Design Critique of the House Maintenance Management Application

By -Team simplifiers, Abhimanyu Abhinav

April 8,2020

Description:

This document enumerates problems and good aspects of the design in the recommended UAR format. There after it entails the summary of the problems found in the design and suggested solutions/recommendations to improve on the problems suggested.Finally it summarises the effort out in and the tools used to produce this report with any suggestions to improve on them.

Executive Summary:

The design in contention certainly had few good points which in my view were really appreciable. The features such as “notifications”, “Messaging” in the application are really cool as it allows users to keep updated real time and sort their queries and minor contingency issues comfortably. Also customisation while account creation according to different roles with which a user might choose to use the application and upfront “Emergency Maintenance Number” flashing would certainly make the user experience a lot more easy.

Having said that there were some problem aspects also with the design basically pertaining to “Navigation and Freedom” , “Visibility of system status”, “Error Prevention” , “Recognition rather than call”, “Mapping between real and system world” heuristics. Though majority these problem aspects did not have detrimental impact on the goal achieving that this application tried to carter to but improving upon these problems aspects would certainly make the application more intuitive and more rich in terms of functionality.

The good and problem aspects are mentioned in detail in the UARs furter in the report.

Introduction:

The application is basically about letting users book for their household maintenance services and being able to track the services progress. Additionally users can contact via text messaging feature of the application to their individual service provider and clear any of their queries along with receiving timely updates in the form of notifications. The application has a very clear and neat UI to enable the above mentioned features to the users in the form of "Requests", "Messages", "Notifications" "Maintenance Form" screens apart from Obvious "Login" and "Create Account" screens.

There aren't any peculiar jargons that have been used in the application that one needs to understand except for the word "Paused" that is being used as one of the request status. But this has been identified as one of the minor system status problem aspects and sought to be made more self explanatory.

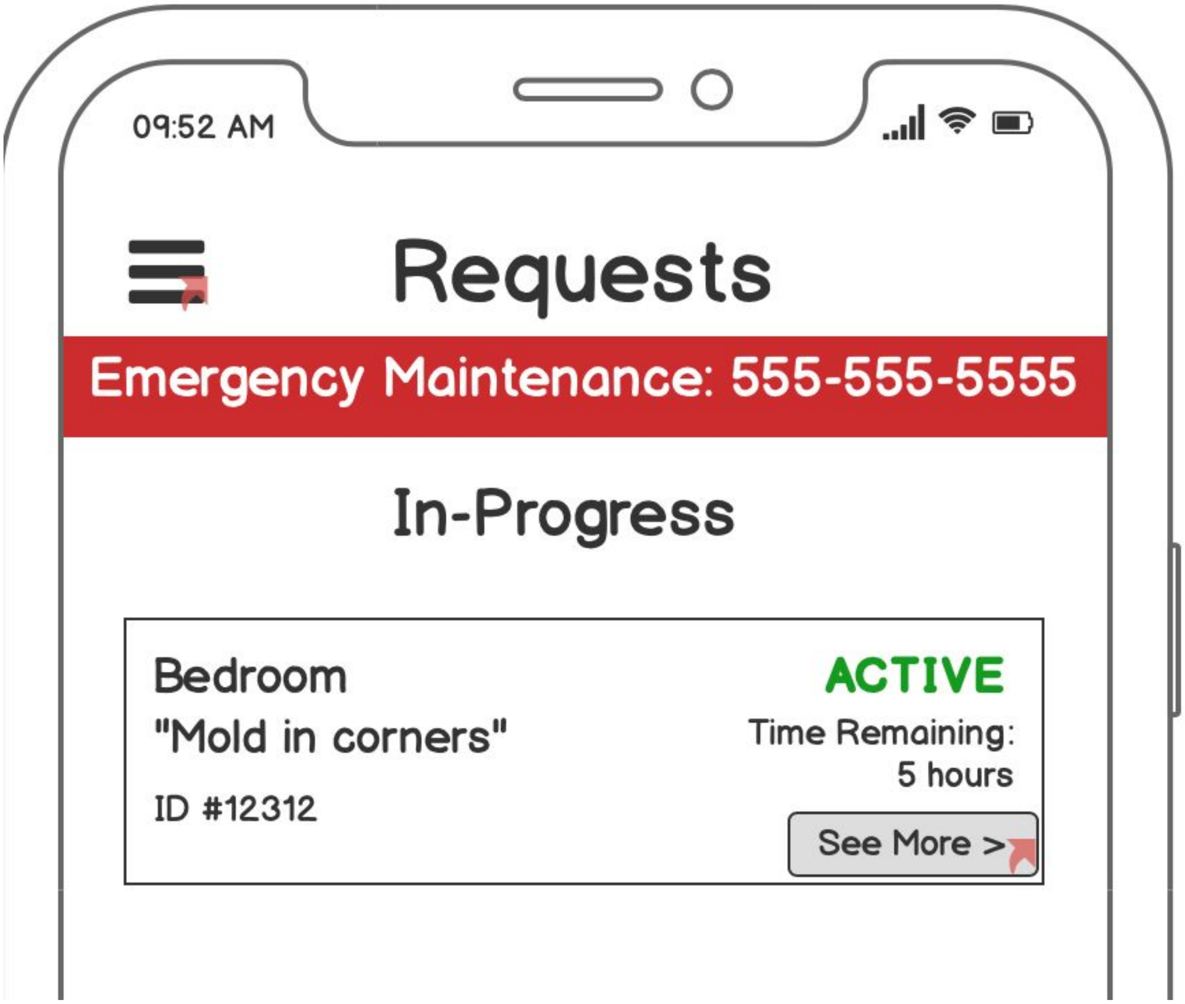
UARs(Usability Aspect report):

UAR 01

No. **Abhimanyu - 01**	**Problem01**
Name: Time remaining information is ambiguous.	

Evidence:
This screen lists the requests that have been raised by the user along with summary heading and details. In each request thread, “Time remaining” label is the problem aspect.

Interface aspect:



Requests

Emergency Maintenance: 555-555-5555

In-Progress

Bedroom

"Mold in corners"

ID #12312

ACTIVE

Time Remaining:
5 hours

Heuristic:
Visibility of system status

Explanation:
The time remaining info can be misunderstood by the user as time to process the request or otherwise as time before this request should be fixed. Hence this would not help in painting a clear idea to the user about his maintenance request.

Severity or Benefit:

Rating:
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):
Frequency: High
Impact: Low
Persistence: Low

How I weighted the factors:

Since this aspect is going to interface each user and everytime they raise a request in the system, it’s going to be a high frequency problem.Also this is going to affect the users regardless of their system expertise.
The impact though, in my view is going to be low as this would not affect the users ultimate aim of raising a request in the system just that users will not have a clear idea about the time in which their problem is going to get resolved by just looking at this request entry.
Persistence is also going to be low in my view as users are not constantly presented with this time estimation and the application does provide it’s users with messaging functionality where they can clarify this aspect.

Possible solution and/or trade-offs:
A simple change in messaging label would help in conveying exactly the message that is meant to be conveyed. For the trade-offs, in my view there doesn't seem to be any trade off for this solution and things are going to be more clear .

Relationships:
This UAR is not related to any other UAR.

