings. They would need to set up a payment method in order to post tasks. They would also need to designate a payment destination in order to start accepting tasks, as the application needs to know where to route the funds once they have successfully completed a task. This page will also allow users to change their phone numbers and select which category of tasks they want to be informed about.



Figure 6: A sample profile page

There is also a dashboard page that helps track the tasks that the user has engaged with. The page filters tasks based on the current state of the task, and sorts them by when they were most recently modified. Each task has an image that is indicative of the current state of the task, which serves as a visual cue whenever there is a change in the tasks state. This is illustrated in Figure 7. The website also has a "How it works" page which serves as an instruction manual for new users. This section will map out the value of the task's image with the task's status, as well as illustrate the entire work flow of the website.

Completed tasks awaiting your confirmation

Your tasks that are currently in progress

Figure 7: The dashboard helps keep track of the user's tasks

3.2 Experiment Design

The core purpose of this research experiment is to study the effects of disruptive notifications on the user's use of the application in relation to the types of notifications the user receives. Moreover, this experiment looks into the variance between disruptive and non-disruptive notifications. The term 'disruptive' describes the type of notifications that interrupt the user from their current state in order to deliver some form of information. This experiment categorizes text messages and emails as disruptive notification methods.

In order to conduct the study this experiment assigned users to different subgroups. When the user object is created, each user is assigned to one of three notification groups in a round-robin fashion. My placing users into each subgroup in this fashion, this experiment can roughly get an equal number of subjects within each subgroup. Each subgroup was either sent notifications by email or by text messaging. The third group would be the control subgroup, for baseline comparison, in which users in the group didnt receive notifications and are left to seek out changes on the website themselves.

It is important to note that this research focuses on a real life implementation of an on-demand service application. Users would be given the option to change their notification method and move between subgroups. However, they were only allowed to make the change after completing the registration process at which point they had already been assigned to a subgroup with a specific notification method. Allowing users to switch groups was beneficial as it provides additional insight into the notification preferences of users utilizing the application. Moreover, by not restricting users to a particular notification method the experiment was likely to retain more users, as users would be free to adjust their settings as they please. The same paradigm exists within web applications in production, and the findings using this approach would be beneficial in the context of production level applications.

3.3 Software

The web application is built on the Meteor JavaScript framework. Meteor was chosen as the framework of choice due to its implementation of web sockets on both its frontend and backend. This allows any slight change on the server or database to be automatically received on the client without the need to poll the server and wait for a reply. JavaScript is single threaded and prone to extended process wait times.

By using the Meteor framework, the benefits of web sockets ensured that information on the website is always up to date. This feature is important as the website deals with payments. Moreover, it significantly improves the fluidity of the user experience when using the application.

Cloud based web services have been integrated within this application. A company called MongoLabs hosts the application's database. Meteor is packaged with MongoDB as its primary database and Mongo-Labs has good support for MongoDB and other NoSQL databases. Moreover, this application is hosted on Heroku, a cloud application platform that provides platforms as a service. The Meteor framework runs on the Node.js webserver, and Heroku has inbuilt Node.js application containers which made deploying a Meteor application on Heroku quick and simple.

The web application makes use of many third party web services. Notifications are sent from the application through third party APIs and libraries. Twilio is a company that provides an API in order to send text messages. The call to the API whenever a task is posted is done asynchronously, as JavaScript is indeed single threaded and the application wouldnt stall with multiple wait times when multiple method calls are made at the same time. MailChimp and Mandrill are two services that this application uses to handle email notifications. MailChimp allowed the application to make and use dynamic emails: emails that have a standard template with the ability to merge values from variables into placeholders in the template. This created a uniform email for each different type of email that the application sent out. Mandrill is the name of MailChimp's library that integrates with Meteor in order to process the distribution of emails. MixPanel and Google Analytics are two other web services that are integrated into the website. These services are used to collect user data and monitor their engagement. Google Analytics works as a standalone tool, once it's integrated within the application, it essentially captures all the information about the user. MixPanel's API is a bit more detailed than that of Google, in that it is based off event triggers. Event triggers allow the study to measure custom occurrences on the website such as when a task is completed, and information about a user as they trigger a particular event.

