Commute Assist

Getting to Campus Quicker

Team Roles

Stefan Kapetanovic - Team Manager
Alec Adair - Lead Tester and head of QA
Warren Schweigert - Head of Communications and PR
Rusty Griggs - Lead Designer

Problem and Solution Overview

Many campus commuters do not know how to most efficiently commute to campus and are unaware of available parking. The information to make a commute more efficient is publicly available but difficult to coordinate especially while commuting. Our solution aims to make this information easily available and quickly accessible so that a commuter can make on the fly decisions on how to best get to campus and park when driving. To combat parking congestion we have designed a parking reservation system that will guides a user to a parking spot. In addition to this we have also designed a route scheduling guide to calculate the most efficient route to campus based on where a commuter is located and their means of transportation. The routing system is especially useful for commuting via public transportation. The application is also interactive through our voice user interface that provides hands free service when desired.

Initial Paper Prototype

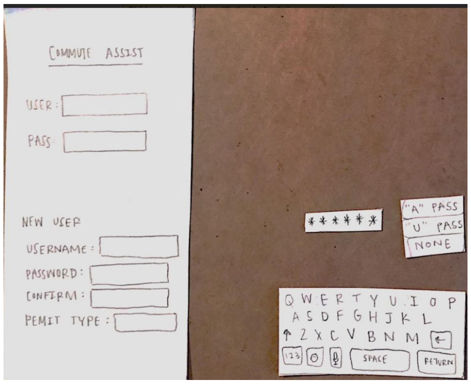
Our final design has changed significantly from our initial paper prototype. We found through testing and heuristic evaluation that we were originally presenting too much text based information to the user as well as focusing too much on helping a driver get to campus when they already usually know how. Although we had definitive flaws in our design some of the main aspects we knew we wanted to keep were being able to generate routes for people using public transportation as well as a parking lot information system that would quickly inform a user of where to park on campus.

Although we have changed our design our two primary tasks in using our application have not drastically changed since we designed our paper prototype.

TASKS

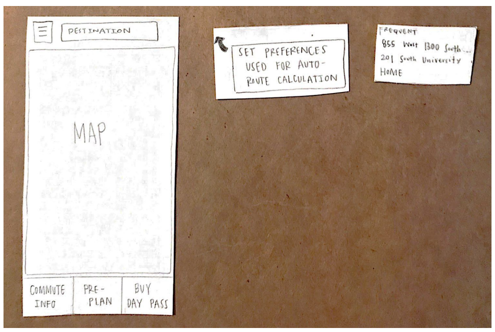
1. Determine different route methods from a specific origin and destination including walking, driving, public transportation (bus and trax), and university shuttle to arrive at a campus destination as quickly as possible.
2. Determine confidence margins and ease for the commuters parking on campus with a pay for a day parking permit option giving you an exact spot.

Below are the detailed images showing each component in our two primary task



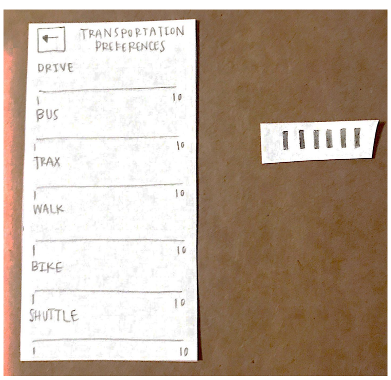
Login Screen: One time user login and new users accomodation. Sign on allows notifications, speaker interface, and GPS location.

This screen has retained its original design with an added button to login.



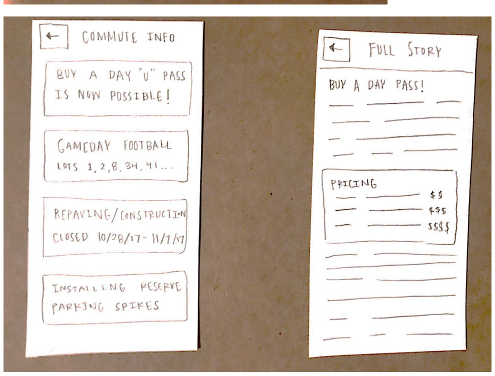
Main Screen: Three primary tabs. Pop up prompts user to set preferences first. Destination saves user's frequent locations.

The map will be a map with a default view of a university campus.



Transportation Preferences Screen: Ticks allow for 1-10 preference of each transport option.
This preferences screen would not be frequently changed by the user.As a user commutes the app will be able to change these preferences for them by tracking their commuting habits.

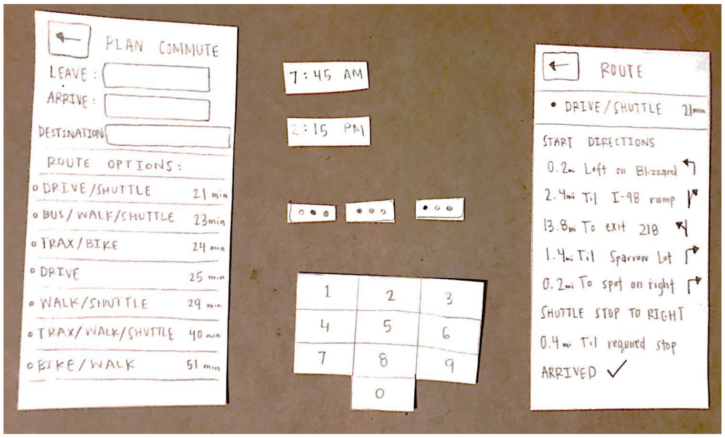
These values have switched to binary, and those values should remain mostly static.



Commute Info Tab Screen: Shows updating commute info. Adjusts to the user's personal commute statistics to target commuting information pertaining to each individual user. The second screen shows the full story of an event when a story is tapped on.

