Paper Prototyping, Testing, and Refinement

Amy Brumet, Meleigha Holt - Design and Test
Matthew Bryan, Kyle Hipke - Writing, Team Management

It is common knowledge that responsible, working parents have busy lives. To stay on top of things, these parents need to stay organized with respect to time and space. They must organize their time since they have to negotiate every family member's schedules and allocate time for each task they need to do. They must also organize their space since they have to keep the house clean and make it easy to find belongings when they are in a rush. These organizational demands prevent parents from doing more desirable activities.

Our domestic robot system aims to decrease the time parents spend organizing the house. Parents could automate this robot to periodically pick up and store objects in their designated storage locations. The system allows the users to schedule cleaning, which enables them to remove this task from their daily planning. The parents can also order the robot to clean on demand through a web or phone interface. The parents define where items in each room go. They do this using the interface before the robot actually performs the task, and the robot's job is to reorganize the items in the room to get them back to that state.

1 Paper Prototype Description

Our interface has two main modes - the viewfinder and the map. The viewfinder shows the current camera view of the robot. This allows the user to monitor the robot as it goes about its tasks. It also allows the users to monitor whatever is going on in the environment. The map provides a top-down view of the household, allowing the user to specify tasks related to areas away from the robot's current location. When in viewfinder mode, a slide-out menu on the right provides an opportunity to issue tasks related to the room the robot currently finds itself in, or to schedule a task. When in map mode, the slide-out menu allows the user to perform more high-level tasks such as seeing the cleaning schedule or scheduling a task. The cleaning schedule depicts which rooms are to be cleaned and at what times. The user schedules tasks on a screen where they select the day and time, then choose the recurrence pattern for the cleaning (e.g. daily vs. weekly, etc.).

The interface presents a "learn" command which requests the robot learn what the clean environment is supposed to look like. When the user selects the "learn" command in the slide-out menu in the viewfinder, we present a confirmation screen. Learning can take the robot some time, and if the user wants to make a last-minute change to the room before learning occurs, they may want to cancel. The confirmation screen allows them to back out if necessary. We also display another "cancel" button while learning is in progress. This provides the user a recourse "when something goes wrong" (e.g. they change their mind).

Figure 1: Menu flow for each task. Zoom in for more detail.

We have designed the interface to allow at least three tasks, which are listed below in our testing methods section. In order to demonstrate how a person uses this interface more clearly, we will walk through a task with screen shots. You can find additional details about the paper prototype in Appendix C, if needed. Let us schedule a cleaning for the dining room, assuming the robot is currently in the living room. Assume we begin on the the viewfinder as depicted in Fig.2.

To schedule a cleaning we could press the "map" button in the lower-left-hand corner, bringing us to the map view depicted in Fig.3.

At this point we can press on the dining room, which brings up a context menu as depicted in Fig.4.

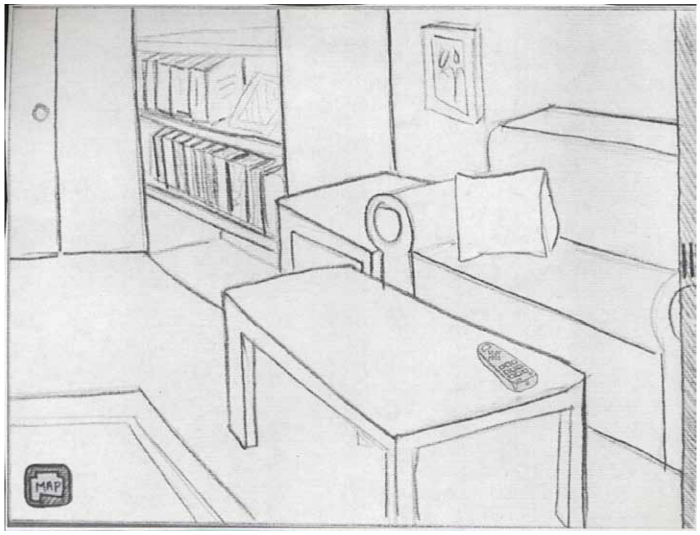


Figure 2: Viewfinder mode - looking through the robot's eyes

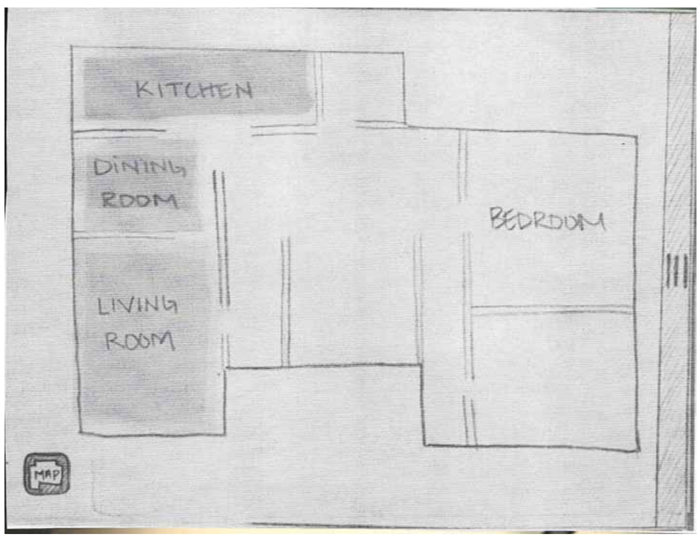


Figure 3: Map mode - view of the whole house

After clicking on "schedule now", the scheduling screen comes up and the user can enter the parameters for the task. See Fig.5.

2 Testing Method

In our experiments we had four participants aged 29, 43, 47, and 50. All were parents or had children on the way. We selected these participants since they represent our primary target audience - parents. Since the subjects have a good range of ages, we were able to sample with some amount of demographic diversity. These participants also had a considerable range of

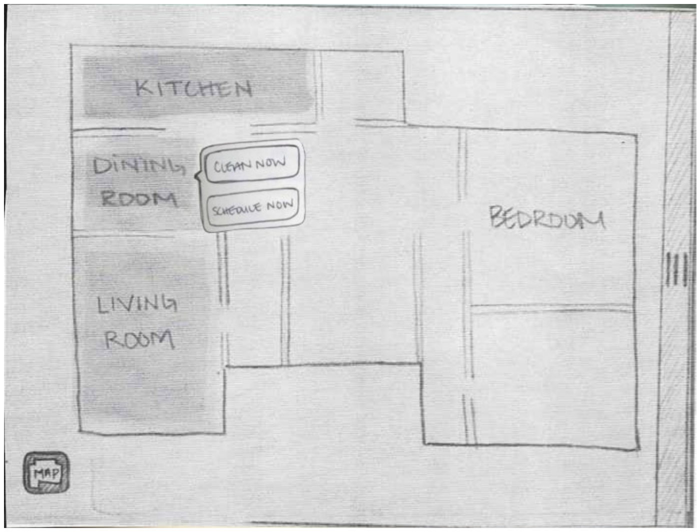


Figure 4: Map mode with context menu - perform commands relative to a given room

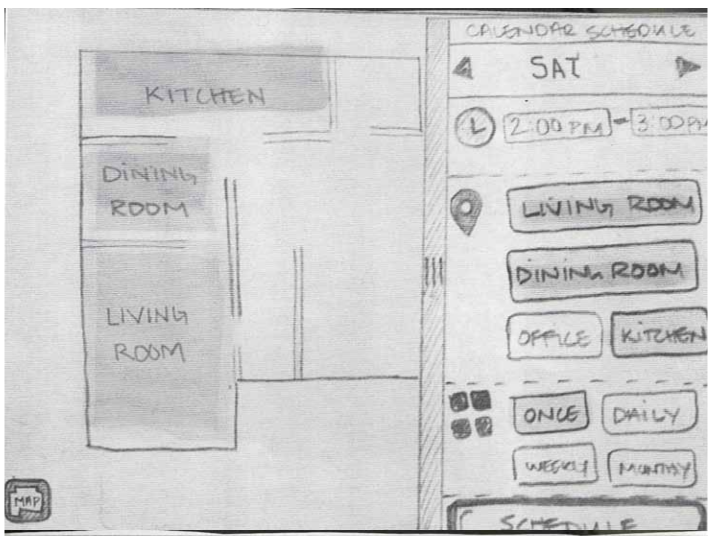


Figure 5: Scheduling screen - schedule a cleaning task, possibly with recurrence

technological experience. We tested our interfaces in their home setting, though we did not make explicit use of their space as props. In other words, we did not use wizard-of-oz testing with a simulated robot physically performing the tasks. Instead, we verbally described what the robot was doing. The home setting simply allowed the participants to be comfortable and to be in the same mindset they would during the majority of the time the interface will be in use.