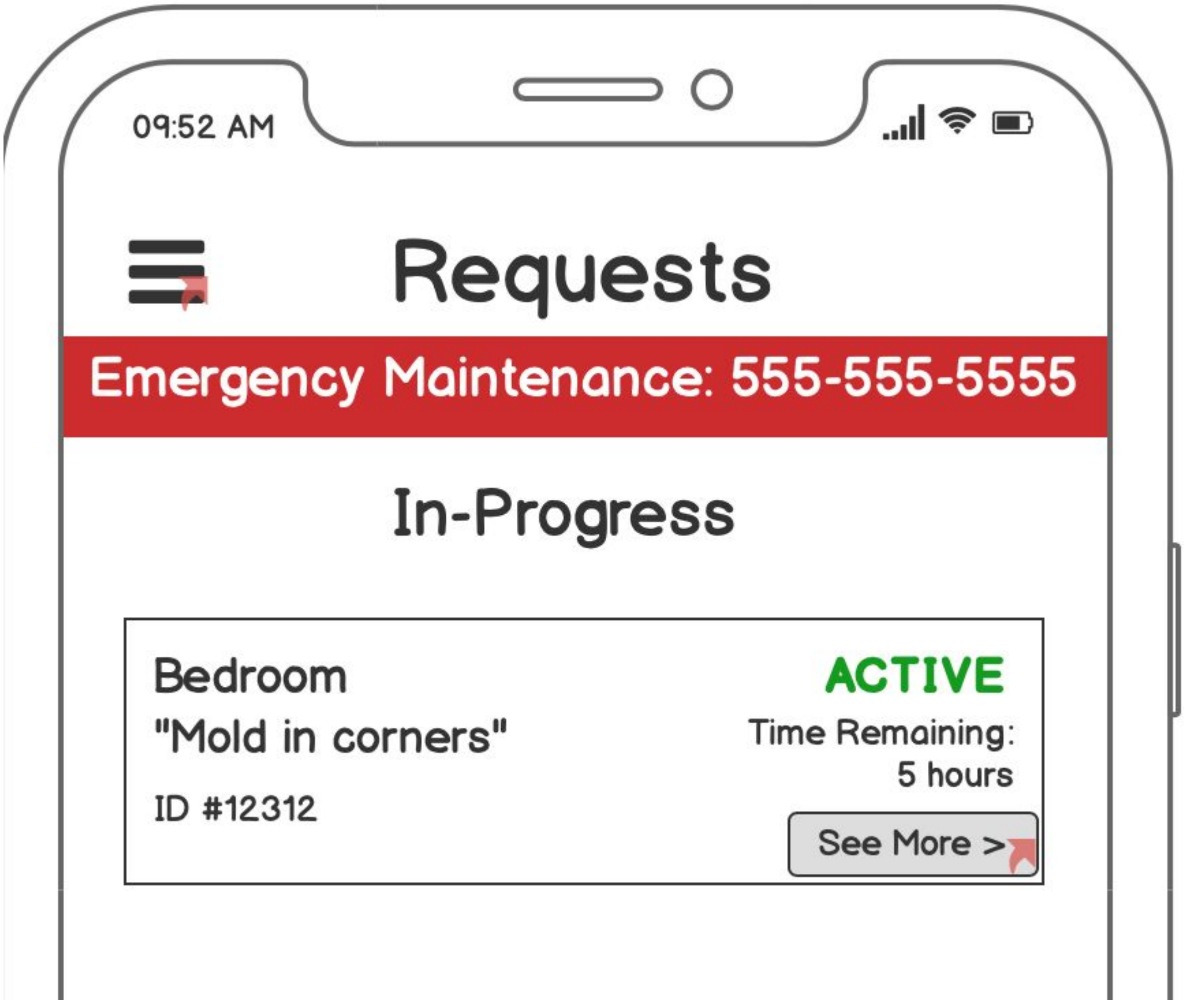
UAR 02

No. **Abhimanyu - 02**	**Problem02**

Name:
Making an emergency maintenance call from the system is not very obvious/present

Evidence:
This screen flashes an emergency maintenance number on requests listing page without a provision of making a call.(or if this option is there it is not very obvious)

Interface aspect:



Requests

Emergency Maintenance: 555-555-5555

In-Progress

Bedroom
"Mold in corners"
ID #12312

ACTIVE
Time Remaining:
5 hours

Heuristic: User control and freedom(Navigation)

Explanation:
Here I do assume that just by clicking on the red bar , the application would not be making a call to the emergency maintenance center and a user might want to call at the emergency maintenance from the application itself rather that going in his phone app and calling from there. In such a case this can be a major issue.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: Medium
Impact: High
Persistence: Low
How I weighted the factors:

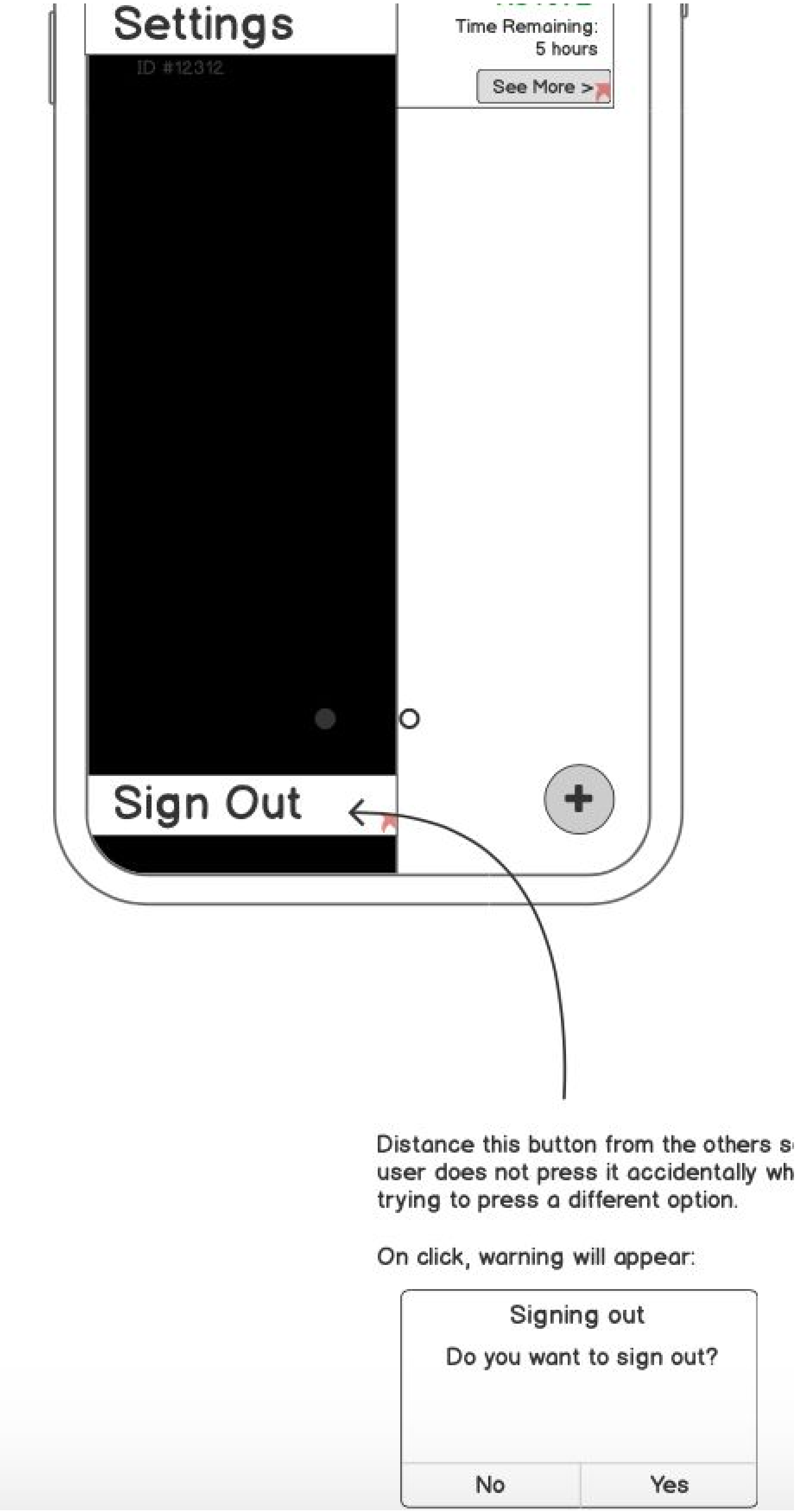
I rated the frequent factor as medium assuming only 30-50% of the cases are going to be emergency cases.
For the impact factor, I think this aspect is really going to be detrimental in users wanting to make a emergency request from the app and take users a lot more time that otherwise.
The persistence is going to be low as once the user makes his emergency requests somehow, he will not have to be bothered about this anymore.