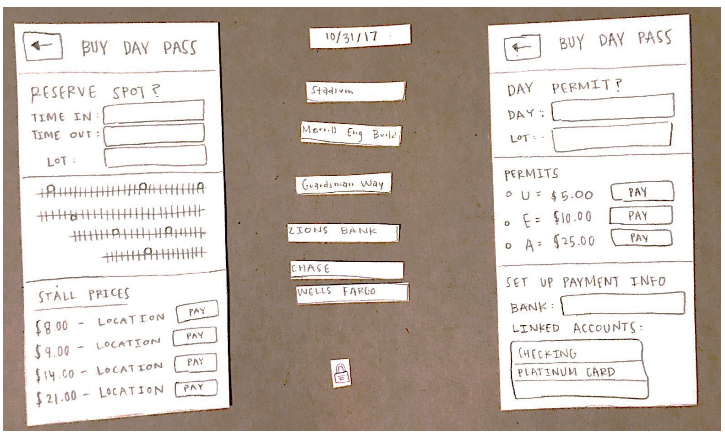
The commute info screen is not crucial for a user to accomplish task 1 but will be necessary to accomplish other tasks we have defined such as quickly becoming informed of the current campus commuting situation.

The commute info has been almost completely omitted from our final design.



Plan Tab Screen: Allows user to plan next day's commute ahead of time. Routes are always in order of fastest to slowest. Once route options is selected the second screen gives directions.

The directions info screen has now been constrained to a much smaller "space" within the application. We found the main use for a route info section would be for public transportation commuters only.



Buy Day Pass Tab Screen: Allows daily stall reservation and permit purchases in specific lots. Allows user to link a bank account.

To accomplish task 1 a user may need to buy a day pass if they find it necessary for time and convenience.

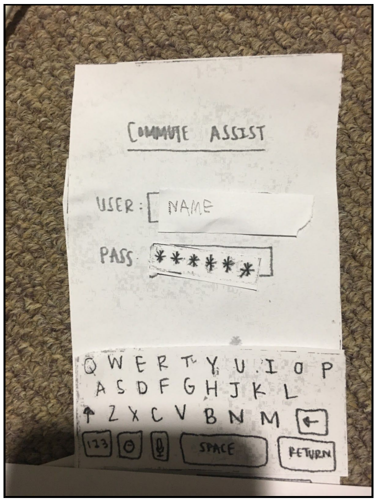
The time in and time out scroll boxes as well as lot will need to be filled out by the user. After these are filled out the user will need to hit a pay button for the respective type of parking they want. When the pay button is hit the Buy Day pass screen will pop up. The user will then input the lot they wish to park in, hit pay for a certain type of day parking pass, and then choose a linked bank account that the University of Utah also has linked.

In our final design we still support parking stall reservation and payment but we no longer have a user choose a stall or parking lot themselves. Both of these are now done by the application. We have also changed the interface significantly to separate and eliminate some of these components.

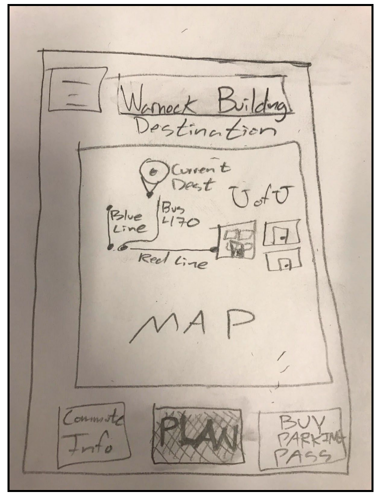
TASK 1

Determine different route methods from a specific origin and destination including walking, driving, public transportation (bus and trax), and university shuttle to arrive at a campus destination as quickly as possible.

Step 1:
Enter username and password to login. This will only have to manually input once on the first run of the application and will be automatically filled in after.



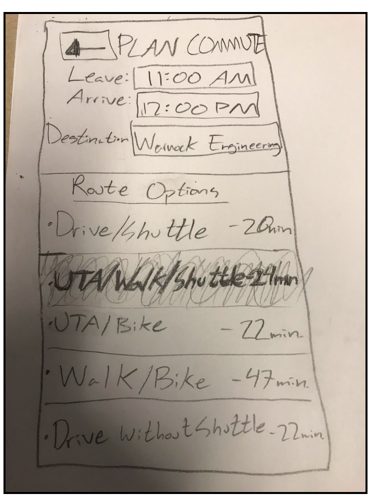
→



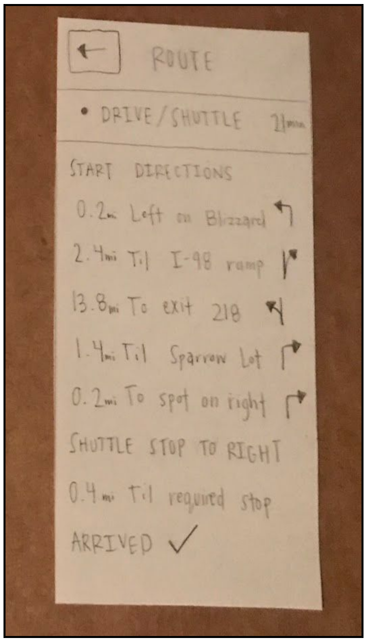
Step 2:
To accomplish task 1 a user would usually have their destination preset but may have to manually input their destination using the destination window for special cases. On the map current bus, shuttle, and trax train locations would be displayed simultaneously. A user can tap on an individual transportation route to single it out and hide other routes. When a route is tapped the plan button can be pressed to get information on how to take the specific route.

Alternatively a user is not required to tap a route and can directly push plan to get a plan of how to get to campus quickest. Either by selecting a specific route and hitting plan or by tapping plan by itself. The plan commute screen will be followed.

A user will have their destination prefilled in.



