|  |
| --- |
| **Senior Design: Adio** |
|  |
| **Group Number** **19**  **Team Members:**  Sneha Advani, Bharath Jaladi, Arjun Lal, Romit Nagda, Sneha Rampalli  **Advisors**: Sangeeta Vohra |
|  |
|  |
|  |
|  |

Abstract

|  |
| --- |
| 000  001  002  003  004  005  006  007  008  009  010  011  012  013  014  015  016  017  018  019  020  021  022  023  024  025  026  027  028  029  030  031  032  033  034  035  036  037  038  039  040  041  042  043  044  045  046  047  048  049 |

This document serves to describe the work done by Group 19 on Adio, an application for Uber and Lyft drivers to play their music during their shifts interspersed with location-based advertisements at a frequency they set.

Motivation & Goal

Advertisers, especially small businesses, want and need cost-effective ways to reach high-potential consumers. Additionally, there are currently cost-barriers for entry into advertising mediums such as radio for smaller businesses. Moreover, consumers want and need more relevant and effective methods to discover and connect with businesses that are easily accessible to them.

Most Uber and Lyft drivers play music on the radio, through which ads earning revenue for radio stations, play every few minutes. Thus, our team wanted to take this advertising opportunity and improve upon it while also providing another source of income for these drivers.

No formal survey was conducted, but each team member spoke to Uber/Lyft drivers to gauge interest from their end. Specifically, we gave a one-minute pitch of our idea and asked for their feedback on the following points: (1) if they play any music while they’re driving their riders to see if most drivers would play ads during each trip, (2) if they would use the product, given the existence of products with similar goals, (3) if drivers would be interested in beta-testing our service. We received positive feedback from the drivers, as the use of this service requires little effort on their end, besides downloading the application. The only concern on their end was that the user experience could be reduced because of ads. However, drivers can set the frequency of the ads, and given the existence of Uber/Lyft ratings, the driver will be incentivized to balance the number of ads playing during each ride with the user experience.

We also spoke to our networks of family and friends to both hear their thoughts on our idea and explore any possible concerns they would have. The feedback we received was generally positive, especially amongst college students. Given that students on campus call rides to Center City Philadelphia for downtown events, it gives them an idea of activities in the area. Adults also expressed an interest in this idea, for example, when they are traveling on business trips and want to explore the area around them during their free time.

Add M&T stuff here

Below, we provide specific details on the stakeholders and value proposition of our product.

Related Work

To determine how to best differentiate ourselves, we did a great deal of market research to identify several competitors, listed below. We categorized competitors as follows.

**Tablet-based Rideshare Advertising:**

1. Octopus (<https://www.playoctopus.com/)>: provides free tablets to drivers that play trivia games and location-based video advertisements.
2. Vugo (<https://govugo.com)>: provides in-car entertainment in the form of video games, apps, film shorts, sports, and news.
3. Surf (<https://ridewithsurf.com/)>: provides interactive tablet-based entertainment for rideshare, such as videos, music, podcasts, and live radio.

**Rideshare Billboard Advertising:**

1. Firefly (<https://fireflyon.com/)>: provides lighted billboards for the top of cars that cycle through different targeted advertisements in the area. Firefly pays drivers about $300 per month.
2. Halo Cars (<https://www.halocars.co/)>: provides LED billboards for the top of rideshare vehicles. Halo Cars pays drivers about $400 per month.

**Music Advertising Phone Application:**

1. Steereo (<https://steero.com/)>: creates playlists for drivers with sponsored music from independent artists paying to promote their songs. Steereo pays drivers once they have reached a minimum of $100 in earnings.

Given the current competition in our target market, we decided to differentiate our product in several ways that we feel are most beneficial to businesses, drivers, and riders. Firstly, our product is purely software-based and thus eliminates any hassle that may come with hardware solutions (such as billboards). Secondly, we integrated with the Spotify API to deliver ads that are interspersed with predetermined playlists set by drivers, which makes the experience more personalized. Thirdly, our solution allows for a passive way of making income for drivers. Lastly, our product will specifically be geared towards smaller businesses who typically do not get the exposure that larger and more well-established franchises receive.