Possible solution and/or trade-offs:
A possible solution to this problem could be to include a Phone symbol at the right corner and implement a dialog box with a message ”Make an emergency call” with YES/NO option as an error prevention mechanism. With this possible solution there seems to be no(significant) trade-offs in my view.

Relationships:
This UAR is not related to any other UAR.

UAR 03

No. **Abhimanyu - 03**	**Problem03**
Name: Signout Option placing	
Evidence: The interface here is an options menu that shows up on the click of hamburger menu. **Interface aspect:**	



Distance this button from the others s
user does not press it accidentally wh
trying to press a different option.

On click, warning will appear:

Signing out

Do you want to sign out?

Heuristic:

User control and freedom(Navigation)

Explanation:

Though I know that “Signout option” has been deliberately placed afar at the bottom as a measure of error prevention, nevertheless this brings in the possibility of the user not being able to find this option or missing it as this is located in isolation at the bottom.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: Low
Impact: Low
Persistence: Low
How I weighted the factors:

I have rated all the factors as low because in my view this particular problem isn’t going to affect the goal as such and rarely people logout of their apps instead just close it.

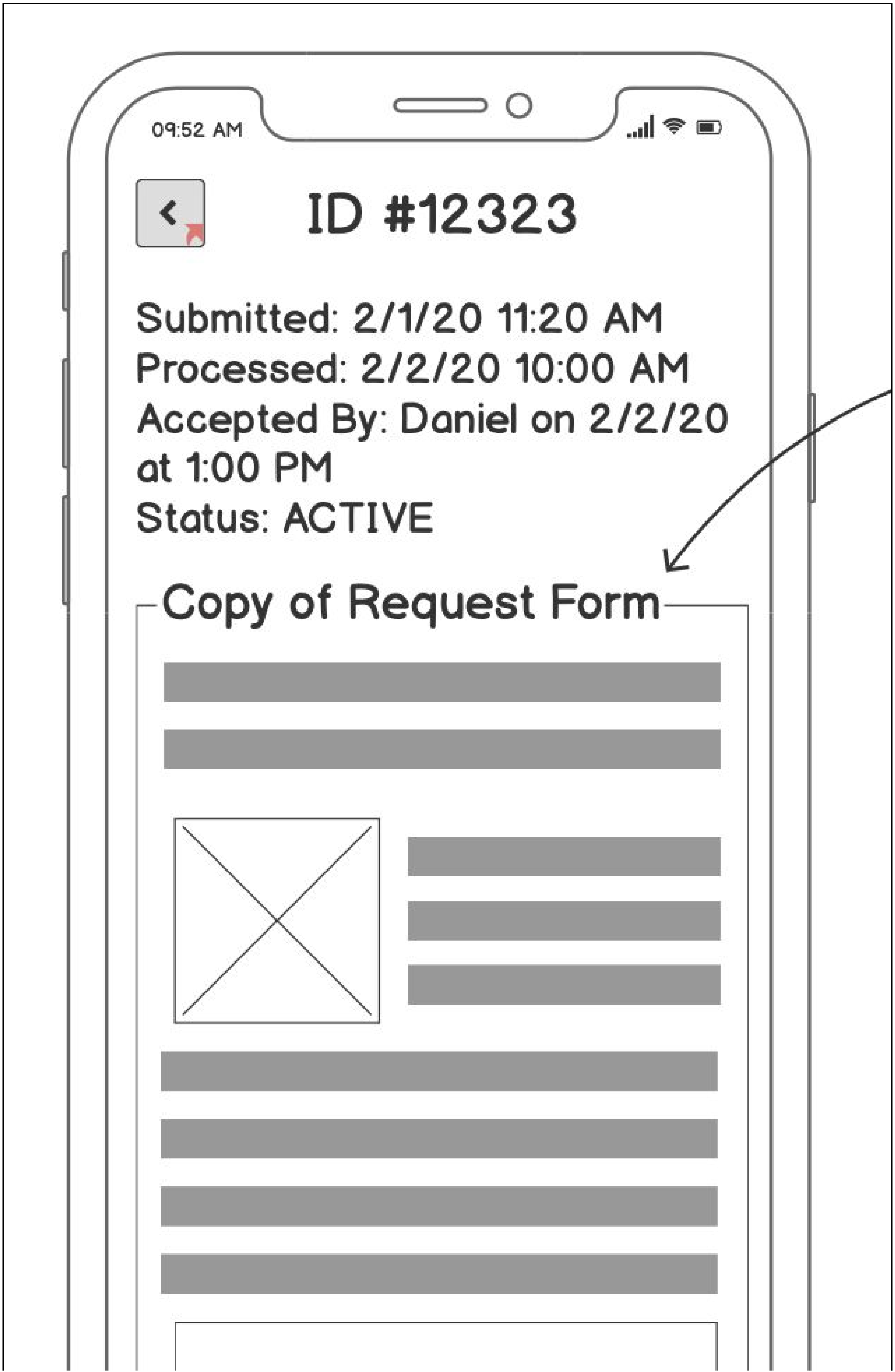
Possible solution and/or trade-offs:

A confirmation dialog box which already has been provisioned as an error prevention mechanism should be enough to prevent accidental sign-out. But to make it more error free making the sign-out button little larger in width than the other options providing larger surface area would also be an effective solution measure. The possible trade-off would be violation of the “ standards and consistency” heuristics but the possible solution in my view would make aesthetics a little more better.

Relationships: This UAR is not related to any other UAR.

UAR 04

No. **Abhimanyu - 04**	**Problem04**
Name: Request Withdrawal/Defer option not there.	
Evidence: The following screen provides the details of a request that a user can opt to see but does not give him a defer/withdraw option. **Interface aspect:** .	