STEP 3
The user must now select a preferred mode of transportation and ensure that the departure and arrival times are correct. Both times should usually be automatically filled out but may need to be changed for certain special cases by the user. In this case the user is selecting to use
UTA/walk/shuttle. The approximate commuting time for this route is
displayed as 24 minutes.



STEP 4

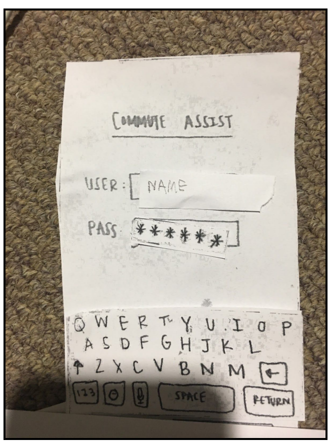
The only step left for the user to determine their best route is to read the steps given on the route screen. This screen will show the user step by step instructions of how to commute to campus.

As the user is commuting the application will be able to know where in the commute the user is at using the phone's GPS and check off each step in the commute. After each step the user will be notified of the next step in the commute either by their phone's default notifications or if the user is driving an auditory voice will be used to notify the user of the next step in their commute.

These notifications can be turned off in the phone's application settings.

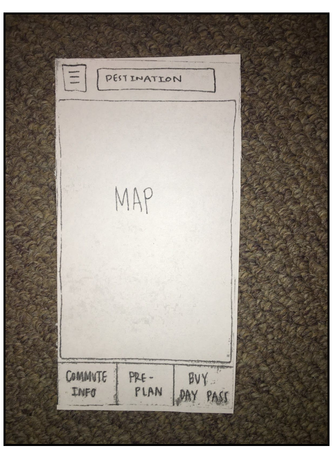
TASK 2:

Determine confidence margins and ease for the commuters parking on campus with a pay for a day parking permit option giving you an exact spot.



Step 1:

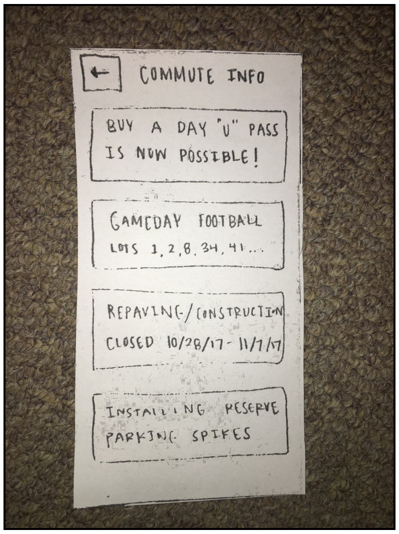
Enter Username and password to log on, or if voice recognition has been set up you can use that instead.



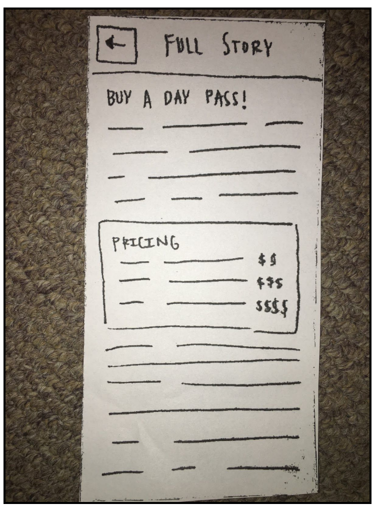
Step 2:

a.)The upcoming route for your schedule will be shown on default. However if they want a different route they can do this in multiple ways. First they can select a route as in task 1, or they can enter a destination to check what would happen if they just wanted to go there now.

b.) The user then either clicks on or speaks the words "Commute Info" to go to the info.



Step 3:
The user now has access to basic commute info, they can scroll up and down to see more items, if they need more info on a topic they can either click on it or ask the VUI "Give me more info on ~A" where ~A is the topic they want more info on.



Step 4:
Here the user can easily see the info they were looking for. They can scroll up and down the page to get all the details on anything they want. If the user wants to head to a previous screen they can do so by hitting the back button, or tell the VUI "Go Back".

Testing Process

Usability Tests

Our usability testing process was mostly sitting the participant down at a table with someone in our group presenting them with the appropriate application screens while another person in our group took notes and observed. We gave the participants scenarios in which they would use our application with their primary task(s) and tried to only give them background information about the application and paper prototype, but not actually any information on how to use it. In Most of the tests we administered it was assumed the user had just downloaded and installed the application and is using it for the first time, but also conducted tests assuming they had logged in before.

Participant 1: University of Utah Engineering Student
Commutes daily using only public transportation
Test Date: Nov 6, 2017

Roles: Warren Schweigert - Computer & Test Moderator | Alec Adair - observer, recorder
Scenario: Participant 1 is at home getting ready to leave school in 15 minutes and would like to know the quickest way to get to class using only public transportation.
Tasks: 1. Find the quickest route to class using public transportation and campus shuttle
2. Board all transportation necessary and follow through with the route identified

Participant 2: University of Utah Computer Science Student
Owns a vehicle, parking a pass, and drives or carpools daily
Test Date: Nov 8, 2017

Roles: Stefan Kapetanovic - Computer & Moderator | Warren Schweigert - observer, recorder
Scenario: Participant 2 is at home and their car is broken down. They are at home at 900 East and 900 South in SLC getting ready for school and would like to find the quickest mode of public transportation to get to the WEB.
Tasks: 1. Find the quickest route possible using his vehicle and campus shuttle to get to class
2. Board all transportation necessary and follow through with the route identified

Participant 3: University of Utah Student
Owns a vehicle, parking pass, and drives daily
Test Date: Nov 9, 2017
Roles: Warren Schweigert - Computer & Test moderator | Rusty Griggs - observer, recorder
Scenario: Headed to campus during peak parking rush hours and wants to find a parking spot
Tasks: 1. Navigate to campus
2. Reserve a parking stall

Usability Testing & Design Refinement

Although our test format did not change considerably throughout our three usability tests, our paper prototype we were conducting the tests with did change considerably. With each refinement of the paper prototype in each test we were able to more effectively test the core feature and task support of our application instead of trivial or extraneous features. This lead to better usability tests each time where we really learned what inherent problems we still had to overcome in our designs.

