

# Passio Go v2

## (BullRunner Destination Stop Finder)

Sai Chaitanya Balli	Jaipal Reddy Kuntla	Manish Kumar Reddy Peta	Praneeth Kumar Reddy Vemireddy	Ujwal Saini
U83920820	U84497179	U57253689	U78965510	U84606858

### ABSTRACT

When we arrive in the USA for the first time, it is hard for the user to fulfill his basic needs like grocery shopping, banking services, and bill payments. So, he uses google maps for searching his destination. Whereas within the USF parameter google will not show the Bull runner bus stops. So, they use an existing application called passigo. When we open the app, it shows us the bus routes and a search bar for searching the bus stops. Here the users face a problem because the app shows its bus stops but it doesn't show us the user's desired destination nearby address. So, we are proposing a solution for this problem by adding a destination icon in which when it is clicked it provides an option for searching the desired location. When we search for the desired location by copying it from google maps the results show the bus route and a bus number.

### 1 INTRODUCTION:

To begin with, every year a lot of people come to the University of South Florida to pursue their education in graduation, under graduation, and doctoral level programs. So, when they want to travel outside for their needs like grocery shopping, restaurants, banking services, Network services, etc. The users will search on google maps but they cannot find the bull runner bus stops on google maps. If they want to travel within the parameter of the University of South Florida they have to use an existing application named passigo. In the application, the user can find the bus stop location, but he cannot find the bus stop near his desired location. The bus routes are highlighted with different colors for a better understanding of the app. When the users open the application, it shows the bus routes with different colors as well as a search bar for finding the bus stop. But it doesn't show the desired location bus stop of the user.

This application is mostly used by international students, MOFIT employees as well as university staff for reaching their destination within the parameter of the University of South Florida.

So our proposal is we design a destination finder icon for the user so that when they click that icon it opens a search bar for finding the destination bus stop. They copy the destination address from

google Maps and then paste it into the search bar of the destination finder in passio-go app. This will check with all the nearby bus stops for the desired destination and output with the bus route and bus stop number. This helps the people for a better understanding of the app as well as they will not be misguided with their assumptions. By this, they can reach their destination safely with proper guidance.

## 2 ESTABLISHING USER REQUIREMENTS

### 2.1 Method:

We gathered both qualitative and quantitative data regarding user satisfaction and their measure of feedback. We conducted a survey, composed of qualitative and quantitative questions. Specifically, we employed a Qualtrics survey utilizing these questions.

### 2.2 User Demographics

In all, 15 users were polled. Two of them were between the ages of 27 and 32, while 13 of them ranged in age from 18 to 24. Eleven of them identified as male, and four as female.

### 2.3 User Requirements

Based on the statistics, it was determined that our users must be college students or employees who are, respectively, between the ages of 18 and 25, and 27 and 32. They ought to be utilizing the Bull runner app and ought to be comfortable with it.

## 3 INITIAL DESIGN:

### 3.1 Current Interface and Solutions

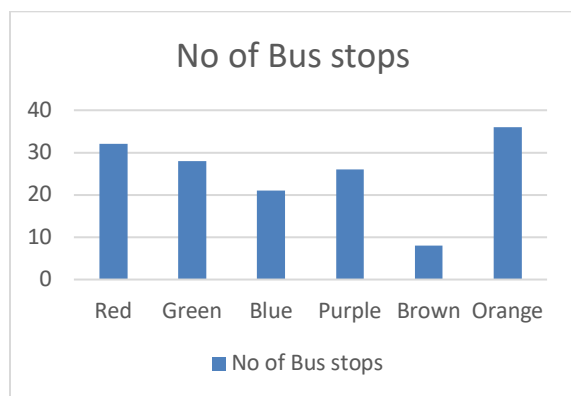
The current interface that we are looking at is the Passio GO app where all the students, Employees of the university of south Florida, and Moffit employees can check the bus routes, at what time the buses will arrive, and which route bus services are available to the certain bus stop. If we open the app, we can observe that all the bus locations are shown using a bus logo and the routes are highlighted with a specific color for that route.

In the current Interface if the user wants to look for a specific location other than the bus stops while clicking on the search icon, he can see only the bus stop locations and no result appears while entering other locations. The current solution to the problem is that the user can use Google maps to search for his destination location, but he cannot find the nearby bull runner bus stop.