We believe that these points make our product much more seamless and individualized than other solutions currently in the market. By catering to all stakeholders involved (small businesses, drivers, and riders), our product enhances the user experience on all ends.

Add M&T stuff here - Market research & Competition

Components (Completed and In Progress)

|  |
| --- |
| 100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149 |

|  |
| --- |
| 150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199 |

There are two main components to the application: a web component, developed with JavaScript, EJS, CSS, and AWS, and a mobile component, developed with Swift, SceneUI, and several relevant APIs. At a high level, the mobile app allows drivers to sign in and set various parameters regarding the advertisements and music that they play, such as volume and frequency. In addition, drivers can navigate to a dashboard that details the amount of money made by drivers over specified timeframes, such as the past month or the past year. The web app is meant for advertisers and allows them to sign in, upload audio advertisements, and set the center and radius of the area their advertisements will be targeted towards (usually centered on physical locations of businesses).

Below, we list out the specific sub-components that comprise the mobile app and the web app, descriptions of their functionality, as well as their current status (**completed** or **in progress**).

**Mobile App**

(1)  Initial driver interface. This is essentially the home screen that allows the user (driver) to open the app and begin use of our product. **Completed.**

(2)  Settings interface. This view allows drivers to calibrate the settings that dictate how ads are played e.g. frequency of ads, volume of audio, allowing location sharing, etc). **Completed.**

(3)  Profit dashboard. This view renders a gridded histogram detailing the amount of profit (passive income) made by a driver using our app specifically. In addition, it allows a driver to calibrate the time period over which profits are calculated and displayed eg. the previous year or the previous month. **Completed**.

(4)  Stopping/resuming a song with ads. This feature allows the driver to pause a song that is currently playing from Spotify, play an ad, and then resume the song. The ad is played through a music player created using Apple’s AVFoundation library. This particular feature is located on the driver interface described above. **Completed.**

(5)  Integration with Spotify. As referenced above, our product integrates with the Spotify API to authenticate drivers with their personal accounts and play playlists that are actually curated by them, as opposed to being created by any third party. **Completed - see appendix for details.**

(6)  Playing of ads after a set number of songs. This feature allows the driver to connect to Spotify and only play an ad once a set number of songs have been played, creating seamless transitions between music and ads. **In progress - see appendix for details.**

(7)  Integration with database. This component allows for the pulling of specific songs our of our database. **In progress - see appendix for details.**

(8)  Driver statistics computation. This feature will provide specific calculations of statistics for drivers regarding profit/the actual rides driven and will then render the statistics to drivers. **In progress.**

(9)  Terms and conditions. These will detail terms of use on our app as well as what qualifications a user must have to join. **In progress.**

(10)  Geolocation algorithm for ad-selection. This will search through the entire database of ads that have targeting regions that encompass the current location of the driver and then selects these ads and queues them up. **In progress - see appendix for details.**

**Web app**

1. Account creation. This allows a new user to sign up and create an account with an email, company name, first name, last name, and password. **Completed.**

|  |
| --- |
| 200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249 |

|  |
| --- |
| 250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299 |

|  |
| --- |
| 300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349 |

|  |
| --- |
| 350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399 |

|  |
| --- |
| 400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449 |

|  |
| --- |
| 450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499 |

|  |
| --- |
| 500  501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547  548  549 |

1. Password encryption. This feature uses sha 256 to create an expectedly unique 256 bit signature for a user’s password, which enforces security for our users. **Completed.**
2. Verification of login credentials. This ties in with the above to verify that an email/password pairing does indeed exist in our database when a user is trying to login. **Completed.**
3. Map window for geo-location. This allows a user to click on a point of their choosing on a Google Maps API driven map. This click-point serves as the center of their ads’ targeting. Additionally, users can specify a targeting radius here. **Completed.**
4. Upload of audio files. This feature allows users to upload audio files (likely .mp3) that contain individual advertisements. This uploads the advertisement to S3 for later access and stores the ads’ metadata in dynamo db. **In progress.**
5. Evaluation of Completed Components

Below, we list out the various completed subcomponents of our product as well as the methodologies we used to evaluate them.