Payment on the web application was handled by an online payment gateway provider called Braintree. Braintree was chosen as the payment gateway as it allows payouts to the Taskees on the app to process within one-to-two days after the task's completion. Furthermore, Braintree provided an extensive API which allowed the web app to verify that Taskers had funds available before they were able to post a task. Once the task has been confirmed by the Tasker, a call would be made to Braintree's API to the wire the funds from the Tasker's account over to the Taskee. Braintree is also the payment gateway for companies such as Uber and AirBnb and thus had a lot of experience with apps that followed the collaborative consumption model.

The web app had to follow Union College's Code of Conduct as well as a compliance checklist set forth by Braintree. The app needed to make sure that users weren't posting tasks that were deemed illegal or against school policy. This was achieved by using regular expressions to filter tasks before they were posted to the public. If a task contained a key word that was predefined as 'not acceptable', the task would be flagged and routed to the system administrator who would review the task before it was published to the public. If a user was to submit a task that was illegal in nature, they would receive a warning from the administrator, and their user object on the app's database would be updated to indicate that they've been warned. A second attempt to post an illegal task would have resulted in their profile being blacklisted and they their access to the app would have been revoked.

3.4 Data Collection

Data collected in this experiment had to encapsulate the user's engagement with the web application. This is done mainly through Mixpanel, the analytics software. The primary metric chosen for this experiment was user events. Every action the user performed on the site would fire an event trigger on the analytics software's API. Events captured the time, user, and event description. 'Posted a task' and 'Viewed profile page' are examples of events that could be fired. Figure 8 illustrates possible event triggers that a user could fire on the app. Additional information about the user and the user's session, such as browser and OS version, are also sent with each event.

Viewed current Tasks Page	Viewed landing Page	Viewed howItWorks Page	Viewed venmo Page
Signed in	Viewed logMeIn Page	Viewed dashboard Page	Payment Destination Created
Signed Up	Dropped Task	Posted a Task	Signed out
Viewed verifyEmail Page	Viewed signMeUp Page	Viewed newTaskForm Page	Accepted Task
Viewed profile Page	Logged In	Update Notification Settings	Completed Task
Confirmed Task Completion	Updated Phone Number	Password has been reset	Viewed contact Page
Viewed terms Page	Viewed forgotMy Password Page	Payment Destination Updated	Deleted Task
Removed Payment Method	Viewed add Payment Method Page	Payment Method Created	Viewed home Page
Viewed register Taskee Page	Viewed taskDetails.:_id Page	Marked Task 'in progress'	

Figure 8: List of events that a user can fire. Events highlighted in red best fit my description of 'engagement'.

This experiment monitored the number of accepted tasks as a secondary metric. The sole purpose of the website was to allow users to outsource their tasks to other students on campus. This metric allowed the experiment to gauge the website's level of success per subgroup.

The ability for users to change their notification method means that data collection would have to monitor this change as well. When collecting data on the user's engagement, each data point will contain meta data about that particular user. By knowing the user's notification method every time they interact with the website, this research would be able to monitor changes in the user's notification settings. The study can then seamlessly attribute each interaction data point to the correct subgroup.

4 Analysis and Evaluation

Data was collected over seven weeks following the app's release to the public. There were two primary data sets for the analysis. There was an 'events' data set that contained information on all events that occurred on the app, and a 'people' data set that contained information about all verified users who used the app. Over the seven week time frame the app had a total of 430 people sign up, verify usage, and fire a total of 13,123 events. There were a total of 65 tasks posted on the site, of which 45 were accepted. Preliminary observation shows that 37 of the 45 tasks were accepted by users in the text message subgroup, 7 from the email subgroup and the remaining 1 task was accepted by the no notification subgroup. Figure 9 shows preliminary analysis of the data.

Notification Type	Task's Accepted
Email Notification	7
No Notification	1
Text Notification	37

Notification Type	Task's Completed
Email Notification	5
No Notification	4
Text Notification	14

Figure 9: Preliminary Analysis

Chi-Squared Test				
Notification Type	Number of People	Percent of population		
Email Notification	140	32.6%		
No Notification	103	24%		
Text Notification	187	43.4%		
p value = 0.006195				

Figure 10: Distribution of user's notification method

4.1 Preliminary Analysis and Evaluation

The first step was to see the current state of the population and observe the distribution of people between the three notification types described in Figure 6. At the end of the seven-week data collection period, 43% of the current population is now in the text notification subgroup, 32% in email, and 24% in no-notification, as illustrated in Figure 10. It is important to remember that the experiment was designed to have equal-sized notification group prior to people switching notification groups. Initial observation shows that there has been some movement between groups. However, we wanted to know if this change was a sign of preference or whether this was a result of some noise in the data and resulted in trivial reasons for a change. Using a Chi-Squared test and comparing the current distribution with that of an expected even distribution can help determine this. Results, as seen in Figure 10, determine with 99% confidence that there is evidence that the current distribution is different from that of the initial equal distribution. This illustrates that there is some activity in the app - in that people are consciously making a change to their notification method. It is noteworthy to point out that users are predominantly switching to text-based notification.