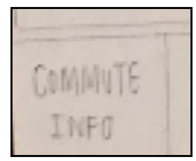
Testing Results

Issues Identified and Prototype Revisions

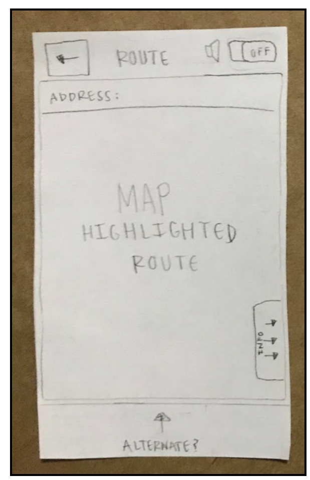
Usability Test 1 Results

The following issues were identified during the first test and corresponding changes were made:

- The Commute Info button was mislabeled and misleading and was changed to Commute Data. After further testing we have almost completely done away with this feature as we found it was impractical and not useful. Issue Severity - 3



- Our main map screen was not clear how to use. S1 had difficulty getting started with the application, and it showed us that we need a more intuitive introduction screen. Issue Severity - 3
- Commute info button was not used nor did S1 express any interest in commuting news as long as the application could switch his route in real time as needed. Issue Severity - 3
- We did not have an arrived at destination screen or notification of arrival and our participant was unclear what do with the application when he arrived. Issue Severity - 1
- Positives included a strong potential, proactive use of the Voice User Interface, and a progression towards simplicity that we can produce throughout the app

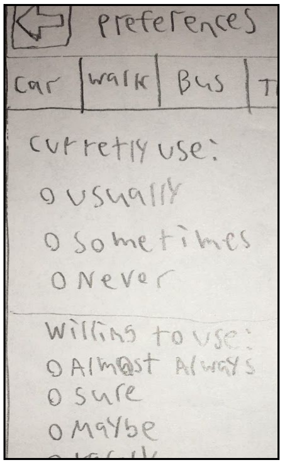


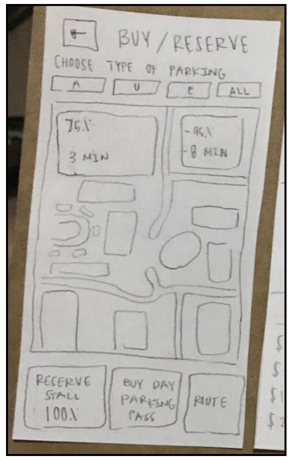
Usability Test 2 Results

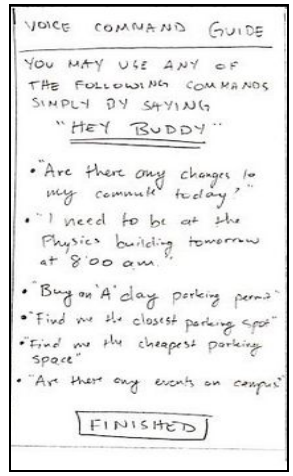
The Voice User Interface was a confusing topic that S2 did not know how to interact with. S2 didn't know what commands he could and couldn't use. This issue was not the only one. S2 also had problems understanding the preferences page as well as the reservations system. We identified the following problems in our design after the second test:

- In the preferences it was not made clear what the words "use now". Issue Severity - 1
- The button to the preferences pane should not be a box with three lines in it, it has been changed to a box with a gear icon. Issue Severity - 2

- The meanings of get route and reservations was not made clear enough, and implied that get route was dependent on reserving. Issue Severity - 3
- No way to cancel reservations, or make multiple reservations. Issue Severity - 3
- Duration of reservations not clear. Issue Severity - 2







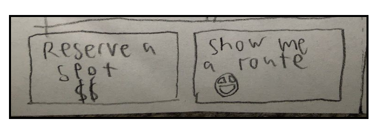
- While the VUI tutorial gives example commands it is hard to know what they were if after the tutorial you wanted a list of VUI commands. Issue Severity - 3

Usability Test 3 Results

S3 was asked to get to campus using this application. This resulted in S3 making a reserving a spot with this application. This implies that our application left the impression of needing to reserve before finding a route. Essentially the user ended up completing the reservation task while working on finding his route to campus. This is a serious issue as our application was not meant to require the user to reserve a spot (or imply such a requirement) .

We identified the following problems in our design after the third usability test:

- “Use now” was changed to “currently used” but is still confusing. Issue Severity - 2
- “Willing to use” was also confusing. Issue Severity - 2
- One of the selections was on the preferences was “absolutely always” when it should have been “absolutely not”. Issue Severity - 3
- Travel time on maps and percentages are confusing. Issue Severity - 1
- Stall reservation is probably the biggest problem. It is not clear how or when to reserve a stall. We needed to answer the following questions: Do you do it before you leave? Why would you ever reserve a stall? Can I still use the app but not reserve a stall and find a spot? Issue Severity - 1



Crucial Revisions

In our heuristic evaluation and design usability test review we really learned that we needed to slim down our application. We had too many screens with trivial or unneeded features as well as relied on the user for too much use. Nearly all the feedback we got on our designs was that they were overly complex, confusing (felt nonlinear), and relied on the user too much for trivial tasks.