We had the users perform the same three tasks as in our video prototype. First, we had a "learning" task where the robot was to learn what a room looks like when clean. We verbally told the users that the robot was in the current room with us and to assume that the room was clean. We said the robot needed to understand what the room was supposed to look like after cleaning. Since each family organizes their space differently, they needed to command the robot to "learn" the layout of their particular space. Their task was to command the robot to go about autonomously observing the room so that later when the robot cleaned it knew what state the room should be in.

Second, we had the users schedule a nightly cleaning. We said that many families may want to schedule a regular cleaning for a particular room (in this case the living room), so our system allows the user to do that. The users were to schedule a daily cleaning of the living room to happen in the evening time. It should be scheduled to take no longer than 1 hour.

For our final task, we told the users to pretend they were away from home and needed to have the childrens' room cleaned immediately. We used a scenario where the user had a dinner party that evening and wanted to come home to find the place clean. How would they go about having the robot start cleaning that room immediately?

The first step of our experimental procedure was to explain to the participant that we were trying to test our design and that we were going to test it using a paper prototype. We explained that this was just a test design and that any sources of friction encountered were not their fault but rather the fault of our design. We asked the participants to think out loud and to let us know any difficult which they ran into. We did not use any participant waiver forms.

We set up our paper prototype to displayed on a fake iPad so it had the look of an actual tablet. We explained that we would be putting paper pieces on the "tablet" that represented screens they would see. Since we did not use wizard of oz testing, we simply described the robot verbally and explained the connection between what they saw on the screen and the robot. Due to our disparate schedules and one of our teammates being ill, we could not coordinate times when we could do these experiments together. As a result, each experiment was conducted by a single teammate. In order to make up for this lack of eyes and ears, we tried to be mindful and slow and ask the user many questions, stopping them and pausing what they were doing in order to take notes.

We did not have any quantitative way to measure the users' experience of the interface, but we asked them to talk their way through doing the tasks. This meant being inquisitive and directly asking what seemed confusing. We paid attention to what happened in the "big picture" by asking the users questions like "how did you feel overall about using the interface during this task? What seemed to be particularly confusing?" We paid attention to the bottom-line by noting all specific points of friction the user encountered with the interface and listing their severity. This data could be pieced together to form an overall impression of how easily the users were able to perform the tasks with our interfaces. See Appendix B for notes.

3 Test Results

Our testing methods uncovered a number of very interesting and useful results. Unfortunately we do not have space in this report to expand all of them, but we present the most commonly occurring and severe issues here. You can get a broader sense of the issues in our notes included in Appendix B.

The first and most frequent issue which arose was our lack of function visibility. We initially designed our interface to save screen real estate by tucking commands away in a slide-out menu. We also broke the design up into the two sections: viewfinder and map, since we thought that the user would want both and logically they should be separated. By tucking commands away in a side menu and having some functionality available through the map alone, users were sometimes bewildered by where to go to perform a command. Some even had trouble finding the slide-out menu. It seemed to be too hidden and putting commands away in a screen like that meant they had to go searching each time they wanted to perform an unfamiliar task. As a result, it appears we have not made our available commands visible enough to the user.

Another common issue was that the interface did not give enough feedback about what the robot was doing. With the learning procedure we show a "learning in progress" screen along with a cancel button However, other than that, we did not really communicate the robot's state. One specific example came up during the task to immediately clean the childrens' room. We showed a navigational path for the robot, but the user did not know that cleaning occurred since no indication was given. In other words, they did not really know when the task was complete.

Corollary to this, we also failed to provide any emergency stop or cancel button for cleaning (though once again, we did for the learning procedure). Users suggested that some sort of "cancel" button would be useful for the other procedures as well so that they would be able to stop the robot from doing something it should not. This relates directly to one of the points of our VSD analysis - how can we design the system to operate safely in a domestic environment?

Generally users found the scheduler and calendar confusing. This seemed like a less severe problem than the others, but nevertheless is an important design issue to consider. How can we include a function like scheduling, which is likely the most complex function the interface currently supports, without making it too confusing? One specific issue that came up was that a user did not know how to interact with the calendar screen. She did not know if a specific block of time would be manipulable or if it could be edited once it was set. Another user was confused by the specific way that the scheduler screen worked: did the arrows represent changes to the time or to the day? This is not immediately clear without manipulating them. Overall, this sort of feedback seemed to suggest that the scheduling and calendar screens needed improvement.

4 Interface Revisions

Our first suggested change addresses the issue of feedback. We decided to abstract the idea of the in-progress menu from learning to all tasks (including learning). We will depict an "in-progress" dialogue which will show the current task and its status. We will show a persistent "cancel" button in the upper-right-hand corner which the user can press to stop the current job. This will appear when the robot performs any task. See Fig.6.

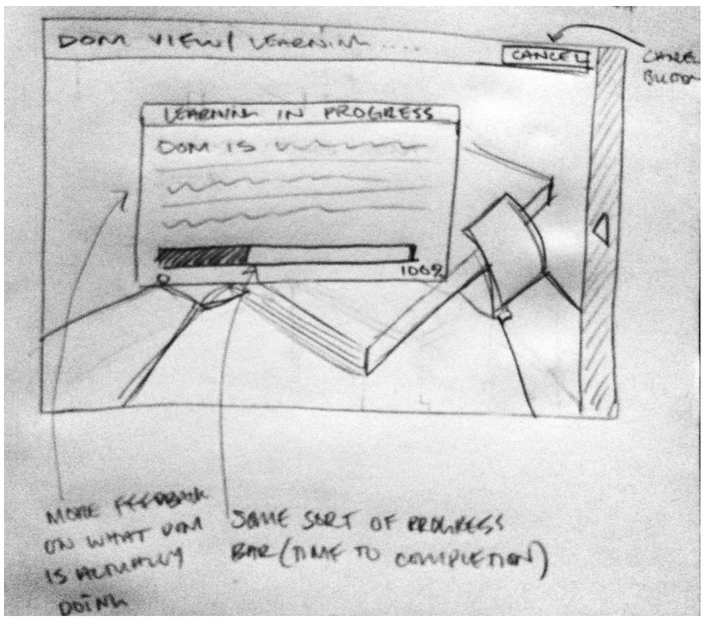


Figure 6: Pop-up dialogue for feedback

We will also add an information bar to the top of our main screen, which will now be the map. We will show the viewfinder as a smaller thumbnail in the lower left-hand corner. This should solve one of our visibility issues - the bifurcation of the menu system into two parts which have different functions, causing the user to hunt for commands. Under this new system we have one main screen with practically all of the needed functionality, making commands easier to find. The information bar at the top will change to the present context to adjust instructions for them. See Fig.7.

We also changed the slide-out menu to have an arrow. Between this and the fact that the color in the main window will contrast with the gray of the slide-out, we feel the slide-out will be much easier to find in the interactive prototype once implemented. This also helps address our visibility problems.

Showing the viewfinder on the screen at all times helps with feedback. The feedback issue is worst while making commands in the map mode since the robot cannot actually be seen doing anything in this mode. With this new set-up we effectively depict a map mode where the user can also watch through the robot's cameras. This adds another level of feedback beyond the in-progress dialogue.

When the user presses on the viewfinder, it will expand to fill most of the screen. This can be likened to the viewfinder mode found before. Rather than accessing the contextual commands which were formerly in the slide-out menu in this screen, they will be available in a pop-up located directly on the screen. The user can press this area similar to pressing on the rooms of the map, which creates a pop-up near their finger's location. Since this interaction modality will be familiar to the user from the map menu, this should be intuitive to learn. We further make this easy to learn by showing instructions at the top stating the user can press on the viewfinder for more options.

To further eliminate our former bifurcation of the menu system, we will present the same content on the slide-out menu regardless of the state of the rest of the menu. This will now show only the calendar and scheduling screens. This consistency should make it easier for the user to anticipate what the slide-out menu will show.