### 3.2 Proposed Interface Design:

We had done some research on the other apps like google maps. If we want to go to someplace anywhere in the state or country google map provides the ways, possibilities, etc. as we are international students when we are in Tampa for the first time as we don't even know the necessities requirement places in the town/area. So we initially used google maps and paid extra for cabs and public transport. As we already have the college bus facility (Buller Runner) app. In the app, we already have the route map from where to where the buses move or we can say the basic route maps with the stops have been displayed alongside the bus tracking, this is so helpful for the students to track down the buses and to reach the nearest bus stops for boarding the bull runners. Coming to the User interviews we had asked a few international students about the app, mentioning how could they know the stores, and places they want to visit on their first trip, like going to the mobile store for buying sim cards, groceries stores, restaurants etc. One of the guys stated as they are new, they don't even have credit cards to book a cab and go, so they just used the HART

busses by using google maps, walked for miles to find the HART stops, and even from stops they carried their grocery bags to their homes and just over-stressed themselves. Some random international students like three of them said although they are new, they have some contacts may be relatives or friends, they guided them to all the stores and gave them credit cards which made them easy to get their required products. Two local people who were born/brought up or migrated here many years ago said they already know the locations and bull runner's routes quite well, they also added they know to compare and relate the routes to the original google routes which made them easy. So, from all these interviews and the incidents that occurred for us, we came to one solution adding some facilities to the already presented Bull Runners app. As we planned, we have all the planned app settings in the PPT.



## 4 SYSTEM:

### 4.1 Existing System:

Currently we have apps like uber, google maps. In that firstly the user searches for his desired location from his current location. For example, a person wants to travel from Walmart to Eagles points. In the first case he first uses google maps for the distance he has to cover to reach his destination and for the nearby bus stops. If he finds a nearby bus stop, he reaches there by a walk and he covers the remaining distance by catching a bus. In the second case if he couldn't find any near bus stops, he books an uber for reaching his desired destination. Whereas for the USF students they have a pass for the Bull runner app. So, they choose Passio-go app to reach their desired location.

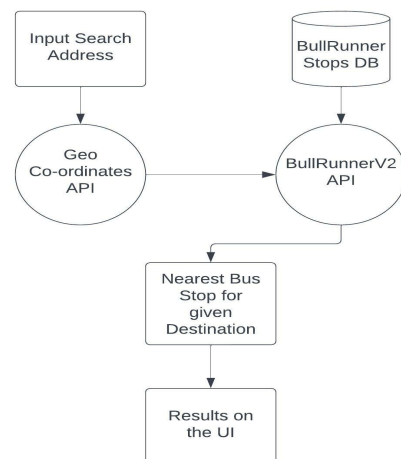
### 4.2 Proposed System:

With this new functionality, we can now conveniently reach nearby locations that are both on and outside of Passio Go bus lines. In order to visit a mobile network store last week because of a network issue, I searched for it on Google, found its location, and compared it to the Passio Go map. After searching continuously for 30 minutes, I discovered the closest stop by comparing the locations and the road numbers that are displayed on the maps.

Then, using our suggested system, we simply copy the geographic coordinates from the closest network provider store and use them to launch the app and our new destination feature copy the geo coordinates there, and it can take 3-5 seconds to compare and display the results, much like a store. is on a certain bus route and makes reference to the Passio Go bus stop number while getting off the bus.

Due to the fact that Bull runners are free for students, this allowed us to make the work easier and even cut down on the time it took to discover the specific sites. If we succeed in implementing this in the Passio Go app, we can do the same for all other apps other than Google, or we can just claim that we can lower the dependency on the simply google

### 4.3 Architecture Diagram:



- The first step, we get the address from the user.
- Then we will be passing the address to the open-source geo-coordinate API.
- This API is going to return the coordinates for the given address.
- Then, these coordinates will be passed to the bull runner go v2 API.
- The bull runner API is going to use the coordinates from the bull runner bus stop database to find the nearest bus stop and return the nearest bus stop to the user.