After the usability testing, heuristic evaluations, and peer critiques, we made the following changes to our design:

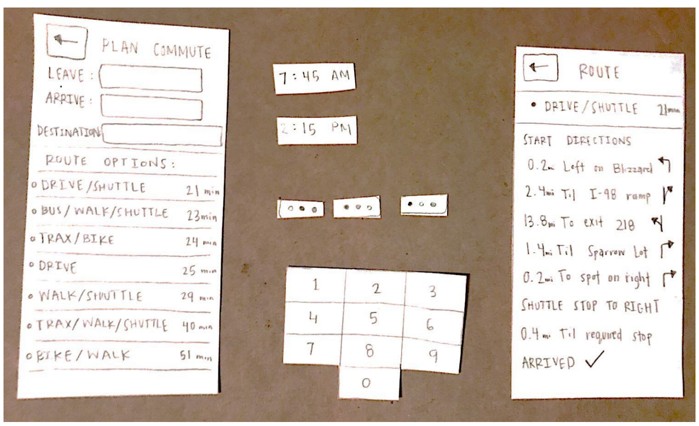
1.) We got rid of the commuting news page, it was relatively useless and just confused the users and updated our home screen to cater to the most important feature of the app.
2.) The preferences have been updated to help the user understand what they should enter and how the preferences are used. It is more intuitive for the user to get to this screen afterwards and correctly use it.
3.) Parking stall reservations have been streamlined and integrated into the route finding, with the VUI or GUI asking if you want to reserve a spot (this can also be disabled for those who never/rarely want to reserve a spot or set to remember your reservations for those who want to reserve the same spot each week).
4.) We removed numeric representations of how full lots were and replaced it with color coding to represent how full parking lots are so that the user can easily get an overview of the situation.
5.) Notifications and automatic Voice User Interaction (VUI) are features enabled initially and can be disabled through voice command or through the interface at any time. We found that the user would use the VUI more if the VUI started the interaction and asked if it would be disabled. The VUI can handle all interactions that are done over the phone without the user ever having to touch the screen.

Final Paper Prototype

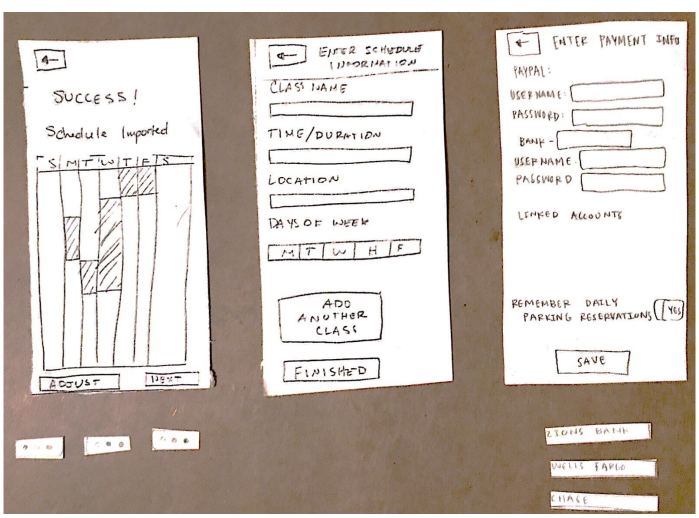
Our final paper prototype is considerably different than our initial prototype. We have made a great deal of simplification which means many parts of our app have been taken out completely. We realized after testing and getting feedback that our application has useful aspects but were being swallowed up by other features that were not important. We wanted our application to do *everything* involving commuting and realized that it was much too complicated. The biggest changes were to remove the commute data screen, improve the parking reservation system, improve the preferences section, and remove a lot of superfluous screens that don't add substance to the core design of the app.

Primary tasks for final paper prototype

1. Find most efficient route to campus based on origin location, destination on campus, and preferred mode of transportation.
2. Find campus parking, predict lot availability, and/or pay for parking and reserve a spot if needed.

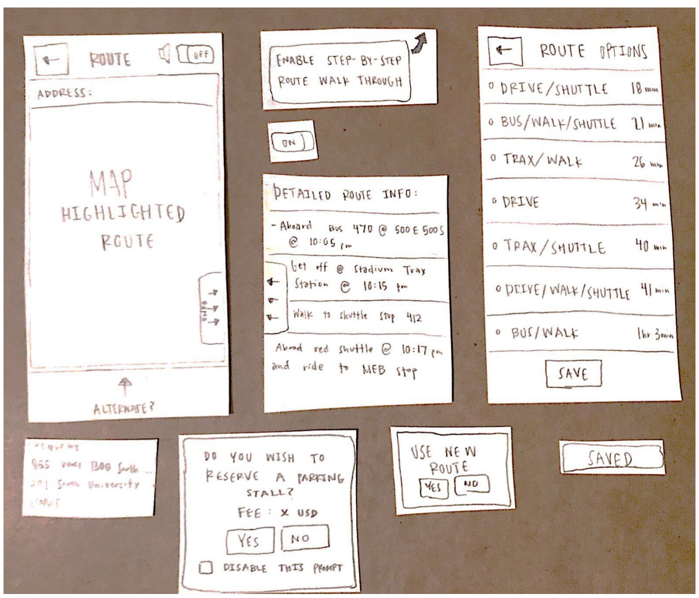


The user must still login on the first run of the application and go through the preferences page, but the page has been greatly simplified and only has binary switches for transportation preferences.



When the import schedule button is hit in the preferences, a confirmation screen will show the user their schedule was imported, or if they choose to enter it manually they may do so.

Our main Commuter Assist screen is where a user gets directions both to campus and a parking spot when a reservation is placed. The map shows the campus with parking lots that are color coded.The colors correspond to the availability of parking in the lot. A destination address and arrival time may be entered but they will be autofilled from the imported schedule. The user will usually just use the filled in values and simply tap the “Get Directions” button on the bottom of the screen.



Once the user has chosen to “get directions,” the most efficient route according to their preferences will be chosen and the app will start guiding them. It is at this point that the user may choose to change their route by sliding up the “alternate” tab on the bottom of the screen. The route options screen then appears giving all the different alternative transportation options and their corresponding times.