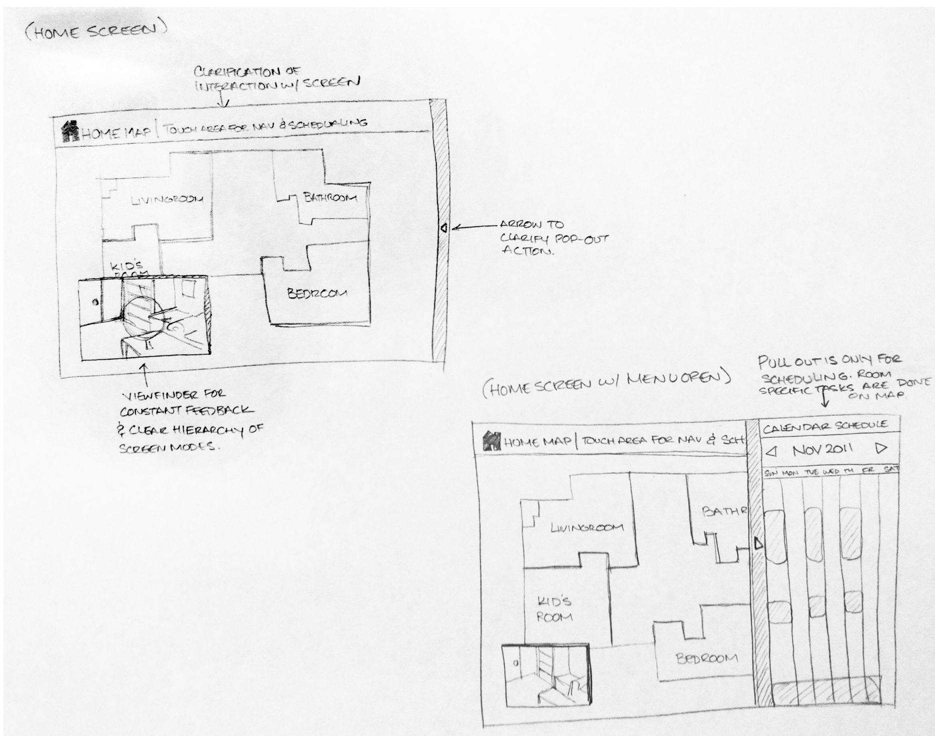


Figure 7: The new home screen - zoom in for detail

Finally, we will try to clarify one of the conceptually difficult tasks - the learning process. Rather than having an explicit learning process which the user must manually initiate, the robot will simply send a message to the interface saying that the robot does not recognize its current context. This will ask if the user would like the robot to learn the current room. See Fig.8

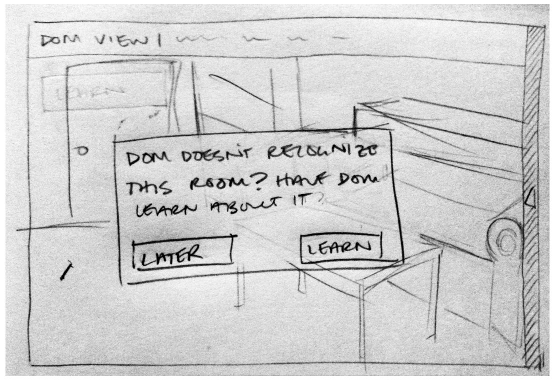


Figure 8: Prompt which appears when the robot does not recognize the room it finds itself in

5 Discussion

This testing process greatly shapes our revisions. We received several strong messages from these tests. First, we need to be careful with trade-offs between cleanliness/minimalism of the interface and function visibility. In this case it appears we emphasized the cleanliness too much and the visibility too little. One could call this effect the "sweeping under the rug" effect. We provide a rich set of features through the interface but in our original design we shoved them mostly into the slide-out menu. This made the slide-out menu too complicated while at the same time hiding away the real content of the interface. This realization will directly shape our design since we will move many of the functions directly onto the screen, making them more visible.

Second, feedback during complex processes should come in multiple forms and "back out" options should be available while the process progresses. This directly supports the safety aspects of the system. We provided feedback in our previous interface, for instance, by allowing the user to monitor the robot through the camera. However, this is really ineffective since users do not want to sit watch the the robot perform its tasks and try to interpret what the robot is doing. The interface should also depict an additional level of direct feedback that simply says what the robot is doing. A set of controls to cancel or alter the current task could allow the user to monitor the robot and ensure it operates safely. This will obviously shape our design since we will add more levels of feedback and control. One potential feature which we do not mention above is some sort of texting or email system that sends messages to the user when a task completes. While this does not support the safety feature per se, it frees the user to go about other things while the robot performs its task. This supports the original goal we envisioned for the system - freeing up time.

One thing we did not investigate here, which could have been useful, is the effect of adding color the the interface. In particular, the calendar screen uses color-coding to show the rooms to which a given schedule block corresponds. Without this color coding one of our participants was confused about what the calendar really meant. We had to verbally explain it. She responded that she thought it would be clear with color after we explained it. However, we do not really know what people's experience of the color-coding would be until we actually implement a color interface and test it.

6 Appendix A - Forms handed out to participants

We handed no forms out to participants other than the interface itself (if it counts as a form). In this case the tasks were simple enough that they did not have complicated details which needed to be recalled. As a result, we did not feel we needed to write them down for the user to refer to. Also, we chose not to use any waiver forms since that complicates the process and can scare participants away. We simply explained that we will not use their names when summarizing the results and that they can feel confident in giving us honest answers.

7 Appendix B - Raw notes

You can find our notes here. We use the following keys to reference the tasks:

- Task 1: Schedule a nightly cleaning
- Task 2: Clean a single room immediately
- Task 3: Learn the layout of the current room

7.1 Subject 1 - 50 years old

- Task 1,3: where to go to schedule or learn? no information on the screen about where to go. where is the command located? (severity 3 - the command is available, so its still possible to perform the task, but its a serious pain to find. Impact medium high, frequency common since other commands potentially have the same problem.
- Task 1: calendar after scheduling: black/white confusing. after explaining color coding to her she indicated it seemed to make sense. may want to clean this screen up and add am/pm indicators (severity 0 - we do not know that this color coding is actually a problem until we prototype the interface in color and test it. severity 1 for cleanliness of the screen. it seemed to be useable, just takes initial effort to understand) frequency high but impact low.
- Task 2: Feedback after task is complete (other than learning). How do I know that the robot is done? (severity 3 - this seems like a pretty serious issue. How does she know she can issue another command without interfering with another task? Closing the control loop/providing feedback should be emphasized much more) Frequency high, impact relatively high

7.2 Subject 2 - 47 years old

- Task 1: presses the map button because “it is the only thing that looks like something to press” (severity = 2. Will lead to future problems when she cannot find the side slide out menu) frequency high since the slide menu contains so much stuff; visibility is very important. Doesn’t know expected feedback of pressing map button (severity 1 = can click to discover but would be better if expectations were known ahead of time)
- Task 2: What if she wants to schedule the whole house to be cleaned? She has to click every room (severity = 2. Inconvenience but easily added feature) Frequency unknown, impact perhaps medium. Could be an annoyance for some users
- Task 3: cannot find the slide out menu bar (severity = 5. In this situation, it was critical for her to find this menu and she was not able to without prompting). impact and frequency high. does not understand that there is a step after naming the living room. (severity 5 = cannot move forward without pressing ’learn.’ wording of the button needs to be clear and program should prompt next step). impact high since this means the task cannot be completed without further clarification of the interface.

7.3 Subject 3 - 29 years old

- Task 1: participant expressed that he didn’t know when something was scheduled or how to re-edit, no sort of confirmation (SEVERITY = 2–minor usability problem) frequency medium, impact low. buttons vs. non-buttons (SEVERITY = 1 —cosmetic problem, high-fidelity mockups will resolve this) frequency high, but impact very low
- Task 2: wanted to see more capability, not just clean and schedule, but just navigating the robot around (SEVERITY = 2–minor usability problem) impact medium, will add feature. error prevention—participant expressed that there was no place to cancel the act (SEVERITY = 4— major usability problem) impact high
- Task 2: unsure of what the default screen was or confused of which screen was used for what action (SEVERITY = 2/3— important was not quite major, easy to combine screens) impact medium, frequency high. unsure what ’progress’ really meant —- no feedback on how close the robot was to completion or what it was actually learning (SEVERITY = 3— major usability problem) frequency and impact high