The next step was to aggregate all events and observe how they were distributed between notification types. Each event has a given a property that determines the notification method of the user who fired it. Events were then grouped by their notification type and compared across time as seen in Figure 11. It was interesting to note that the distribution of events was almost equal among text and no-notification, while a small number of events were fired from users under the email group, as illustrated in Figure 12. The majority of weekly events occurred from those in the text notification subgroup, apart from the second week. However, it is noteworthy to mention that the second week saw a significant spike in the number of events from those in the no-notification subgroup.

Figure 11: Distribution of events per week, grouped by notification type

Figure 12: Aggregate number of events per notification type, grouped by week since app's release

	Chi-Squared Test				
Overall average number of events per person over seven weeks			3	0	
Notification Type	Number of People	Actual Number of Events	Average events per person	Expected number of events	
Email Notification	140	2690	19	4272	
No Notification	103	4913	47	3143	
Text Notification	187	5520	29	5706	
				p value $< 2.2 \times 10^{-16}$	

Figure 13: Distribution of events by notification method

Similar to the population data, we wanted to know whether this distribution was significantly different to that of a theoretically expected distribution. This was done using the Chi-Squared test, as seen before, from which we can state with 99% confidence that the distribution of events is statistically different from that of an expected even distribution.

It was interesting to see the differences in people's current notification type in relation to the distribution of events. The no-notification subgroup has the least number of users, yet created the second highest number of events. To delve deeper into the data, we set out to compare the averages in events per subgroup. Furthermore, we compared the average number of events per subgroup with the average number of events per person throughout the entire app. Figure 13 contains the output of that comparison. It was interesting to see that the no-notification subgroup's events per person was higher than the apps overall average of events per person.

From the comparison of averages, we asked ourselves: What we would expect given that all notifications produced the same level of events per person? This question was important as it helps conduct a test of our original hypothesis. This question would determine whether the current findings are a result of noise in the data, or whether notification type significantly impacts the average number of events per person. This question can be answered using the Chi-Squared test as before, and results are in Figure 13. This time however, instead of comparing the actual number of events per subgroup to the even distribution of subgroup, we would have to compare it against the expected number of events for that particular subgroup. We can deduce the expected number of events per subgroup as the product of the number of people in that subgroup and the average number of events per person overall. The results from this analysis tell us with 99% confidence that notification type is strongly correlated with the average number of events per user.

4.2 Analysis and Evaluation

In retrospect, we were able to reject our null hypothesis when using all of the collected events. However, this research is looking at engagement from the perspective of notification preference and the real life implementation of an on-demand service that models the collaborative consumption model. Figure 8 lists all the possible events that could have taken place on the app and preliminary analysis so far has accounted for all of them. However, not all of these fit our definition of the term 'engagement'. It would thus be more effective to carry out the analysis using events that best fit our definition of engaged.

To carry out a more effective analysis we narrowed the events data set to only include events that best portray user engagement. These events are colored in red in Figure 8. The included events were those where the user was attempting to find information about a task or when the user has interacted with a task.

The initial step was to compare the distribution of events that best describe engagement. It was interesting to see how the distribution of the filtered data set compares to that of the entire data set. Figures 14 and 15 illustrate a visual representation of the filtered data set. Once again most events occurred by users who were in the text notification subgroup. However, it was interesting to see that number of events from those in the no-notification subgroup wasnt as high as those in the text subgroup, but rather very similar to that of those in the email subgroup.

Figure 14: Distribution of 'engagement' events per week, grouped by notification type

Figure 15: Aggregate number of 'engagement' events per notification type, grouped by week since app's release

Chi-Squared Test of 'Engagement Events'				
Overall average num	ber of events per person o			
Notification Type	Number of People	Actual Number of Events	Average events per person	Expected number of events
Email Notification	140	1154	8	1734
No Notification	103	1343	9	1275
Text Notification	187	2829	15	2316
				p value < 2.2 x10^-16

Figure 16: Analysis of the distribution of 'engagement' events

The next step was to compare the average number of events per person in each subgroup. Similar to the entire data set, the text subgroup had the largest share of events while the email subgroup had the least.