**Mobile App**

In general, we used a similar protocol to test every feature we implemented on the mobile application. We first tested it on different devices on the local XCode iOS simulator including an IPhone 11 Pro, IPhone XR, IPhone 8 and an IPhone 6. Then we tested the interface on our devices with different versions of iOS (iOS 9.3, iOS 12.4, iOS 13.2.3). Our final step of evaluation was to show the application on our phones to possible users and have them interact with the specific feature we were evaluating. Below we talk about specific evaluation points we were looking for.

**(1)  Initial driver interface.** Is it user friendly and intuitive to use? Does the driver feel overwhelmed by the initial home screen? After a couple of changes, we found that potential drivers were happy with an easy to use and obvious homepage that wasn’t too cluttered with extraneous features.

**(2)  Settings interface.** Is it easy to find the settings page? Are there settings the driver feels would be important but are not there? Is it easy to understand what each setting controls? The drivers we spoke to appreciated having the ad frequency and volume controls easily accessible so we moved those to the home page. Beyond that, we added a couple more settings like the options to edit account info and change location preferences to make sure drivers were happy with our settings interface.

**(3)  Profit dashboard.** Is it easy to navigate to the dashboard page? Does the driver easily understand the numbers and figures displayed on this page? Here, we got feedback that just a table of numbers was hard to interpret so we opted to put a chart of earnings over a month by day. Further we made the figures for monthly and yearly earnings large, to make it more attention drawing to drivers.

**(4)  Stopping/resuming song with ads.** Does the background music stop when an ad is played? Does the ad play to completion? Does the background music start where it was paused? These points were tested by ensuring the feature worked as it should. 

**(5)  Integration with Spotify.** Can the driver connect to their Spotify account to play music? Is it seamless to connect to Spotify? Can we ensure that the driver doesn’t need to keep logging into Spotify after the first time? Feedback from drivers told us they would prefer it automatically connected to Spotify when opening the application and they wouldn’t want to keep logging in, so we only have the user log into Spotify the first time they use the app. Thus we made sure our application does this and feedback of this approach told us it was the way to go.

**Web App**

**(1)  Account Creation.** We wanted this feature to be as streamlined as possible, given that it is the first thing a user interacts with when he or she uses our app. Thus, we wanted our UI to be intuitive and simple. We asked several friends to look at our home screen and subjectively rate how easy it is to navigate and understand. We received overwhelming feedback stating that the UI was aesthetically pleasing and that signing up was easy to follow. Objectively, we evaluated this component by ensuring that once a user creates an account, our account database is correctly populated. This did indeed happen.

**(2)  Password encryption.** Using sha-256, passwords are hashed, but if a malicious entity figures out this hashing scheme, then every user’s privacy is essentially compromised. To add an extra layer of protection, we created a 16 byte salt that is also hashed in combination with a user’s password, ensuring that figuring out the password hash function is not enough to infiltrate our database. To ensure the correctness of this approach, we observed our databases for consistency. Furthermore, we tried several adverse techniques, posing as malicious entities, to see if we could break our protection scheme, which we could not.

**(3)  Verification of login credentials.** Here, we wanted to make sure that our app would correctly check that a user’s login credentials exist in our database. If they do, then the desired functionality is that a user can successfully login. If they do not, then our app simply rejects the user. Thus, we evaluated this by testing login with existing credentials (of an already created account) and non-existent credentials. Using existing credentials, we were able to sign in. Using non-existent credentials, we were denied login, as desired.

**(4)  Map window for geo-location. TO DO Completed**

1. Demo

Below is a link to a video demo of both our mobile app and web app.

|  |
| --- |
| 550  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597  598  599 |

1. Conclusions and Future Work

Acknowledgments

We would like to thank Professors Ani Nenkova, Sangeeta Vohra, Zachary Ives, and Clayton Greenberg for their help and insights into how to move our project forward.

References

|  |
| --- |
| 600  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649 |

|  |
| --- |
| 650  651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697  698  699 |