The decision to start guiding the user immediately without giving route options first was deliberate. Since the user will generally use the first and fastest option, it didn’t make sense to place a burden on the user to have to choose that first option every time. However, if the user desires to take a different option (i.e. take trax to have homework time), that is something the user can do. Choosing a different option will likely be a rare occurrence and therefore should not be a mandatory part of the app.

The user may also see the route details by sliding the right-hand tab over to the left. This final paper prototype failed to provide enough detail and attention to the parking stall reservation system. This lacking element was improved for the digital mock-up.

Digital Mockup

Tasks and Application Walk through

Primary Tasks

1. Find most efficient route to campus based on origin location, destination on campus, and preferred mode of transportation.
2. Find campus parking, predict lot availability, and/or pay for parking and reserve a spot if needed.

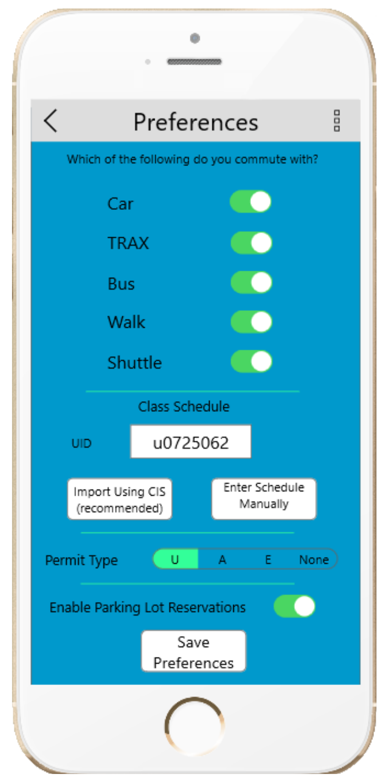
Subtask

A. Create a login and set preferences

Subtask A - Create Login and Set Preferences

The very first step in using our application is to login. If a user has not created an account they will need to create one.

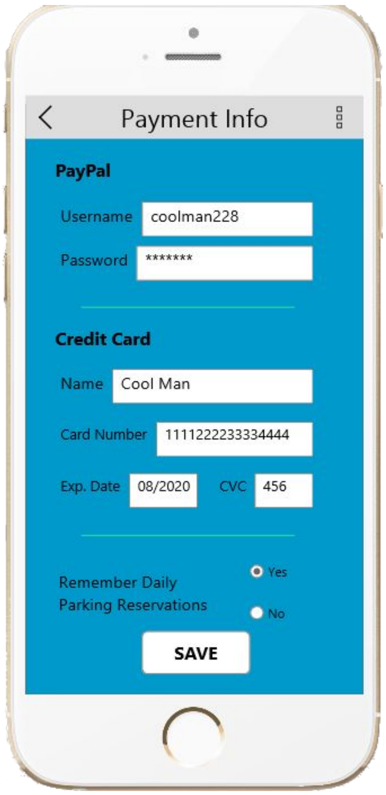
To accomplish both of the primary tasks a user will need to have an account and be logged into the application.



On the first run of the application after a user is logs in they will be directed to the preferences screen. Here they will input the modes of transportation that they would like the application to consider when calculating routes. They will also import their school schedules so that the application knows their schedule, select what type of parking permit they have, and enable parking lot reservations for a cost.

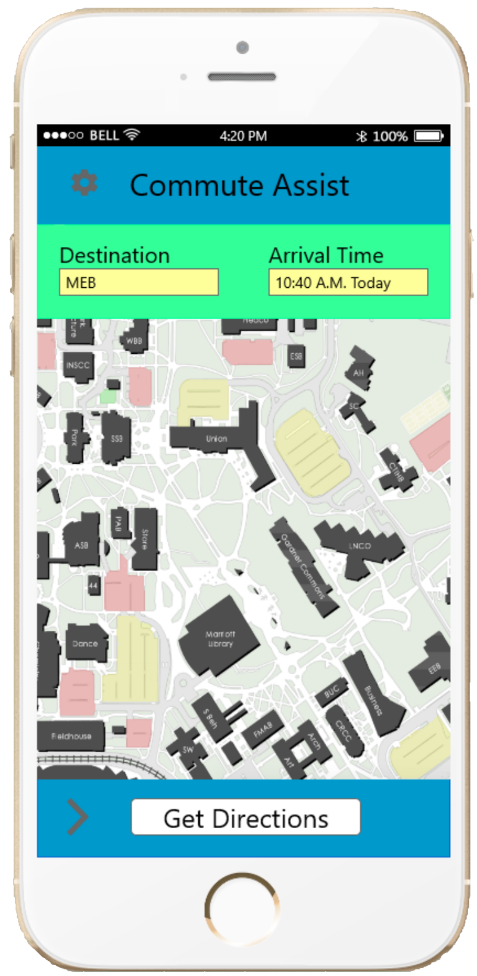
When a user hits the import schedule automatically from CIS button they will be directed to this screen. In this screen the user can view their school schedule to make sure it is right and have a visual confirmation of the system status. If for some reason it is wrong they may adjust it.

If a user does not enter their schedule, they will have to manually input their destinations and times into the route scheduling.



If a user chose to enable parking reservations they will be directed to this Payment info screen. Also if they press the edit payment info button in the preferences screen they will be directed here. In this screen the user can input their credit card information and/or Paypal information.

This is not needed in accomplishing task 1, but maybe crucial depending on the user's parking constraints for task 2.