## 5 IMPLEMENTATION:

### 5.1 Backend:

In the backend of our app, we selected to build a flask API to find the nearest bus stop using geocoordinates. `passigov2/destination finder` – in this flask API, we have templates and statics, which are the two main components of our flask API. Templates are made using HTML/CSS/JavaScript front-end UI files where we show the UI to the user. When writing mobile apps, HTML and CSS are framed into the native app framework. Imagine developing an application that fully displays a webpage within the app and goes to the website specified. JavaScript is frequently used for content modifications, form validation, and picture manipulation. The standard markup language for constructing Web pages, HTML (Hyper Text Markup Language), explains the structure of a Web page. It consists of several elements which specify how the material should be shown to the browser. The labels "this is a heading," "this is a paragraph," "this is a link," etc., are provided by HTML elements.

IOS and Android use constraint-based layouts, which essentially let you establish parameters like a box's width or the distance between its top and bottom. The system will be able to calculate the location and size of each element based on the constraints you provide for each one, so it will appear differently on different screen sizes but will always adhere to your requirements.

Static: This component contains all the vectors, images, and supportive logos. The color, font, size, spacing between items, positioning of elements, background images or colors, different displays for various devices and screen sizes, and more may all be controlled with CSS.

#### Mainapp.py (flask app):

First, the API gets the request in the form of rest, wherein we will get an address or destination. The address which we received from the user must be processed and locate the given address's geocoordinates to ensure that the given address is correct, and we have to return a function that verifies the given inputs address, once after the preprocessing, we pass our input data to the geocoordinate API to extract the geocoordinate of the input address. Here we used position stack rest API for extracting the geocoordinates of the given users. So here, we will get input from the user and preprocess the data for verification. We will forward the address to the position app to rest API for getting the destination address.

### 5.2 Bull Runner Database:

We have collected the data of stop names, stop numbers, and coordinates of the bus stop contain in an Excel document along with the rest of the data. The bus stop names, and control numbers of data is collected from the passigo app, and the coordinates of the bus stops are collected from google maps. Here, we've gathered the latitudes and longitudes of the bus stops for each of the six routes. There are 26 stations on the purple line, 28 stops on the green line, 32 stops on the red line, 21 stops on the blue line, 9 stops on the Brown line, and 36 stops on the orange line. After finding the coordinates of the destination address, we pass the destination address coordinates and database connection to find the nearest stop script. To find the nearest stop script, we first load the existing database in a data frame, then manually check the distance between every bus stop using `gopy.distance` function and finally, we will be returning the stop where the calculated distance is minimum to render this data on the user interface. we must pre-process the data in JSON format, so we convert the data like stop name, stop number, coordinates, and routes in JSON format and will send this JSON as a response to the request.

### 5.3. Front end:

Over here, we have three main components home page, the destination finder, and the result page. On the home page, we will render the bull runner bus map and a navigation bar at the bottom for navigating the map, destination, stops routes, and messages. Over here, we have implemented the destination stack. Once the user clicks on the destination, we will redirect the user from the home page to the destination finder. On the destination finder page, we have a back button and search bar where the user can input the address to find the nearest stop. There is a back button option. By clicking on it, the user will be directed back to the home page if he made a mistake and want to change the destination address.

For the search bar, we have the input text field where the user can enter the destination address. Then we have a find button by clicking on the button so the entered address will be sent as a request to the destination finder flask API. The destination finder API takes the input address and processes the input address, and calculates the shortest distance stop and renders the results of the response of the destination finder API in the result component in the result component of the UI. We will show the routes, coordinates, and stop names and show them on the map button.

When the user clicks on the show on the map button, we will use the nearest bus stop coordinates and redirect the user to the homepage with the pinned stop position of the nearest bus stop for the given destination address.

As we planned to come to the interface part, we separately added a Feature in the already presented app by adding the separate API. Initially, in the interface of the app, we can find the four Features mentioning MAPS, STOPS, ROUTES, and MESSAGES which is quite useful for all USF students. After all the research and issues, we found we would like to add one Feature in between which ends with our API, like finding the best nearest stop for the student as required by searching in the bar separately. All the Designs of the interface are mentioned in the ppt clearly.

## 6 EVALUATION:

### 6.1 Method:

#### 6.1.1 Participant Procedure:

We initially requested that users to install the apk file for our application on their Android smart phones in order to evaluate it. The participants were required to complete the following tasks:

1. Open destination tab in the passio-goV2 android application.
2. Asked the user to enter the destination address in the destination tab.
3. Asked the user to analyze the output/ result/ nearest bus stop details for the given destination address.

We then provided a URL to answer a series of questions. The questionnaire primarily sought input on the functions we have developed.