7.4 Subject 3 - 43 years old

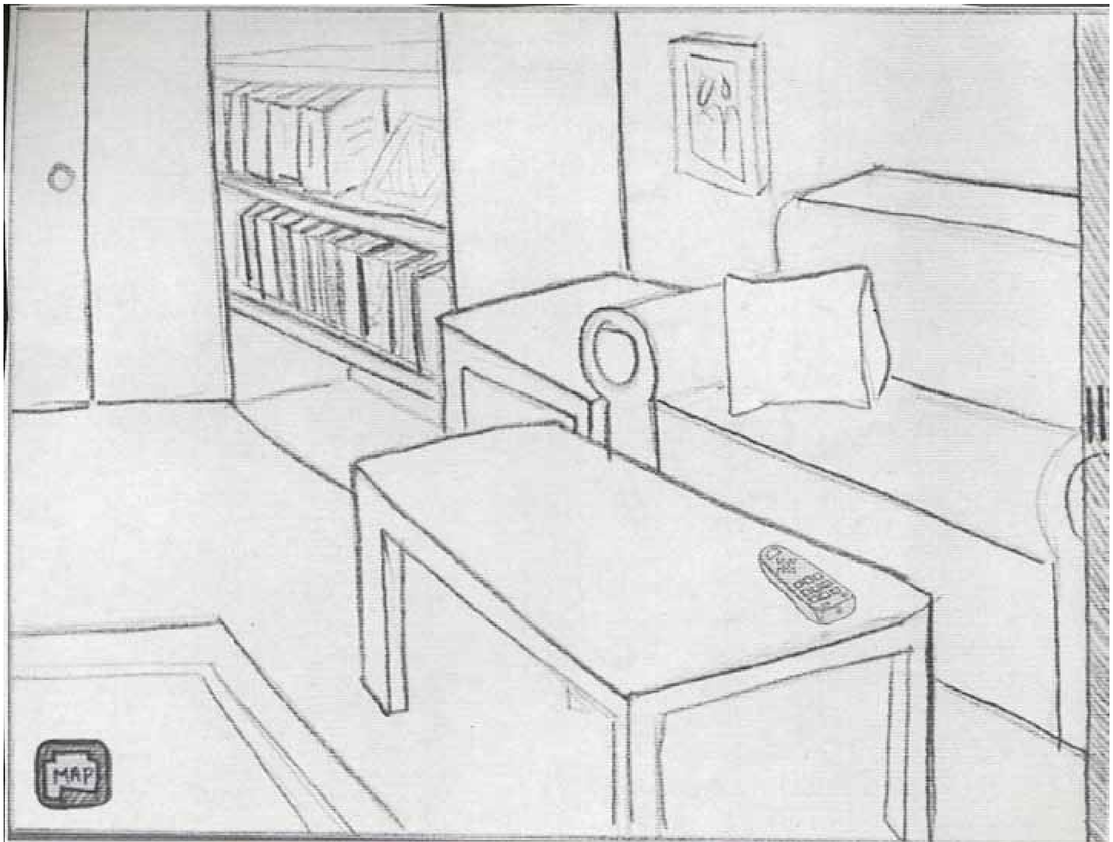
Note this note-taker did not organize by task, but the notes are nevertheless quite complete.

- on the scheduling screen where you pick the time and stuff, she thought the arrows on either side of the day controlled the time (severity 1 - minor cosmetic issue) impact low

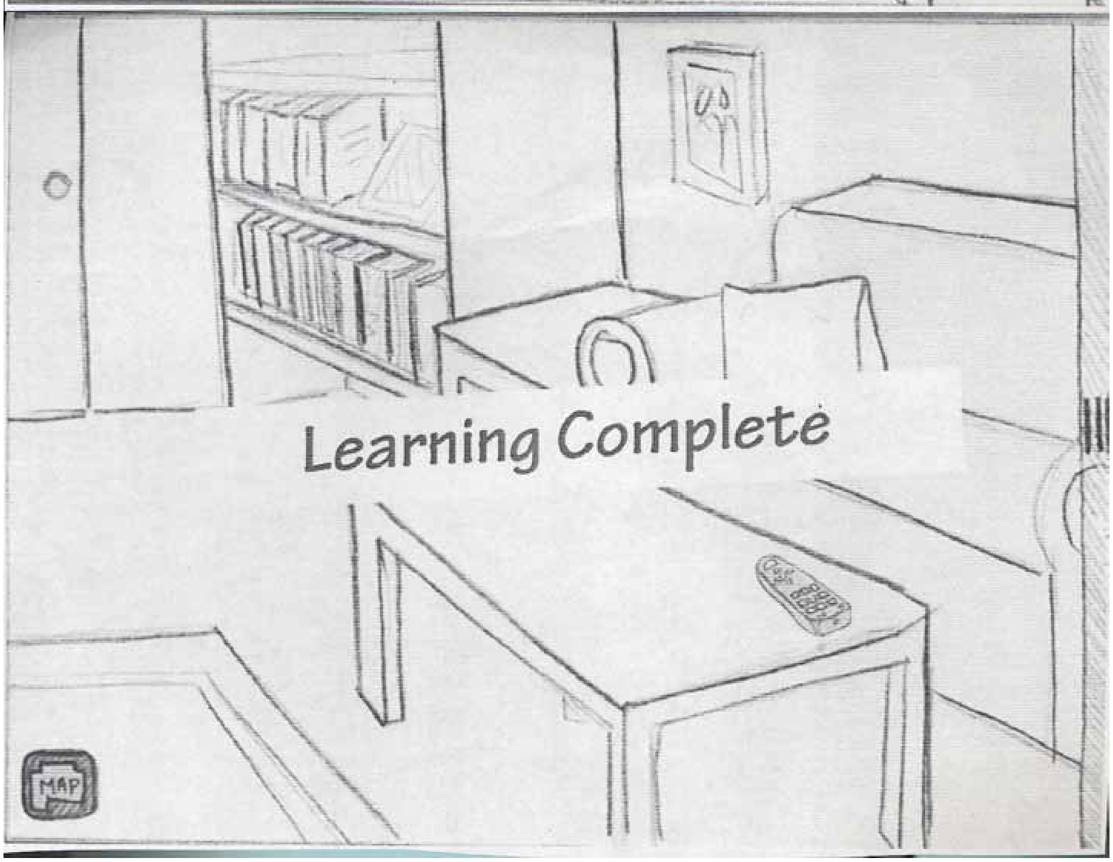
- on that schedule screen, we need to figure out how the time changing menu works. It is unspecified (severity 2 - Minor usability problem, we just haven't thought about how it works) frequency medium, impact unknown
- she needed a way to stop a cleaning in progress. Didn't know how to. (severity 4 - Catastrophic. Need to be able to do this.) impact very high, frequency high
- she didn't know to pull out the side menu to access the train command. didn't know she could drag it out. (severity 4 - catastrophic you need to access the menus) impact and frequency very high
- didn't know that she needed to make sure a room was clean before teaching the robot. should indicate this somehow (severity 0 - people should learn this somehow outside the interface). impact none in practice
- had to wait for a robot to go to a room to start teaching. shouldn't have to do this. She didn't know when the robot was done moving without waiting and looking at the map screen. Should alert somehow when it is done moving. (severity 3 - waiting is not fun and knowing status is important especially when they first get the robot and need to train it) frequency high

8 Appendix C - additional figures of of the paper prototype

In the interest of completeness we include the entire paper prototype here. We do not reference it in the text since we also have screen shots in-text. This provides any additional details that may be unintentionally missing or confusing in the text.



