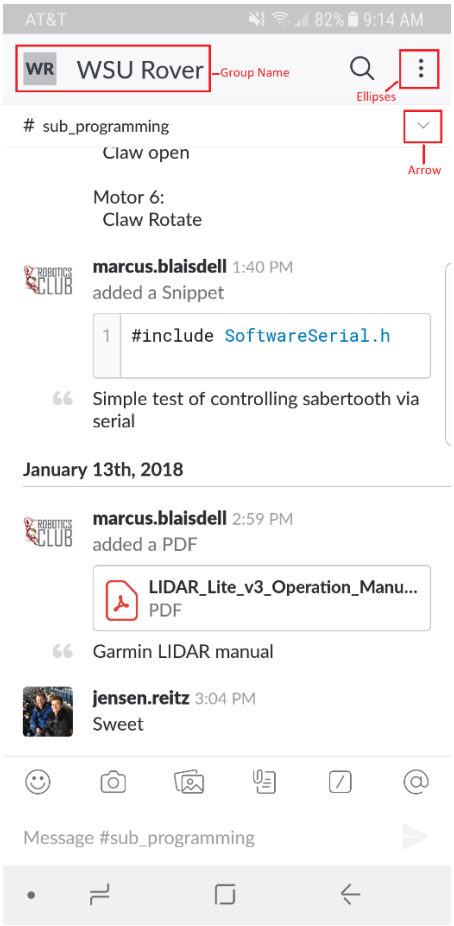
Bad Design: Slack mobile app



Slack is a group-collaboration app that allows a team to share and organize their group communications. It allows users to create sub-channels to segregate conversations. This keeps the data organized and easier to work with. It has a web-based interface that can be accessed from any web browser on any OS and a mobile app. This analysis is on the Android mobile app.

Conceptual Model: The messages appear as a continuous scrolling with the most recent at the bottom. The group that is being viewed is clearly displayed in a larger font at the top. The sub-channel that is currently being displayed is clearly visible above the messages in that sub-channel. There are signifiers to note options that are available to the user. They can enter new text in the greyed-out box at the bottom. They can scroll back through earlier messages as indicated by a scroll bar on the right hand side that was not recorded in this screen capture but is apparent on the actual app. There is a down arrow, vertical ellipses, and icons that indicate that there are options and new actions that the user can choose. This is where the failure occurred. I wanted to change channels from the #sub_programming channel to get to the #general channel to post a notice.

So, expecting the sub-channel options, I clicked on the down arrow next to the sub-channel name which instead brought up channel details. Clicking on the ellipses which I expected to bring up

channel details brought up group options. Clicking on the group name which I would have expected to bring up group options brought up sub-channel options.

It had seemed logical to me to push the down arrow next to the #sub_programming channel name assuming this would be a pull-down menu of available sub-channels. Instead, this brought up the screen that showed information about the channel and options related to that channel. I would have expected to see this information by clicking on the ellipses. Clicking on the ellipses then brings up options for the group. I would have expected that clicking on the group name would have brought up this information. Clicking on the group name brings up the sub-channel options that I had expected from the down arrow next to the sub-channel name.

Affordances: This is on a smart phone that uses a touch-screen interface so the affordances are: simulated buttons (tapping) and dragging. Due to the ubiquity of smartphones, the use of a touch-screen rarely requires explanation or training.

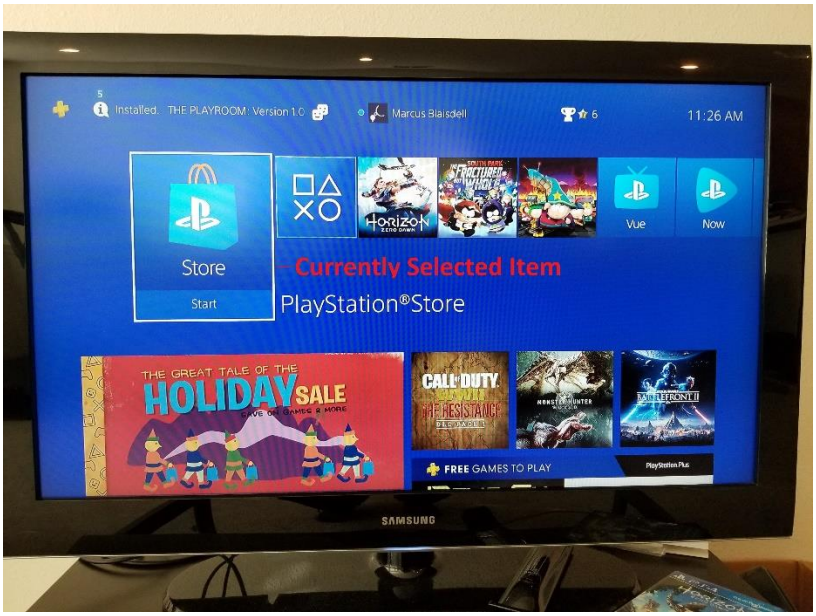
Signifiers: The interface has icons, ellipses, greyed-out text, and boxes that are common to mobile apps and computer software. The icons are selectable so tapping on them activates their function. The scroll indicator shows the relative position in the chain of messages.