#### 6.1.2 Equipment:

To install our application participants requires an Android smartphone. We requested participants to complete the tasks on one of our android smart phones where application is already installed, if they didn't have an android phone to use.

#### 6.1.3 Location:

Most of the participants were contacted outside the campus's bull runner bus stops, while just a small number were contacted within the library.

#### 6.1.4 Time:

The participants installed the program and completed all of the activities in around 5 to 6 minutes. After completing the exercises, individuals spent one to two minutes responding to an online questionnaire.

### 6.2 Participants:

When we were going through the process about the people who are using the Passio go app, we classified the people into Three categories:

- i) Students
- ii) Moffit Employees
- iii) USF Faculty and staff

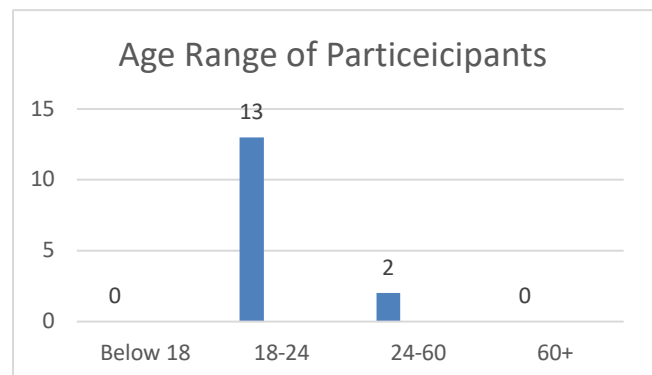
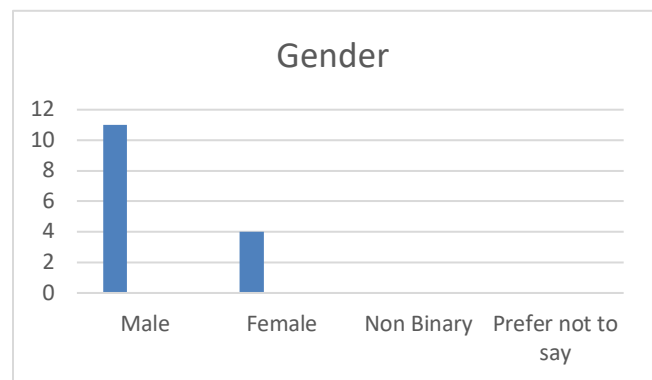
**Group A:** Here we have the Students at USF. Also, again we categorized this group into seven groups: Local Students, International Students, Graduate Students, undergraduate students, PHD Students, On-campus living students, and off-

campus living students. **Group B:** The other group which uses the Bull Runner Busses are private employees at Tampa. They are Citi bank employees, Moffit Hospital Staff and employees. To get feedback from this group, we plan to conduct drives near the Moffit bus stop and the Main bull runner bus stop. **Group C:** USF Faculty, Staff and employees use the Bull Runner busses.

Here, we had a variety of users, including normal Bull Runner users like undergraduate students, graduate students, Aramark employees and Moffit staff. Due to their familiarity with the current app, these users can quickly assess our suggested system.

#### 6.2.1 Age groups:

To examine our application, we enlisted two employees and the rest 13 students of the opposite gender. Most of them are students between the ages of 18 and 25, with the remaining group falling between the ages of 27 and 32. Six of the 15 participants were female, the remaining 11 were male, and one participant was non-binary.



### 6.3 Study Conditions:

We divide the students into study groups, depending on how frequently they use bull runner Busses.

**Group 1:** contains those who routinely board the bull runner using the Passio go.

**Group 2:** contains users who use bull runner buses but doesn't use Passio go application.

**Group 3:** contains users who don't use Passio go and bull runner busses frequently.

#### 6.3.1 Study Conditions for groups:

For group 1, the study environment would be perfect for them to use the program, with everything functioning at its best. As the users are familiar with the passio go application and the Bull runner buses. This group doesn't experience any external stimuli that could affect the study. Study

Study circumstances for group 2: The users use the bull runner buses and don't use the passio go application. Since the users are new to this kind of application, there will be a slight learning curve.

Study circumstances for Group 3: The users are new to bull runner buses and the application. First, we must explain everything about the bull runner buses and the passio-go application to this group. So, there will be a good learning curve for the users in this group.