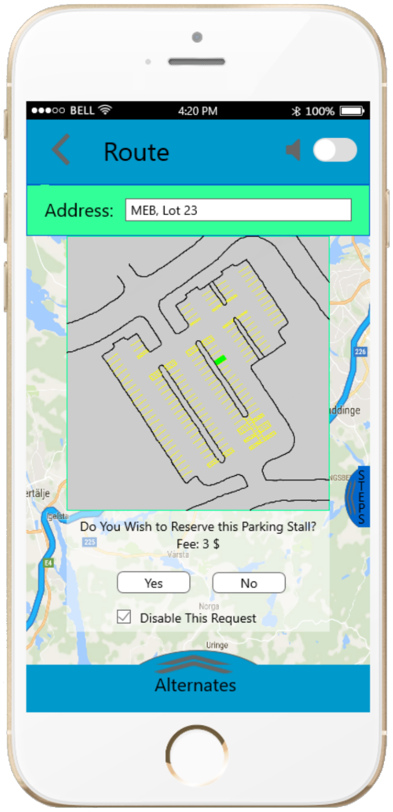


After a user is logged in they will be directed to this main Commuter Assist screen. This is the main opening screen once the user has logged in for the first time.

The user will have their destination automatically input by the app based on their schedule. If they wish to they can enter a different address or set a pin on the map. The arrival time may also be changed, but will be preset based on a user's schedule.

If the user drives, then the parking lots on campus will be color coded. This allows for the user to know the parking situation at a glance.
If a user does not want to reserve a parking spot and only wants to try parking in a lot based on predicted probability task 2 is mostly accomplished at this point.

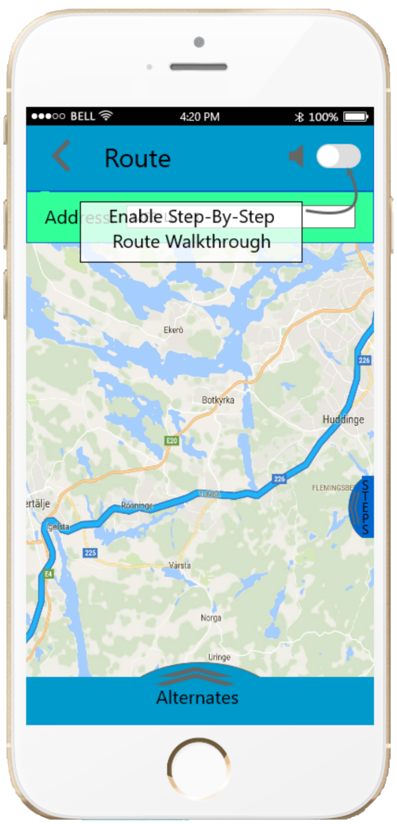
After the user is satisfied with their arrival time and destination they can push the get directions button. This will lead them to their destination. Which if they like the route it provides based on their preferences completes task one.

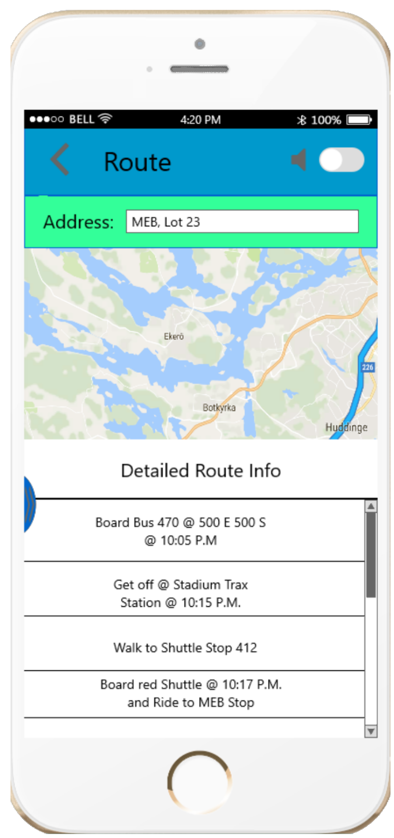


Once they get directions they will be asked if they want to reserve a stall. The user may accept or decline and they may also disable the request from ever appearing again. At this point the user is informed of the parking situation on campus and they have accomplished task 2 if they are a driver/parker.

The map will then zoom out giving the user most efficient route to campus based on origin location, destination on campus, and preferred mode of transportation. For many users this may be the end of task one, however for those wanting to take a different route or needing directions it continues.

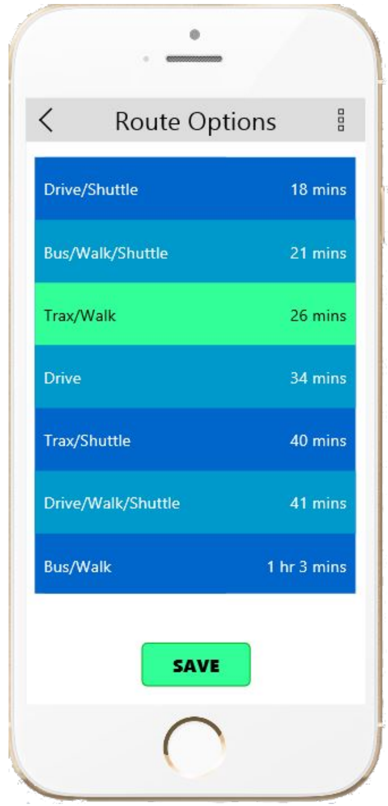
There is a button for an auditory step by step route guide for drivers and visually impaired people riding public transportation. It is also possible for the user to view a list of the steps that their route takes by hitting on the steps pull out on the right hand side of the screen. This will provide in depth details of the route