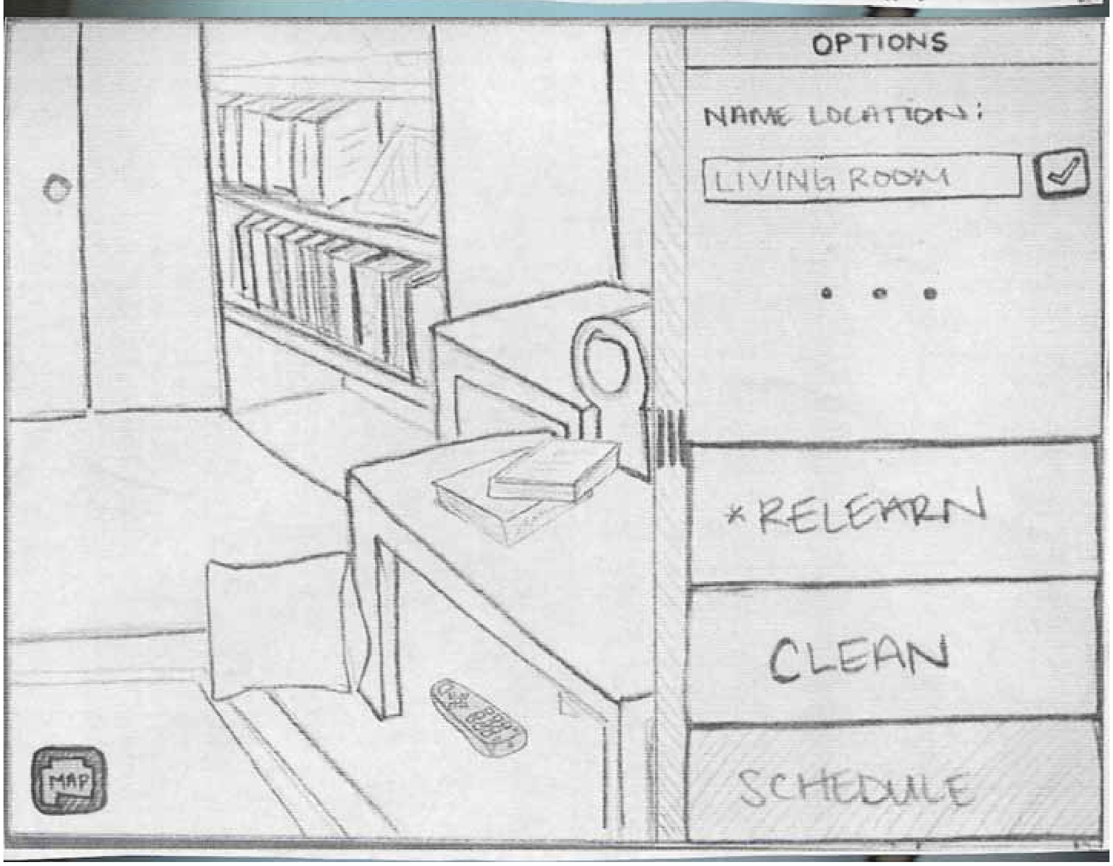
1



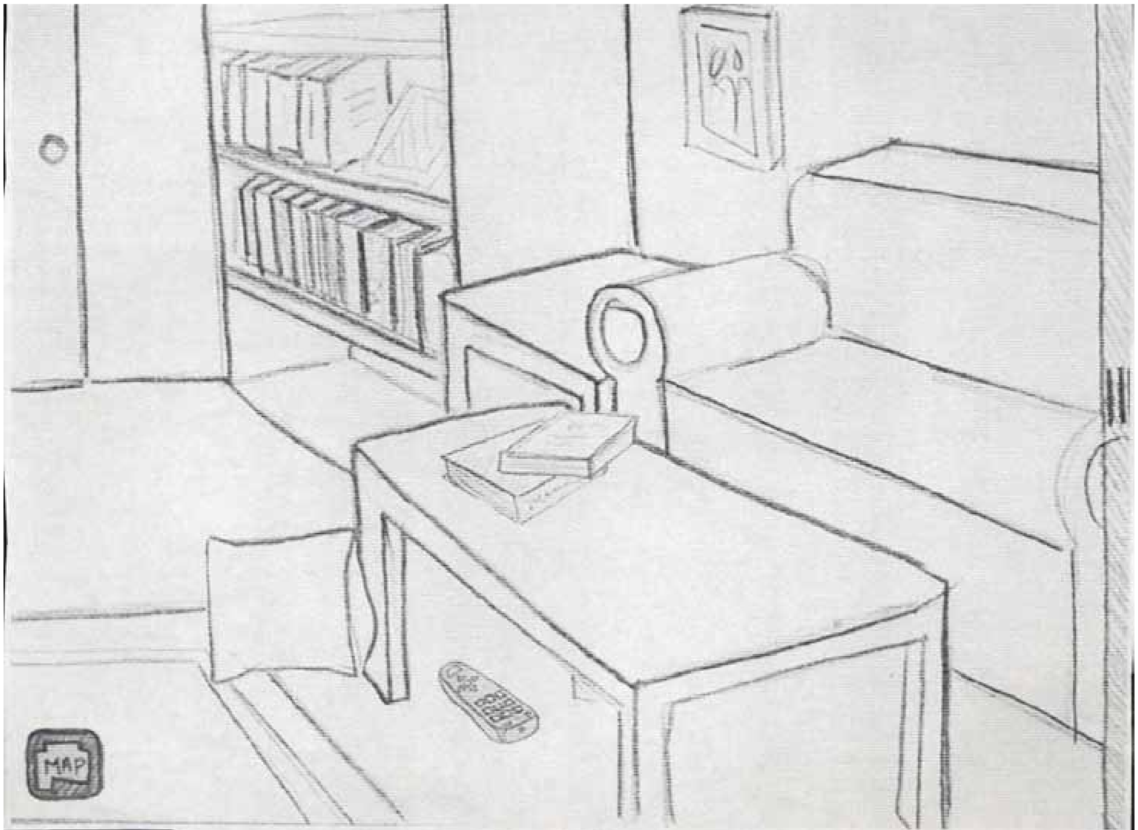
2


LEARNING IN PROGRESS...
MAP

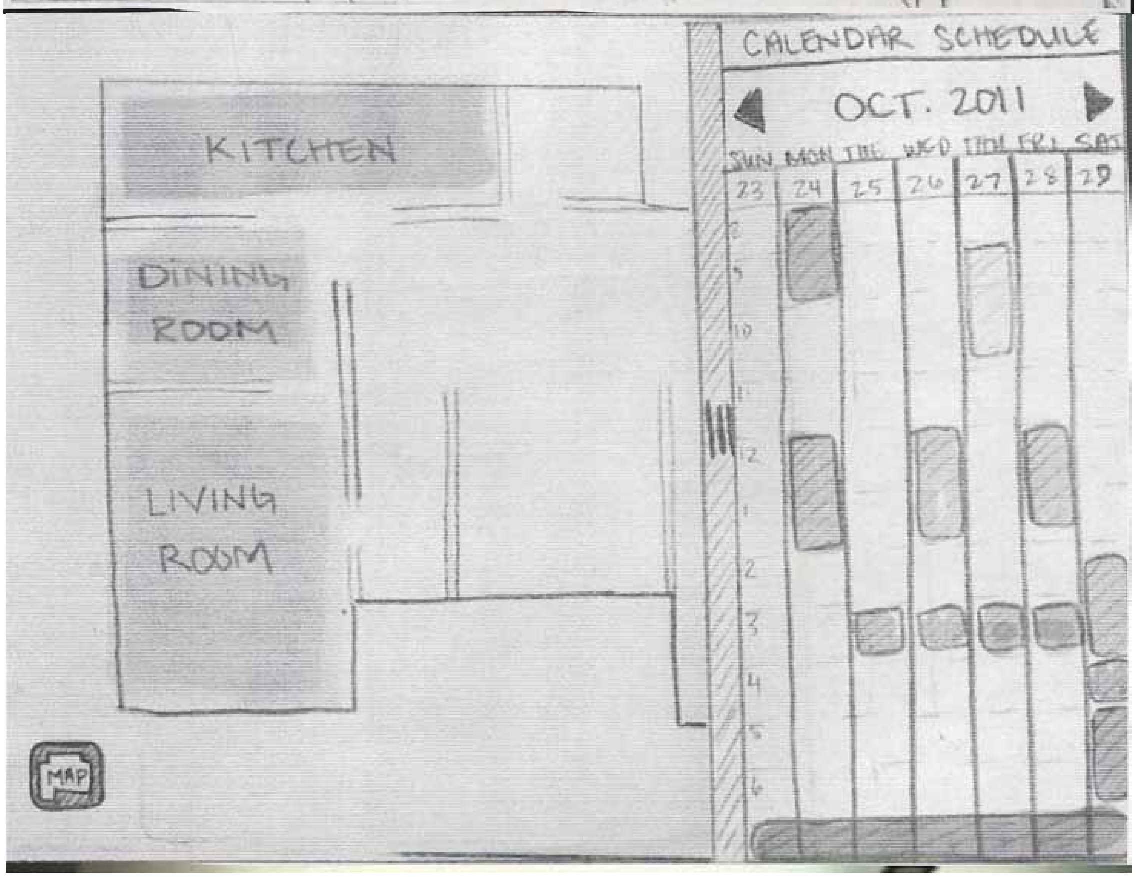
3


OPTIONS
NAME LOCATION:
LIVING ROOM
...
*RELEARN
CLEAN
SCHEDULE
MAP

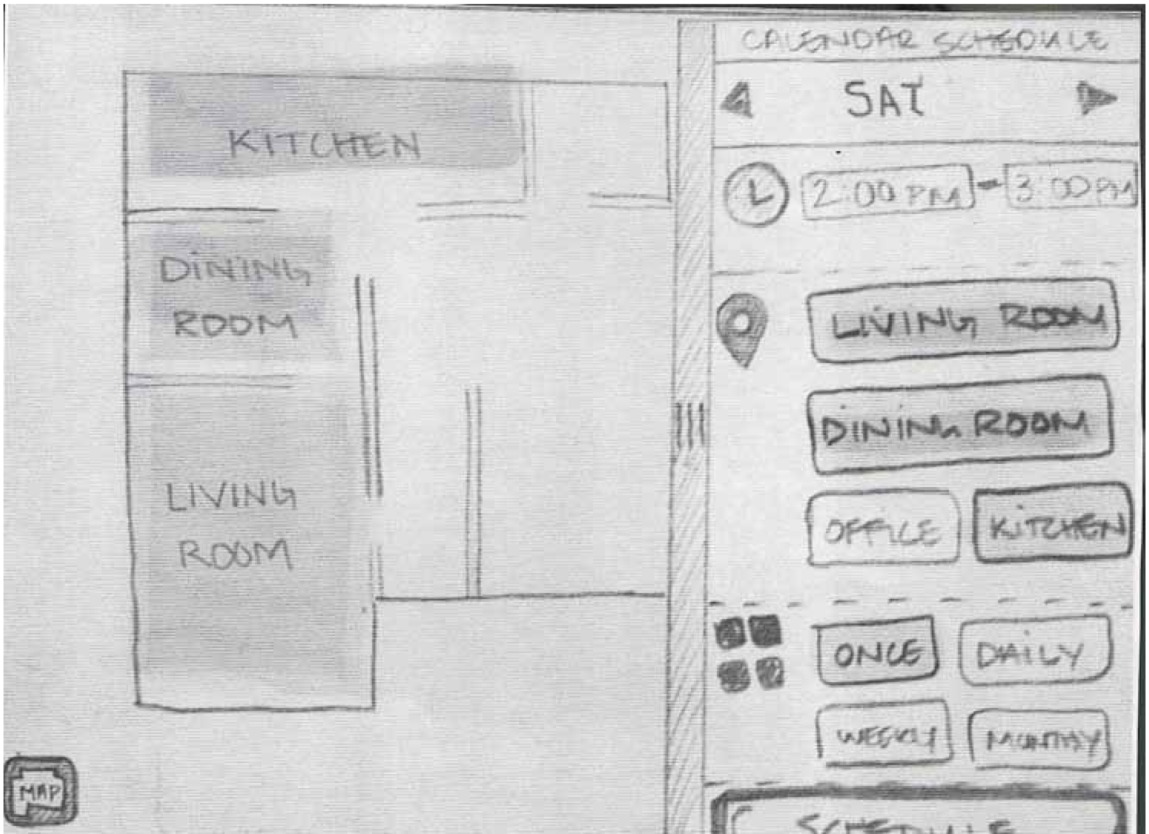
4


MAP

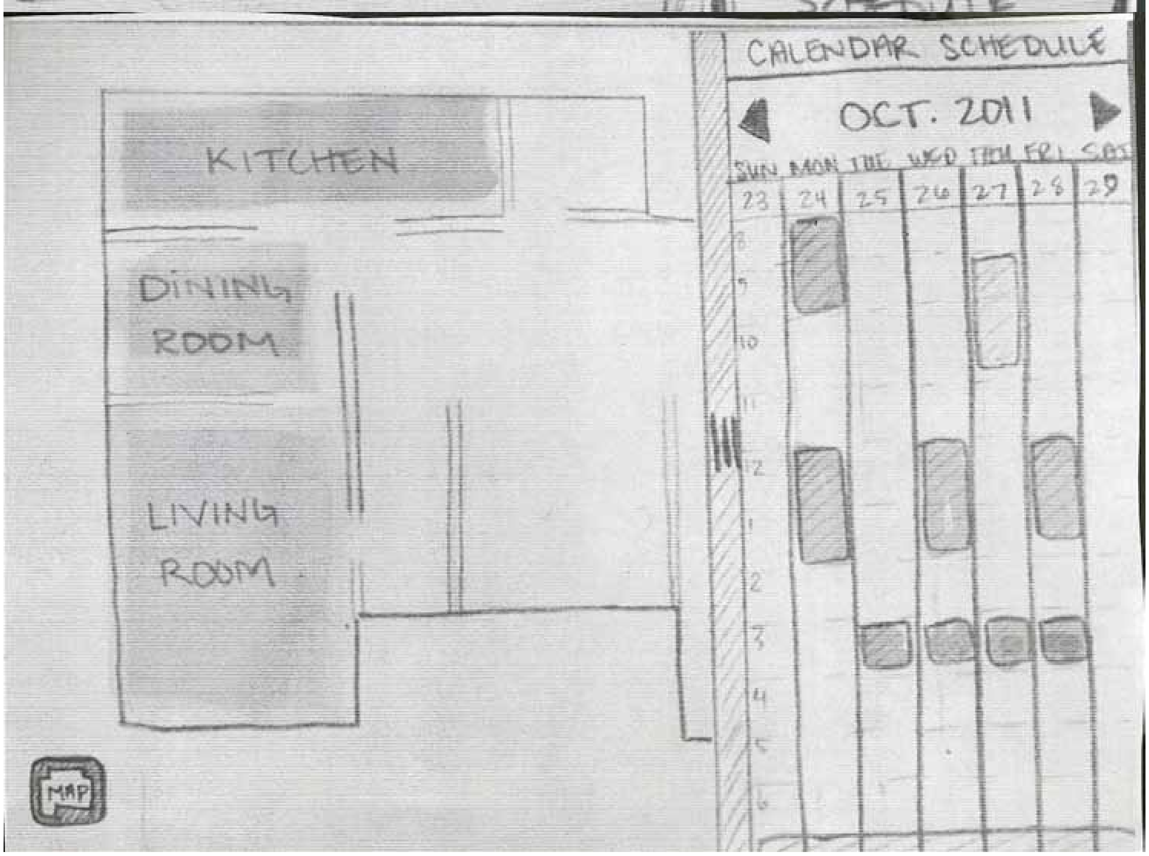