### 6.4 Metrics:

#### 6.4.1 Speed:

By using the Destination feature, the user can quickly and efficiently find the nearest bus stop to the given destination address. The total efficacy has been increased by 50% as no mapping is involved and no analysis is involved.

#### 6.4.2 Error rate:

So, there are possibilities when the user enters the wrong address, so these scenarios happened in the evaluation and the error rate around 4% in total test/research.

#### 6.4.3 Time to learn:

There is a learning curve for the users unfamiliar with the bull runner buses or passio go application as they have to have some background info about these. Rest everything is simple as the user can enter the destination address and navigate the application.

#### 6.4.4 Satisfaction:

Saving money brings the uppermost satisfaction to the users. Using our destination feature, the user will travel

efficiently and save money. Also, by giving knowledge/details to the users about the bus stop details, the user will feel confident and satisfied.

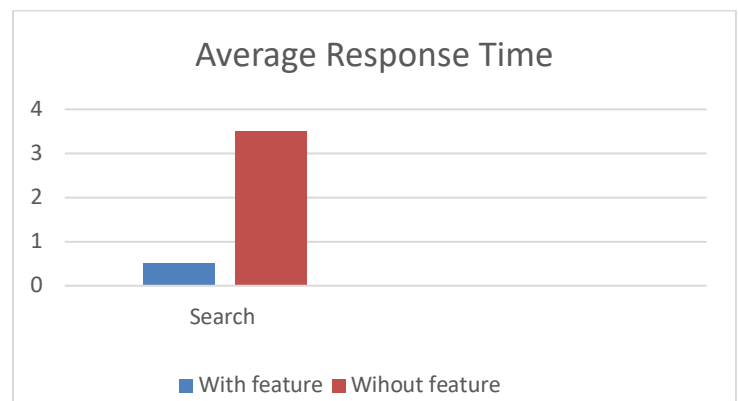
### 6.5 Results:

From our evaluation, we observed the following results:

1. Our suggested method enables users to better manage their time.
2. By using our application, the users felt that they can utilize the bull runner buses even more frequently.
3. By using bull runner buses frequently users can save energy, time and money.

### 6.5 T-Test:

We used a t test to evaluate the participant data because we were only considering two groups of participants. The enhancement features we produced are viewed as independent factors, and the answer search time is viewed as a dependent variable that depends on the independent variables.



## 7 DISCUSSION:

As a result of our studies, users of the application no longer need to be concerned about the new locations they will be visiting along the bus route. When new users want to visit the locations of the usf bus's (Bull Runners) closest stops, the geo coordinates method we used makes it a lot easier for them to do so. which aids in lowering travel expenses.

As a result, when we gave some users access to the app on a need-to-basis, they discovered how interesting it was to be able to locate the closest bus stop to their destination. Even when it is obvious whether a destination is walkable, if it is far, people can use other services or cabs. Evidently, this lowers the cost of travel for users.

Even the users we initially granted access to used the feature for a short period of time when they needed to get to the closest Chase Bank. Previously, they had relied on uber or hart bus services, which can run you up to \$5 to \$15 for a single ride. Their response to using our app is that it's so easy to use, which is similar to the Passio-go app with new features saved them their money. Similarly, they claimed that this application is special and time-saving, and that it is extremely helpful for their day-to-day lives. They claimed that they always used to pay for transportation when they wanted to go to the nearby grocery store.

Additionally, users added a few points to the most recent app update. I'd like to include some customer services where people can provide feedback following either positive or negative incidents. Even when there are two buses on a route, they may occasionally arrive at the same time, costing students and workers valuable time due to careless errors. As a result of these few problems, they lose interest in the apps and services that are specifically designed for people like them and are ultimately useless. As a result, we intend to add more features that enable updating and automatically notify the app's technical team of updates.

### 7.1 Anticipated Outcomes:

We knew exactly what we were going to do and how we were going to execute it from the very beginning of the project. We identified the issue, made a proposal for a solution, created an initial plan for how we would implement the solution, obtained user feedback through surveys to confirm that the current solutions need to be improved, developed a prototype of the initial design, obtained feedback from others (the class as well as the course instructor), decided on the platform on which we would develop the proposal for a solution, and then implemented the coding.