ID #12323

Submitted: 2/1/20 11:20 AM
Processed: 2/2/20 10:00 AM
Accepted By: Daniel on 2/2/20 at 1:00 PM
Status: ACTIVE

Copy of Request Form

Heuristic: User control and freedom(Navigation)

Explanation:
In cases where a user gets a waiting period of say 3 days to get his tap leakage fixed but he is scheduled to be out of town on that day or just a day prior to it, then he might want to withdraw his request or defer it.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

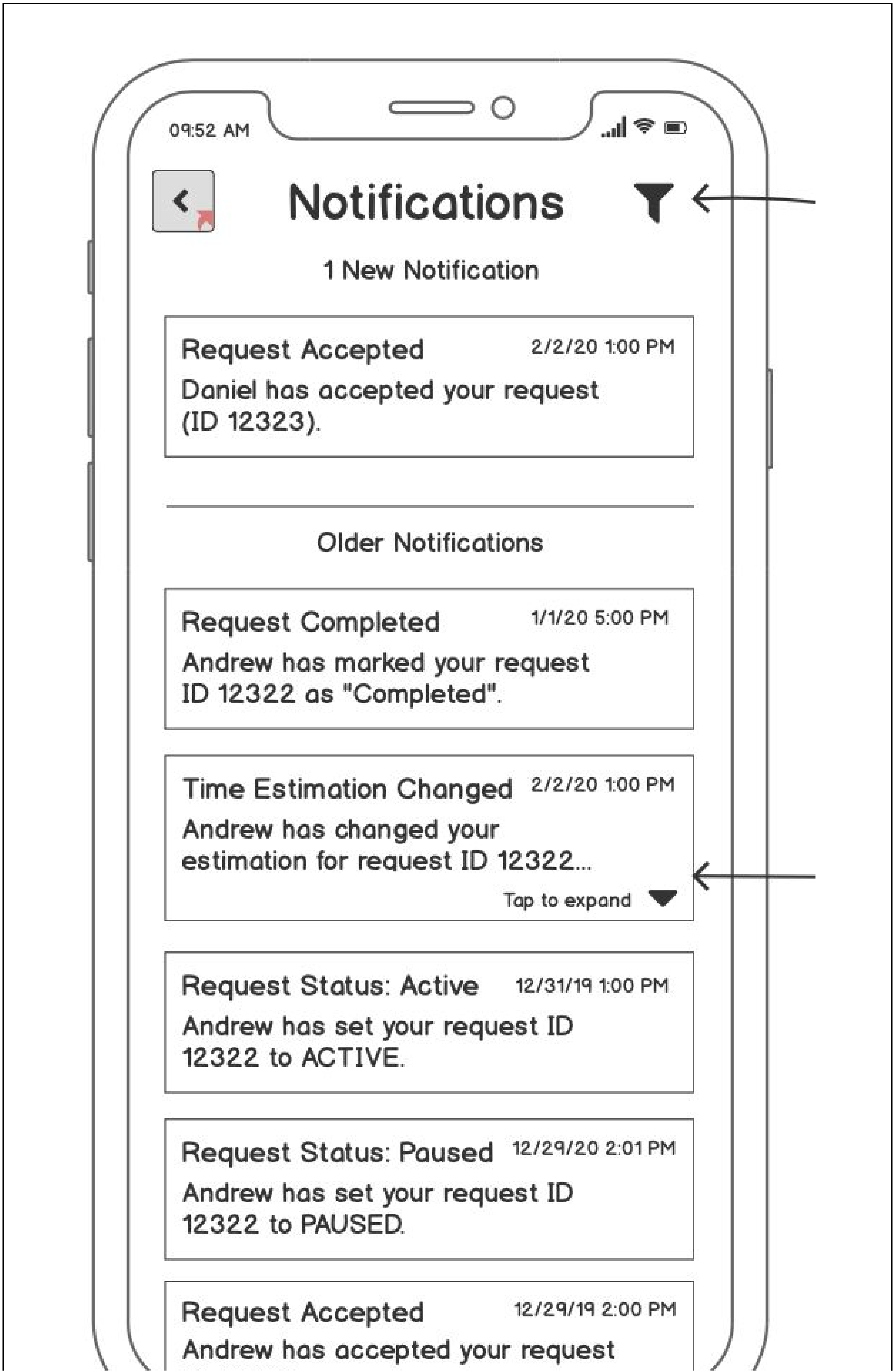
Justification (Frequency, Impact, Persistence):
Frequency: Medium
Impact: High
Persistence: Low
How I weighed the factors:
I rated frequency as medium assuming there's going to be at most 4-5 cases out of all the requests raised in the system where the user chooses to either defer or withdraw their requests, hence medium frequency would be my guess.
For the impact, I think it's going to be very detrimental to the goal of deferring/ withdrawing an request from the application itself.
Persistence of this problem should be low as this in my view should be a one time problem which when dealt with once , should not be a problem.

Possible solution and/or trade-offs:
Providing a "Defer" or "withdraw" mechanism in the application possibly providing a button for it at the bottom on this screen would be a possible measure .
The possible trade off would be that there would be one more functionality to implement in the application that might make it little tricky to implement and at the same make the application complex to design.

Relationships:
This UAR is not related to any other UAR.

UAR 05

No. **Abhimanyu - 05**	**Problem05**
Name: Not enough details and also no way to see details about a raised request from the notification prompt to which this notification is linked.	
Evidence: This screen lists all the notifications for the user without any provision to go to details of that particular request notification **Interface aspect:**	



Notifications

1 New Notification

Request Accepted 2/2/20 1:00 PM
Daniel has accepted your request (ID 12323).

Older Notifications

Request Completed 1/1/20 5:00 PM
Andrew has marked your request ID 12322 as "Completed".

Time Estimation Changed 2/2/20 1:00 PM
Andrew has changed your estimation for request ID 12322...

Request Status: Active 12/31/19 1:00 PM
Andrew has set your request ID 12322 to ACTIVE.

Request Status: Paused 12/29/20 2:01 PM
Andrew has set your request ID 12322 to PAUSED.

Request Accepted 12/29/19 2:00 PM
Andrew has accepted your request

Heuristic:
User control and freedom(Navigation)
Recognition rather than recall

Explanation:
The notification list has no provision by which a user can dig into his request for which the particular notification has come. This can be the case when, say a user puts his request in the app on sunday or on a designated holiday and this request gets confirmation after few days being elapsed, or a user raises multiple requests at a time, then he might not get the exact idea about which of his request got approved or accepted.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

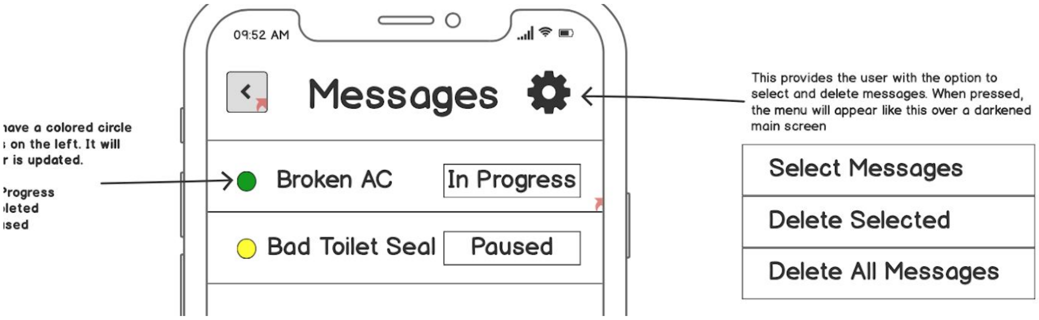
Justification (Frequency, Impact, Persistence):
Frequency: High
Impact: Medium
Persistence: Low
How I weighted the factors:
For the frequency, I think this is going to be high as no matter who the user is in terms of experience or what the request is, user is going to face this problem.
Impact is going to be medium as the users have to go back to their requests list to confirm which request of theirs has been accepted remembering that request id.
Persistence is going to be low as once the user finds the request that got approved then the user will not find this an issue.