5


KITCHEN
DINING ROOM
LIVING ROOM
MAP
CALENDAR SCHEDULE
OCT. 2011
SUN MON TUE WED THU FRI SAT
23 24 25 26 27 28 29

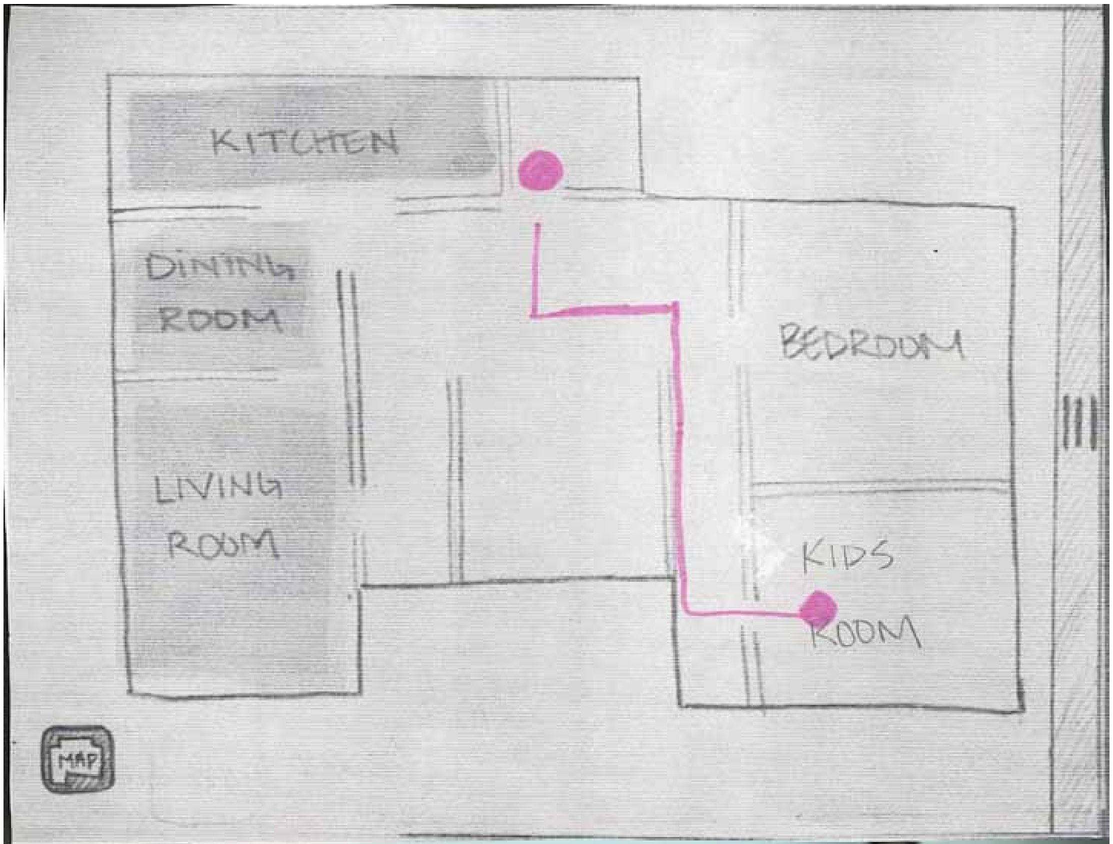
6


KITCHEN
DINING ROOM
LIVING ROOM
MAP
CALENDAR SCHEDULE
SAT
2:00 PM
3:00 PM
LIVING ROOM
DINING ROOM
OFFICE
KITCHEN
ONCE
DAILY
WEEKLY
MONTHLY