### 7.2 Not Anticipated Outcomes:

We developed an android application and implemented our proposed features. After completing the implementation, we asked participants to evaluate our proposed interface. We were facing an issue while locating the geo coordinates of some specific locations, the participates were not getting the coordinates of the destination address, the geo coordinates API is not able to identify the exact coordinates of the address it is displaying the nearest coordinates to the address which the user have entered because of that the output shows slightly different results.

## 8 CONCLUSION AND FUTURE WORK:

After the Research and the interviews, we can conclude that by adding our design feature to the passio go application, it will reduce the time by skipping the combusive task of identifying the destination bus stop. This will be very useful to existing students/ university Staff and the upcoming international students. We took the references from google maps where they used Geo coordinates will help us to add the new feature. So basically, as we use uber to pick and drop locations we use the same API to add our new feature.

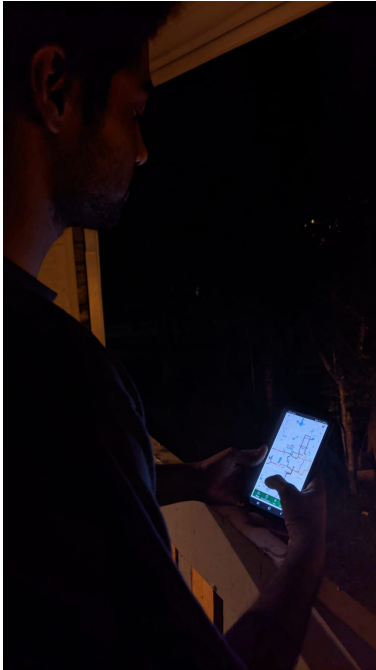
Currently the user has to enter the complete address in the destinations tab the is no auto complete feature. If there exists an auto complete address feature then it would be really easy for the used to use the destination tab.

## REFERENCES

1. S. Guha and N. Mishra, "Clustering data streams" in Data stream management, Berlin, Heidelberg:Springer, pp. 169-187, 2016.
2. S. Wallsten, the competitive effects of the sharing economy: how is Uber changingtaxi, Technology Policy Institute, vol. 22, pp. 1-21, 2015.
3. D. Shah, A. Kumaran, R. Sen and P. Kumaraguru, "Travel Time EstimationAccuracy in Developing Regions: An Empirical Case Study with Uber Data in Delhi-NCR\*", *Companion Proceedings of the 2019 World Wide Web Conference*, pp. 130-136, May 2019.
4. L. Liu, C. Andris and C. Ratti, "Uncovering cabdrivers behaviour patterns from their digital traces", *Compu. Environ. UrbanSyst.*, vol. 34, no. 6, pp. 541-548, 2010.
5. S. Vigneshwari and M. Aramudhan, "Web information extraction on multiple ontologies based on concept relationships upon training the user profiles" in Artificial Intelligence and Evolutionary Algorithms in Engineering Systems, New Delhi:Springer, pp. 1-8, 2015.
6. H. A. Chaudhari, J. W. Byers and E. Terzi, "Putting data in the driver's seat: Optimizing earnings for on-demand ride-hailing", *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 90-98, 2018.
7. Google App Engine  
<http://code.google.com/appengine>
8. Android Developers Reference  
[http://developer.android.com/sdk/1.1\\_r1/index.html](http://developer.android.com/sdk/1.1_r1/index.html)
9. Google Maps API  
<http://code.google.com/apis/maps/>



## Q1 - How old are you?



Subject1 Testing the application

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How old are you?	17.00	29.00	23.4	3.498	12.24	15

## Q2 - Are you an grduate or an undergraduate student?



Subject2 Testing the application

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Are you a grduate or an undergraduate student?	1.00	2.00	1.67	0.47	0.22	13



Subject3 Testing the application

### Q3 - Were you able to successfully find the destination address?

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Were you able to successfully find the destination address?	28.00	31.00	30.64	0.98	0.88	15

### Q4 - Were you able to identify the application's Destinations Tab feature?

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Were you able to identify the application's Destinations Tab feature?	17.00	23.00	21.82	0.61	0.41	12



# APPENDIX

<b>Abstract</b>	1
<b>Introduction</b>	1
<b>Establishing User Requirements</b>	1
<b>Initial Design</b>	1
<b>System</b>	2
<b>Implementation</b>	3
<b>Evaluation</b>	4
<b>Discussion</b>	5
<b>Conclusion And Future Work</b>	6