Possible solution and/or trade-offs:
Possible solution to this would be to link each notification thread to respective request listing in the “Requests” page so the user immediately comes to know which of his requests got accepted. There seems to be no significant trade-off in usability design aspect.

Relationships: This UAR is not related to any other UAR.

UAR 06

No. **Abhimanyu - 06**	**Problem06**
Name: Settings logo not being used for typical settings functions.	
Evidence: This is a messages screen where users can chat regards to a particular request with a settings logo on the top. **Interface aspect:**	



ıave a colored circle
ı on the left. It will
r is updated.

ʼrogress
ıleted
ısed

This provides the user with the option to select and delete messages. When pressed, the menu will appear like this over a darkened main screen

Heuristic: Map between real world and system

Explanation:
For deletion of messages , a symbol of trash bin would be a better option in my view along with checkboxes against each message thread. Settings symbol might give users a different notion such as muting the messages for notification, setting alert for a particular message etc.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):
Frequency: Low
Impact: Low
Persistence: Low
How I weighted the factors:
I have rated the frequency as low because in my view a user might not be interested in deleting messages until he has any storage issue on his phone.
As for impact, I feel that this misnomer of sorts is misleading for the user but wouldn’t impact his ultimate goal of booking maintenance.
As for persistency, I feel once user sorts this out, this shouldn’t be a problem

Possible solution and/or trade-offs:
Possible solution would be to provide the same functionality with “Trash Bin” symbol which in my view would be more clear in the messaging along with check-box options against each message thread.
Possible trade-off would be that the contents already there have to be compressed a little to accomodate a checkbox which might make the reading a little difficult.

Relationships:
This UAR is not related to any other UAR.

UAR 07

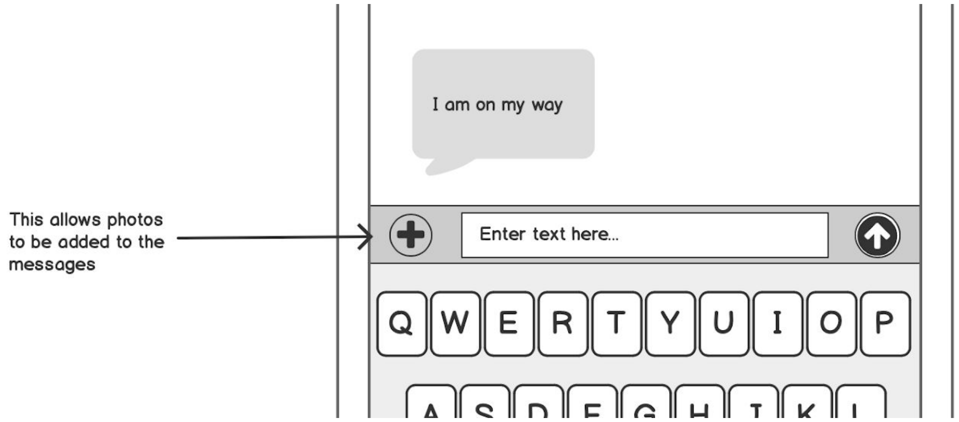
No. **Abhimanyu - 07**

Problem07

Name:
Plus sign to add photos might not be the very obvious signifier.

Evidence: This interface is where a user can chat with his service provider pertaining to a particular request.

Interface aspect:



Heuristic: <name of the heuristic that is violated / adhered>
Map between real world and system
Consistency and Standards

Explanation:
The “+” symbol to add a photo is not a very obvious signifier for the user. Instead a typical symbol for “Gallery” would be a lot more intuitive. The user might think of this “+” sign as adding people to this chat thread. Hence to make clear to the user that he can only add photos with this option , a “gallery” symbol would be more appropriate in my view.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):
Frequency: Medium
*Impact:*Low
Persistence: Low
How I weighted the factors:
I think in about 50% of the cases only where a user would be using this to upload a photo to better explain his situation. Hence marking frequency as moderate.
For the impact, user might be lost for a moment to figure out how to attach his photos but would be able to do so with little exploring and once he knows about this feature he would know his way around.

Possible solution and/or trade-offs:
Signifying with a Gallery symbol would solve this issue with no possible trade-off in my view.

Relationships:
This UAR is not related to any other UAR.

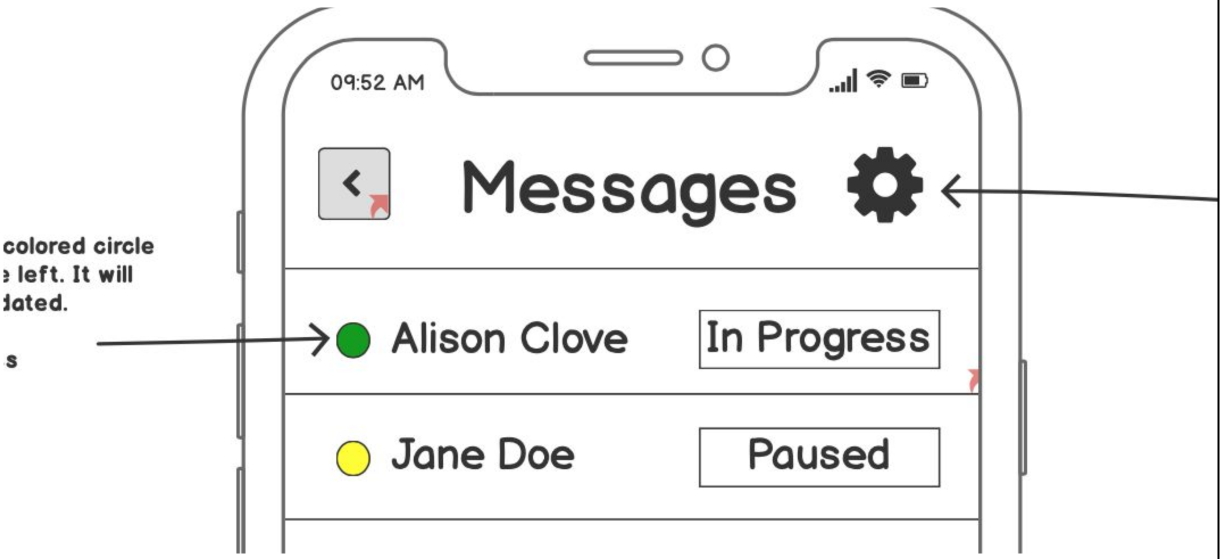
UAR 08

No. **Abhimanyu - 08**	**Problem08**

Name:
“Paused” system status is not self explanatory in the messages thread listings

Evidence: This is a messages screen, listing all the messages thread for each of the requests.