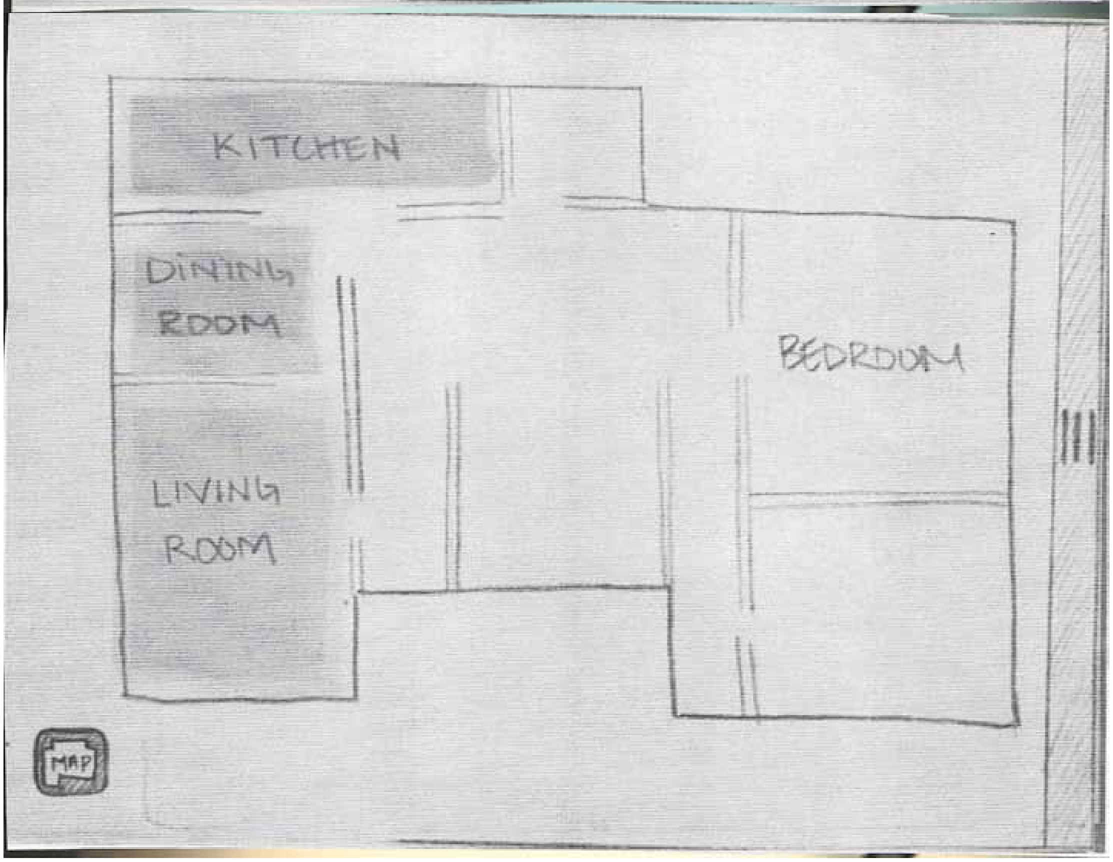
7


KITCHEN
DINING ROOM
LIVING ROOM
MAP
CALENDAR SCHEDULE
OCT. 2011
SUN MON TUE WED THU FRI SAT
23 24 25 26 27 28 29

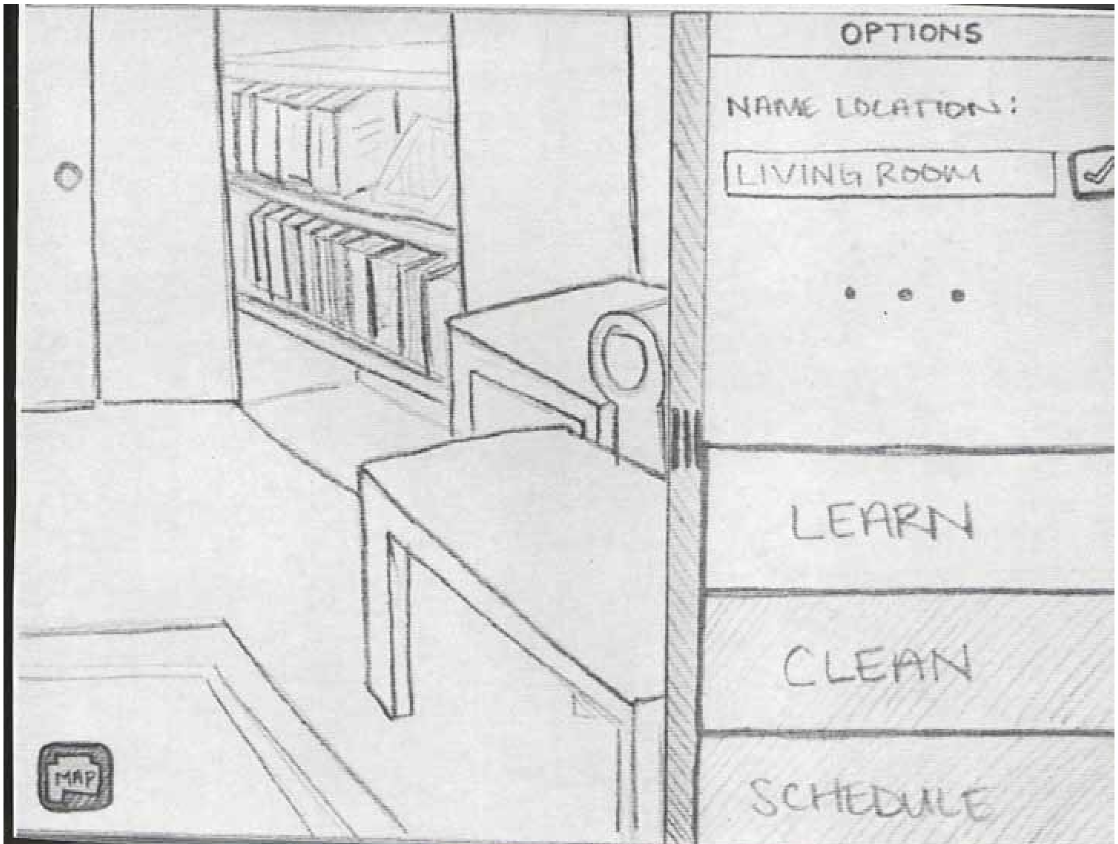
8


KITCHEN
DINING ROOM
BEDROOM
LIVING ROOM
KIDS ROOM
MAP

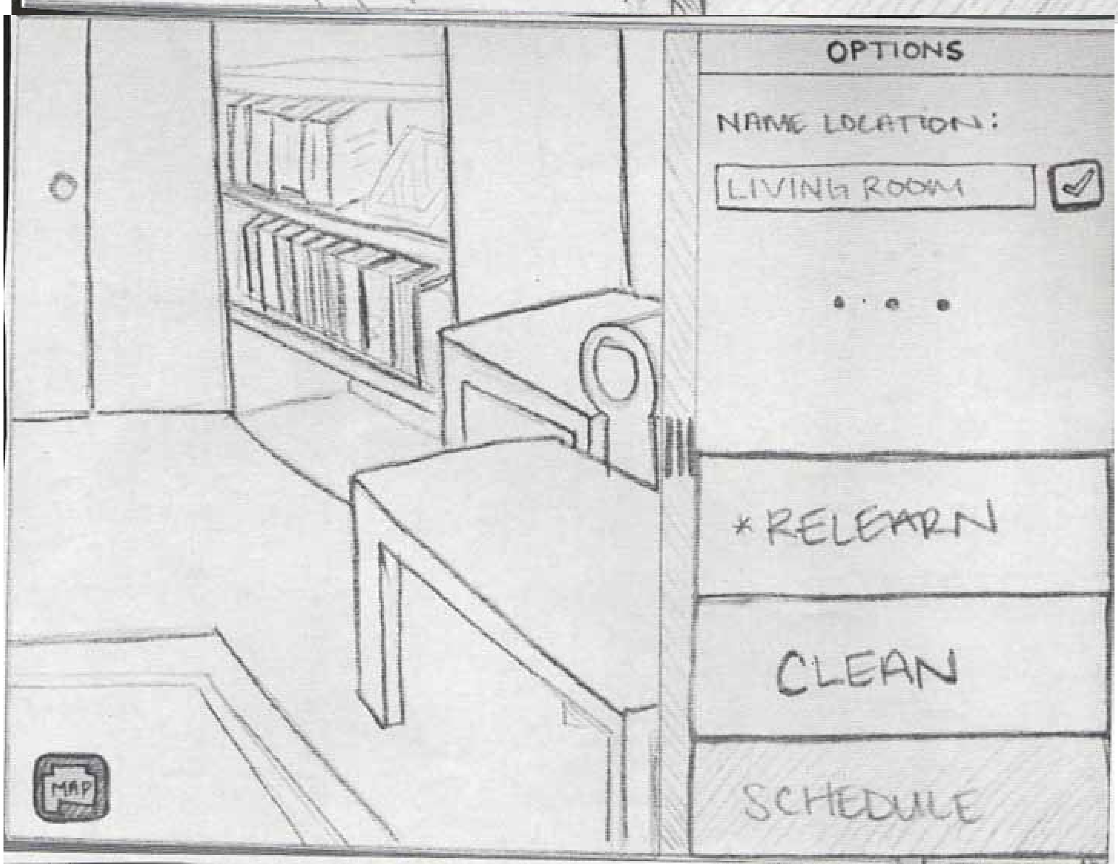
9


KITCHEN
DINING ROOM
BEDROOM
LIVING ROOM
MAP

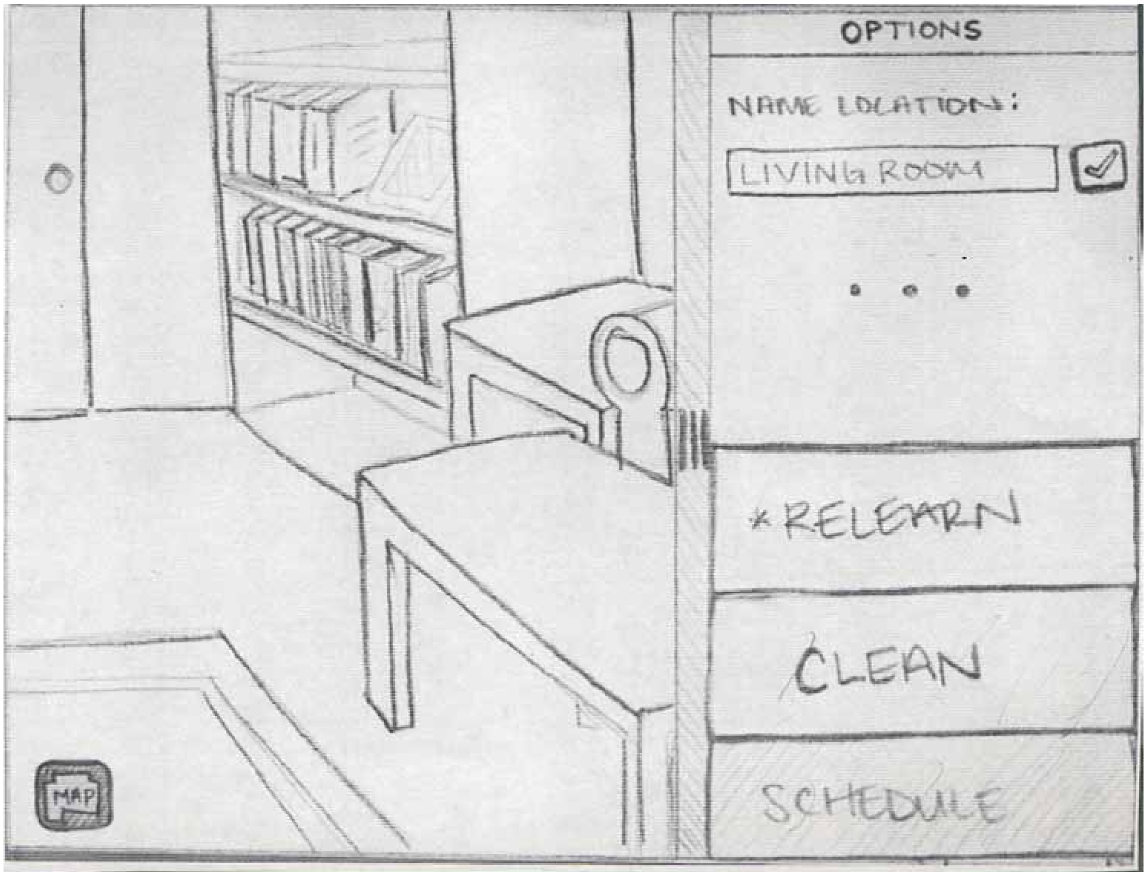