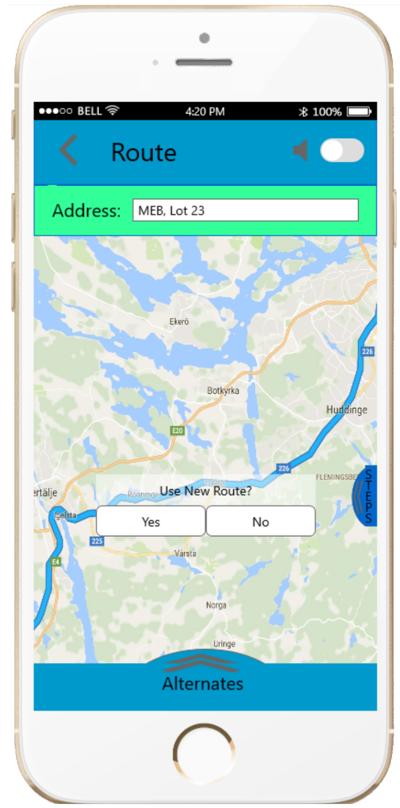




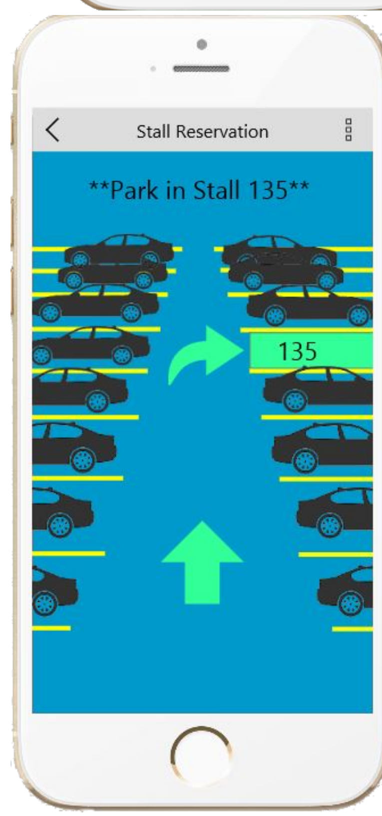
Should the user want to look at alternate routes that would take them to their destination they need only pull up on the alternates tab in order to get to the alternates screen.

After the route is selected the user will be returned to the original map screen. Thus completing task one, finding most efficient route to campus based on origin location, destination on campus, and preferred mode of transportation





If the user reserved a spot they will be guided to its exact location in the parking lot.



Decisions and Changes in Implementation and Design

By the time we had all changes and design decisions made in our paper prototype we had pretty well established what we would be including and designing for in the digital mockup. However the black and white which worked well in the paper prototypes looked to minimal on a digital device. This prompted us to add color to the design, eventually we settled on using greens and blues.

During the implementation of the digital mock-up we did realize we had some errors in our design. We have also learned that the users needed a visual indication for when data is automatically filled based on their schedule and preferences. Finally we added directions to reserved parking stalls within parking lots, as trying to find a parking stall while knowing only its number would be difficult.

Discussion

- **What did you learn from the process of iterative design?**

The process of iterative design was extremely useful in getting the design right. It is very difficult to start out with a perfect design. We did our best to start with a design that would require minimal changes but soon realized that we would need somewhat of a major overhaul. We realized pretty quickly that our design was very busy and lacked a simplistic design pattern that many successful designs have. Each iteration removed a confusing aspect of our design and made our design smoother, less convoluted, and more user-friendly.

- **How did the process shape your final design?**

One of the clearest example of how the iterative process shaped our final design is the change in our preferences page. We wanted to show that the user could choose which mode of transportation was most important. The app would then show them routes based on those preferences. We started this by showing the different modes of transportation with a scales of 1-10 under each one. We expected this to be clear and self-explanatory. The higher the number, the more the user wants to use that transportation and be shown the route options for that mode of transportation. After feedback, we added a clearer explanation what the scale was meant to convey. We asked "How likely are you to use the following modes of transportation? 1 - not at all, 10 - extremely likely." We even added a little info box that would explain for what the preferences you chose would be used. This also proved to be confusing and after further feedback and another iteration, we chose to overhaul the scale/rating system. We decided to highlight each mode of transportation and state "use now" above a list of options showing "yes, sometimes, or no" Then we asked if they were willing to use with a scale of "almost always, sure, maybe, not really, and absolutely not." After more user testing, this still proved to be confusing and unclear. We changed "use now" to "currently use" with no improvement. Eventually we simplified it to a yes or no option for each mode of transportation which has proven to be clearer and more user-friendly.
As shown throughout the report, this is not the only place where this process shaped our final design. Our parking stall reservation system was revamped and overhauled piece by piece thanks to the feedback we received little by little. Some portions of our initial

design were completely removed such as the commute data/info section and the option to buy a parking day permit.

- **How have your tasks changed as a result of your usability tests?**

Our tasks have indeed changed as a result of the iterative process. In our initial design, we chose two tasks from our previous six main tasks that we felt encompassed the main idea of our app. Essentially, the tasks were to determine different route methods and arrive as quickly as possible, and determine confidence margins to find the best parking spot. For our first usability test, we realized that we would need to simplify these tasks. We asked the first and second user to use the application to find a route to campus and find information about the route. We realized that finding information about the route wasn't as important as we originally believed and simply asked the third user to use the app to get to campus and reserve a parking stall. Ultimately, we decided our two main tasks would be 1. *finding the most efficient route to campus* and 2. *finding parking using lot availability and stall reservation*.

- **Do you think you could have used more, or fewer, iterations upon your design?**

We believe that more iterations would have been even more helpful to work out the kinks and problems in our design. Every iteration of testing and redesign fixed a problem but many times, our solution turned out to be another problem. "Use now" was changed to "currently use" but was still found to be confusing. Instead of changing the words, we changed the interface. Since we have seen the profound effects of the iterations on our design, we believe that with more testing, our product could be refined and improved even more. Since we haven't tested our design with a large sample of diverse users, there are likely still problems with our design. Obviously there comes a point when improvement is minimal and is not worth the effort that it requires. It becomes a cost-prohibitive process. However, we are in agreement that to arrive at that point would take much more than the few weeks we have spent refining our design.

Appendix

No additional materials were used in the usability tests that has not been discussed and shown herein.