Interface aspect:



Heuristic:
System status

Explanation:
The “Paused” status for a particular message thread is not very obvious to me to figure out what is happening to my request and probably would have to message him asking what is happening to my request.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

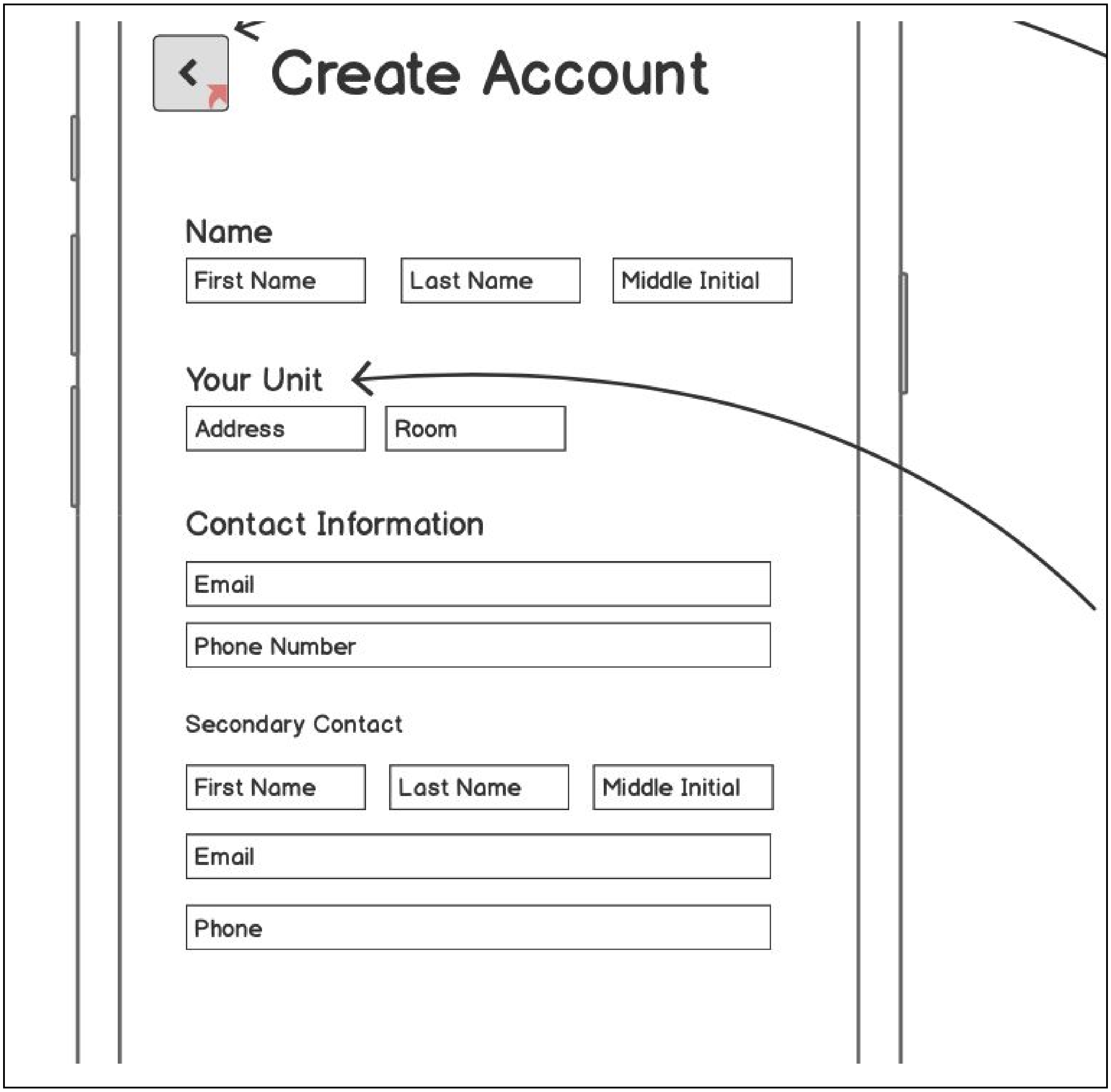
Justification (Frequency, Impact, Persistence):
Frequency: High
Impact: Low
Persistence: Low
How I weighted the factors:
I think a user would like to have a chat with his service provider for a particular service just to make it sure his service provider knows about his visit to their place. Hence I am assuming each time a request gets raised and accepted, this would be the case.And it would be confusing even for a moderate user to fathom what "paused" means in context to different services.
The impact would be low as this can be clarified by asking him over the text and persistence also then becomes low.

Possible solution and/or trade-offs:
The possible solution to this would be to come up with a more clear and self explanatory phrase or include a typical "i" symbol where users can click on it to have more clarification about it.
Trade-off would be that as more elements would have to be included, chances are that UI may look more cramped and bring in "minimalist design" heuristic in play.

Relationships:
This UAR is not related to any other UAR.

UAR 09

No. **Abhimanyu - 09**	**Problem09**
Name: Fields in the maintenance form and create account page are not marked if they are mandatory or not.	
Evidence: This is the maintenance form in the system that is to be filled by the user describing his needs/work required and in create account form also user needs to fill out his/her details. **Interface aspect:**	