10


OPTIONS
NAME LOCATION:
LIVING ROOM
LEARN
CLEAN
SCHEDULE
MAP

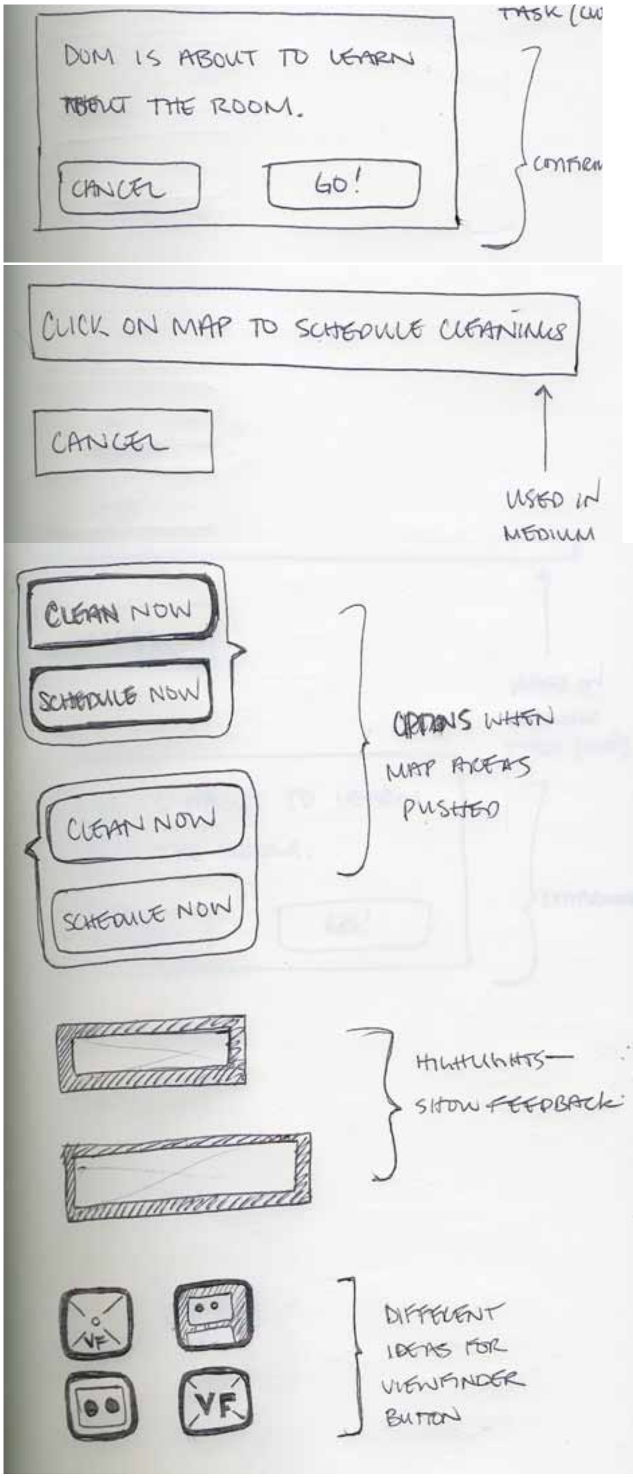
11


OPTIONS
NAME LOCATION:
LIVING ROOM
*RELEARN
CLEAN
SCHEDULE
MAP

12


OPTIONS
NAME LOCATION:
LIVING ROOM
* RELEARN
CLEAN
SCHEDULE
MAP

13



Additions to design ----- you will need to cut out and reference photo and notes for sequence if you forget