With the notable differences between the filtered data set and the entire data set, we again asked ourselves: What we would expect given all notifications produced the same level of events per person? Similar to the preliminary evaluation of the entire data set, a Chi-Squared test was conducted in order to determine the significance of notification type on the number of events per person. As seen in Figure 16, we can deduce that notification method is strongly correlated with the average number of events per user. Particularly, when using a subset of events which we categorized as the events that best describe our definition of the term 'engagement'.

5 Challenges

There was a drawback in the data collection that should be acknowledged. The experiment's design entailed that users be assigned a notification method during account creation. The implementation of this feature was done in a way where a global variable kept track of the number of verified users. The global variable incremented by one each time someone signed up and verified their email address. Users were then assigned a group based on this variable using a round robin approach. However, there were several instances during the apps life cycle where the server systematically reset itself. This was a protocol that the Platform-as-a-service provider had in place for processor and memory management. Every time the server reset, the global variable reset with it. This led to the uneven allocation of users to notification subgroups. In order to assess the effects of this drawback, we looked at the distribution from which users were allocated subgroups, and conducted a Chi-Squared test. The results from the Chi-Squared test tell us that the current allocation of notification types is not statistically different from a distribution where everyone is allocated a notification method evenly. Nevertheless, the uneven distribution is not negligible and must be taken into account when making any inferences from this research project.

Trade-offs had to be made when designing the experiment. Some of the trade-offs I made came at a price as they introduced validity threats into my study. The ability for users to switch notification groups is an example of a validity threat within this project as results could be tainted by users who have moved across notification groups. As mentioned earlier, this experiment was designed to build a realistic on-demand service app whereby users have the choice to be notified as they wish. One could argue that forcing users to specific notification groups introduces a validity threat in itself, as users may not engage with the app as they don't like the notification they've been assigned. Nevertheless, a trade-off was made and it's important to acknowledge the relative validity threat.

The population of study is also a validity threat within the experiment. The app was introduced and exclusively for the Union College community, whereby the audience was mainly 18-21 year old college students. Although there were faculty, staff and other members of the Union College community using the app, the uneven distribution in demographics questions the overall validity of my results. One could argue that notification type had a significant effect on engagement, as one of the notification types may have been a common channel of communication for that demographic.

The user's experience when using the app differed depending on the user. The way the app looked on an iPhone wasn't necessarily the same as the way it looked on an Android device. The user's device, whether it be a laptop or mobile, even the user's default browser, led to differences in app's user experience. The app was optimized to be compatible with as many devices as possible, but the differences in user experience when using the app brought in another validity threat that is not negligible. A skeptic could argue that some users weren't as engaged as the user experience on their device wasn't as seamless as it was on another user's device.

6 Conclusion

From the results of the data analysis I can say that notification type does correlate with whether user's are engaged or not. The distribution of events was significantly different to that of an even distribution which we had expected given that we designed the experiment to evenly allocate a notification type to registered users. Furthermore, the expected distribution was different to that of the distribution we'd expect if all notifications produced the same level of engagement. Results also indicate that users who received notifications through text messages were the most engaged among the different subgroups. However, findings from this project would be made more reliable if this experiment was repeated while ensuring that

users are actually evenly allocated notification types.

7 Future work

It would be interesting to see how users would have responded to a mobile app instead of a web app. The use of push and device-specific notifications would have provided insight into other forms of disruptive communication. It would also be interesting to see a user by user comparison of engagement, whereby each user's activity is compared before and after each change in notification type. Future work should ensure that user's are allocated to a notification group evenly. It would also be good to open this experiment up to the general public and not limit it to users from a specific demographic. Furthermore, it would be great if this experiment ran for a longer period of time and could capture more data. This would have allowed for more extrapolation of the data.

References

- Russell Belk. You are what you can access: Sharing and collaborative consumption online. *Journal of Business Research*, 67(8):1595–1600, 2014.
- [2] Rachel Botsman. The sharing economy lacks a shared definition. Collaborative Consumption, 2012.
- [3] Rachel Botsman and Roo Rogers. *What's mine is yours: how collaborative consumption is changing the way we live.* Collins London, 2011.
- [4] Todd A Finkle. Corporate entrepreneurship and innovation in silicon valley: The case of google, inc. *Entrepreneurship Theory and Practice*, 36(4):863–884, 2012.