Using a Heuristic Evaluation on the mock-ups, we evaluate each tab and how they transition to one another, both evaluating their static designs and how a user would interaction between the tabs.

**First Pass:**

**Map tab:**

The map tab holds very much the same feel as the Transloc map. Using the map service, one is able to zoom in and out of the map. Colored path and dots highlight the shuttle route and stops, respectively, and a pin with an arrow represents the app’s most recent location for the shuttle.

The bottom bar offers something different: two sets of buttons encompass the bar. The left hand side has a start and next button. Pushing start changes the map location and zoom to the nearest stop to your location; the next button does the same for the next stop after. To the right side, one is able to select which route to have shown. However, the design of the buttons means only red and green routes are visible to the eye. We are unable to select the blue route or all three routes.

**Timetable tab:**

Our selected route is reflected on the bottom bar yet again here, and this reflects the information presented to us in the main window. A listing of stops and expect shuttle times populate this list. One can toggle whether to have AM or PM times come about too, but then again the small box to the side allows for changing of hours, too. One might have to squint given the color scheme here; the font color fades into the background color.

**Feedback tab:**

The simplistic tab offers a simple view of the selected route (from previous tabs) and the report of delays for the shuttle. With a tap of a button, users can submit their own report: by selecting their line and then delayed time from a dropdown, they can submit feedback for other app users to see. The tab itself looks kind of bland and disproportional; and given the potential for what the tab can do, it falls short functionality wise.

**Our Second Pass, using Kessel’s list of Heuristics:**

**• Simple and natural dialogue**

Very little text save for the time-table and selections means there is little to comprehend; the abbreviations for minutes in the feedback tab might be the only thing that could take a second to come to a conclusion about what it means.

**• Speak the users’ language**

As stated above, very little text exists, and what text there is there is straightforward and not complicated.

**• Minimize the users’ memory load PAR**

Most of the app’s usage will be found with the map tab, which will update live both the user’s location and the shuttle’s location. Larger icons for stops and the user means a clearer understanding of where a user might have to go.

The timetable requires a few pushes to adjust the time and route desired, but will stay as is if the user goes to another tab.

**• Consistency**

Both the map tab and time-table tab offer similar design principles, with a bottom bar for route selection; however the feedback bar fails to offer something similar, and has its own means of selecting a route for feedback.

**• Feedback**

The only feedback offered is that of the contents of the app (shuttle lateness), and not for the app itself.

**• Clearly marked exits**

The app is strictly an updating app that defaults to the map tab; no specific log-ins or prior setup is applied that require the ‘exiting’ of the app.

**• Shortcuts**

The top bar shows the three main tabs, which serves as the primary form of navigation. All three tabs offer it, so going from one tab to the other is a one step process (no shortcut for it is necessary).

**• Precise and constructive error messages**

No error messages occur in the app: given the lack of input forms used (save for the feedback), there is little that could cause an error to arise by the user’s actions. The dependency for the shuttle tracking part on the map would cause issue, because if something were to go wrong, it would only reflect in the map (ex: Shuttle stalled, shuttle out of service).

**• Prevent errors**

As mentioned above, the lack of inputs means there is no typical way for a user to error in operating the app, as most of it deals with looking at incoming content.

**• Help and documentation**

No help bubbles or documentation of how to operate the app is present.