Create Account

Name

First Name	Last Name	Middle Initial

Your Unit

Address	Room

Contact Information

Email

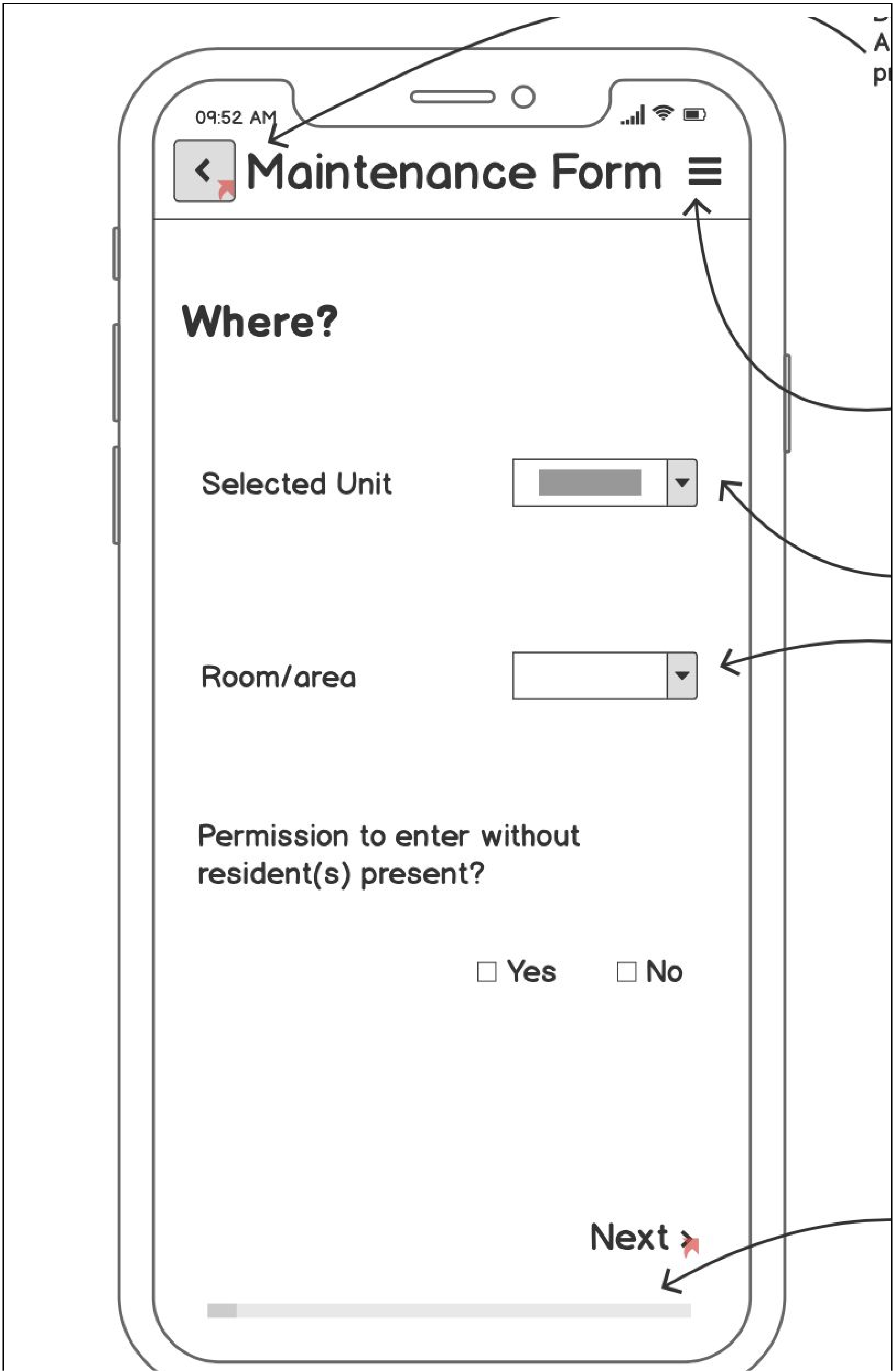
Phone Number

Secondary Contact

First Name	Last Name	Middle Initial

Email

Phone



Maintenance Form

Where?

Selected Unit

Room/area

Permission to enter without resident(s) present?

□ Yes □ No

Heuristic:
Error prevention

Explanation:
Since the fields are not marked as mandatory, the user might choose to skip one of the fields and proceed ahead, but if the system expects that particular field value then the user would get an error for that. Here a user might want to skip room/area when he wants a service for pesticides and proceed ahead but if system expects that value ,then user would get an error.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

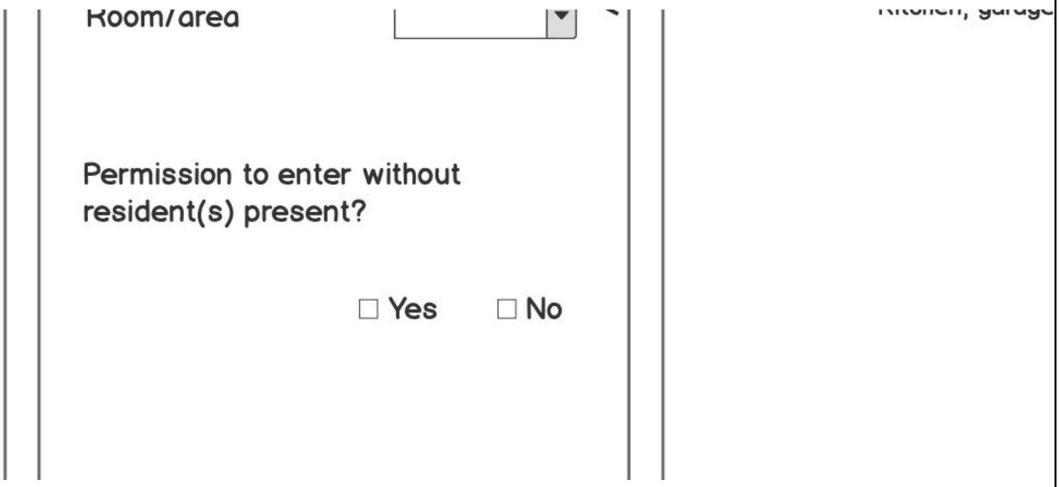
Justification (Frequency, Impact, Persistence):
Frequency: High
Impact: Medium
Persistence: High

How I weighted the factors:
The frequency would be high as this seems to be a mandatory step in completing the goal and one would not know which all fields are mandatory or not. The impact would be medium depending upon error resolution and recognition mechanism provided. With all the fields up for guessing , persistency also seems higher.

Possible solution and/or trade-offs:
The possible solution to it would be to star mark the mandatory fields and this would be without any trade-off.

Relationships:
This UAR is not related to any other UAR.

UAR 10

No. **Abhimanyu - 10**	**Problem10**
Name: Checkbox selection option for permission to enter the house or not the most ideal form.	
Evidence: This is the maintenance form screen in the application asking for permission to allow entry into the house. **Interface aspect:**  Room/area Kitchen, garage Permission to enter without resident(s) present? □ Yes □ No **Heuristic:** Error prevention	
Explanation: Since this question can have only one choice as an answer , this checkbox presentation of choices can let users select both which can induce a system error.	

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):
Frequency: Low
Impact: Low
Persistence: Low
How I weighted the factors:
Frequency of this problem is assumed to be low as I am assuming user themselves would not be choosing both the options, impact and persistency also will be low as even if user selects both , the system would prompt user to check only one choice and thus be solved.