Constraints:
Physical: The screen has a "touchable area", that is, there are parts of the screen that will respond to touch and parts that will not. An internet connection is required for the app to function as the user is viewing data from a server on their device but the data is not on the device itself.
Cultural: Smart phones are nearly ubiquitous but the app does require familiarity with the touch-screen interface. No training is provided and knowledge of certain aspects of smart phone and computer software are assumed.
Semantic: Tapping on the smiley face brings up emojis. The paper clip image denotes adding an attachment. The magnifying glass denotes search. The "@" symbol sends a message to the specified person, group, or sub-channel. There is significant use of symbols to facilitate the use of the app.

Mappings: Some mappings are logical and function as expected such as the magnifying glass performs a search and clicking in the area with greyed-out text allows the user to create a new message. However, three mappings were, in my opinion, illogical. Based on my experience with other pieces of software, I had expectations of what action tapping on the icons would do and they actually did different things. This was not a simple swap of two functions but a circular shift of three functions that took some experimentation to learn their true nature.

Feedback: Pressing buttons provides instant visual feedback. Clicking in the greyed-out area moves the cursor to the text box and opens the on-screen keyboard. Clicking on the magnifying glass opens a helpful window of search filters, dragging a finger up or down scrolls through messages.

Good Design: Sony Playstation 4 interface



Conceptual Model: The interface is menu-driven. It uses a white border around the currently selected box/button that lets the user know where they are. Left and right arrows on the controller scroll through the buttons. Pressing "X" selects a highlighted button and moves to a new screen. In screens such as the Store, there are signifiers on the bottom of the screen to let the user know what options are available. There is also a direct link to the Privacy and Terms of use pages making them highly visible for users that wish to peruse them. The interface behaves the way I expect it to behave making it easy to use.

Affordances: The interface is through the Playstation controller. There are physical buttons that can be pressed.

Signifiers: There is a white border around the currently selected box that is absent on the unselected boxes. There are labels on the buttons on the controller and the buttons have different shapes. Some of the shapes correspond to function such as the direction buttons.

Constraints: The interface selection will only highlight selectable items. If an element on the screen is not selectable, the indicator will not highlight it. The controller must be paired to the console.

Mapping: The mapping is logical. The arrow buttons move the cursor in the selected direction.

Feedback: There is visual feedback from the television. As a button is pressed, a visual change occurs on the TV. There is also audio feedback from TV and the controller. There is vibrational feedback from controller.

Augmented Design: Robotics Lab Card reader

This is the card reader for the Robotics Laboratory. It is used to track people that use the lab and record meeting attendance. The reader has been augmented to include a label to indicate which direction the card should be facing when swiping. This was added because many people would swipe their card facing the incorrect way and would not know if the reader didn't read it because they swiped too fast or too slow and so would swipe again with their card still facing the same direction before turning it around.

Conceptual model: A card reader is a simple model where a card with a magnetic stripe is swiped through a reader to record the data. The currently running program on the computer then takes that data and processes it.

Affordances: There is a magnetically sensitive head on the reader and a magnetized strip on the card that need to move past each other to transfer data. The reader is on a flexible cable that can be moved about to allow the user to access it from any side of the table and be used comfortably if seated or standing.

Signifiers: The reader has a groove that indicates the orientation for the card. The paper label was added to let the user know which direction the card should face when being inserted.

Constraints:
Physical: The card must be oriented vertically along the long side when being swiped. The reader is shorter than a Cougar Card and so the side of the card with the strip must be passed through the reader leaving ample room for the user to maintain a good grip on the card.

Mappings: The reader has a single functional interface with eight possible orientations, the label was added as a signifier to let the user know which direction the card should be facing. The short stature of the reader indicates that the magnetic strip must be on the side that moves through the reader

Feedback: The reader is attached to a display that shows a message after swiping to indicate success or failure. The user can see at a glance if their swipe has been recorded or if they need to re-swipe.