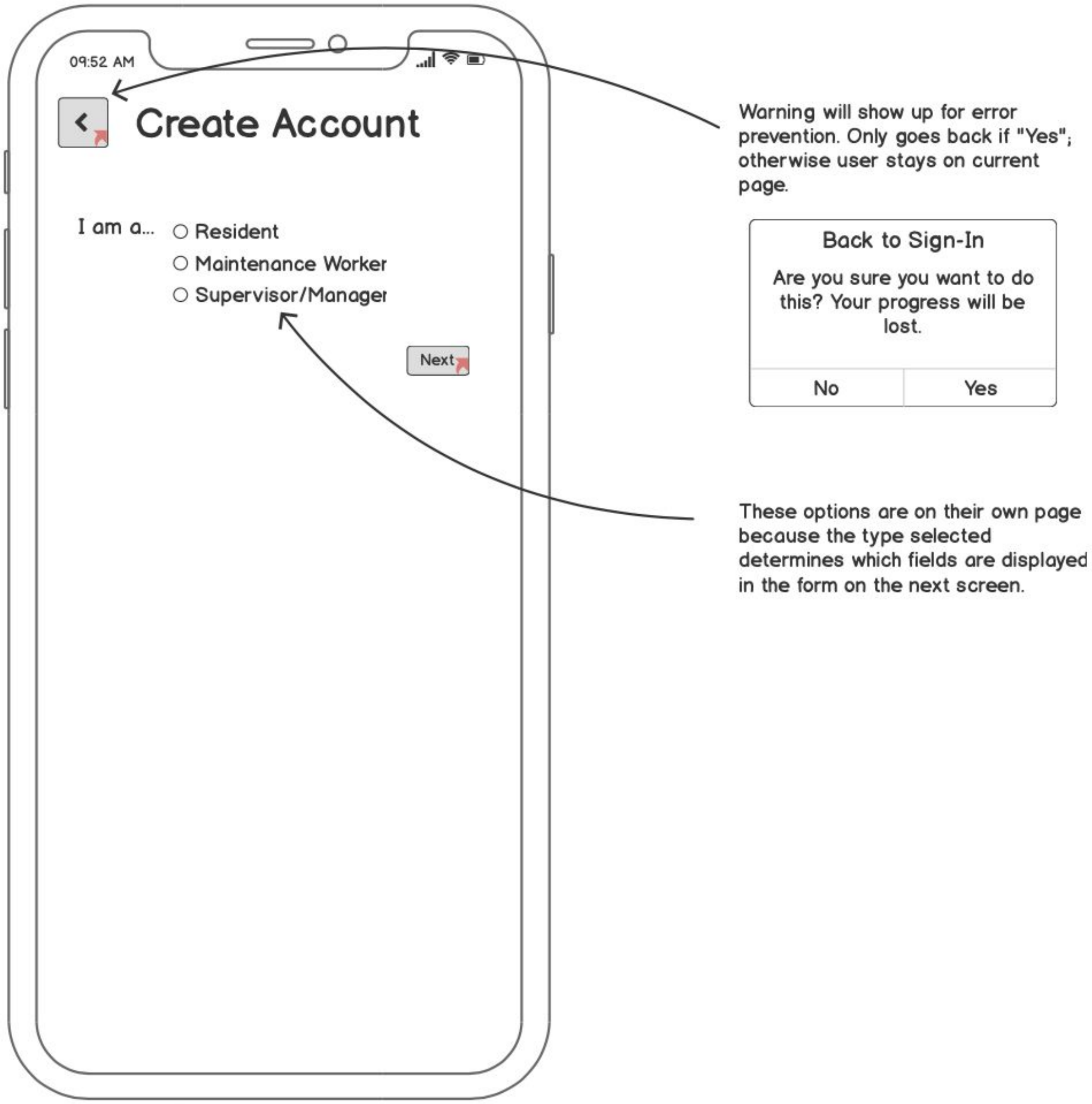
Possible solution and/or trade-offs:
Present the choices in radio button form and this would have no trade-off.

Relationships:
This UAR is not related to any other UAR.

UAR 11

No. **Abhimanyu - 11**	**Good Aspect** 1
Name: Customisation of account creation process	
Evidence: The following UI provides customisation while account creation. Depending upon which role the user intends to use the application, the application presents them with the account creation page asking for the info that is relevant only to that role.	

Interface aspect:



Warning will show up for error prevention. Only goes back if "Yes"; otherwise user stays on current page.

Back to Sign-In

Are you sure you want to do this? Your progress will be lost.

These options are on their own page because the type selected determines which fields are displayed in the form on the next screen.

Heuristic:
Aesthetic and Minimalist design

Explanation:

As different roles would have different ID requirements, presenting all such fields to the user would certainly overwhelm them and can make them averse to using the application.Hence cutting down on information feed to the user will certainly be a attractive feature for the user.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):
As different roles would have different ID requirements, presenting all such fields to the user would certainly overwhelm them and can make them averse to using the application.Hence cutting down on information feed to the user will certainly be a attractive feature for the user.

Possible solution and/or trade-offs:
N/A

Relationships:
This UAR is not related to any other UAR.

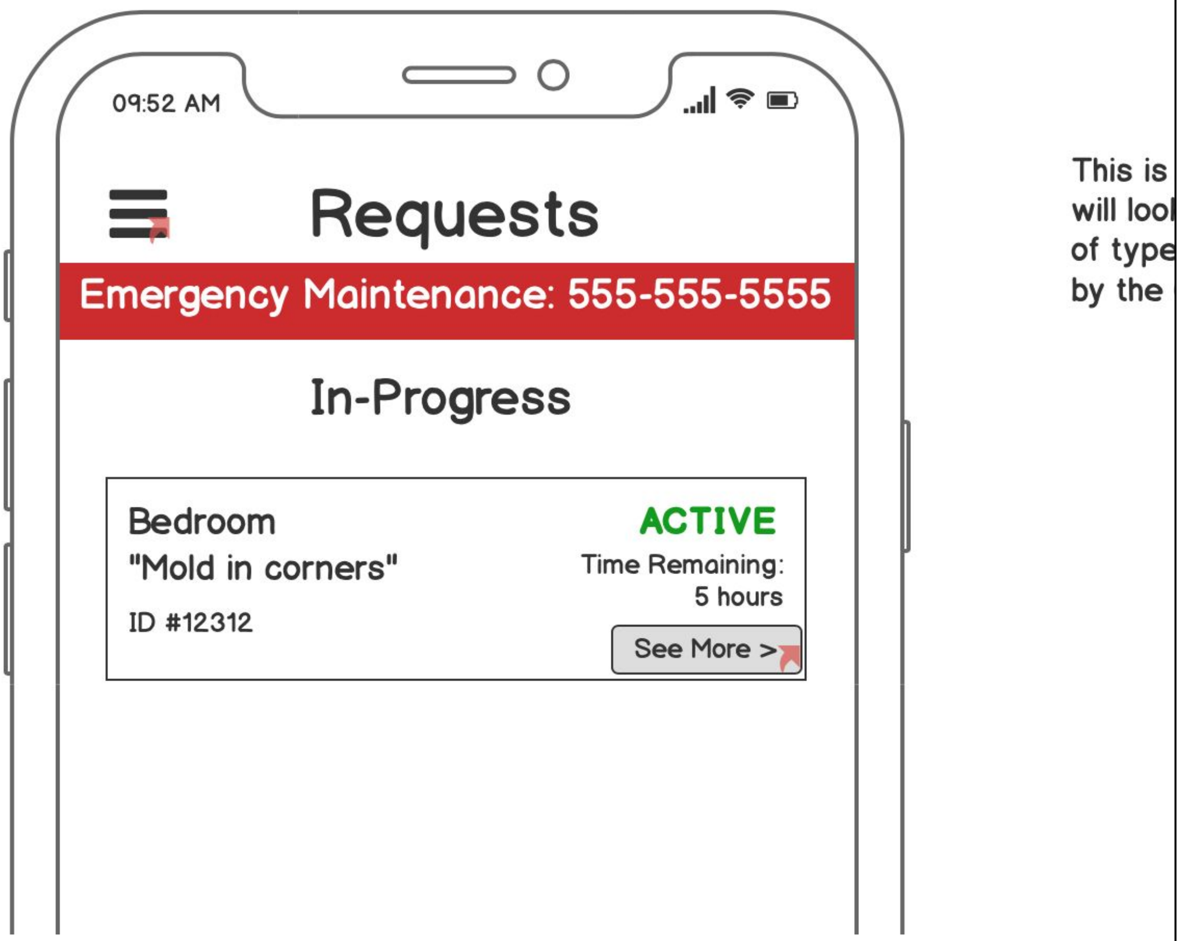
UAR 12

No. **Abhimanyu - 12**	**Good Aspect 2**

Name:
Flashing of Emergency Maintenance number

Evidence:
The following screen flashes emergency number in red

Interface aspect:



Heuristic:
Aesthetic and Minimalist design
Recognition rather than recall

Explanation:

The flashing of Emergency Maintenance number right at the top would certainly able the users to call the number right away for their emergency needs. This saves them from all the searching and navigating through the application to get this number.

Severity or Benefit:

Rating: <choose one of the following:>
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

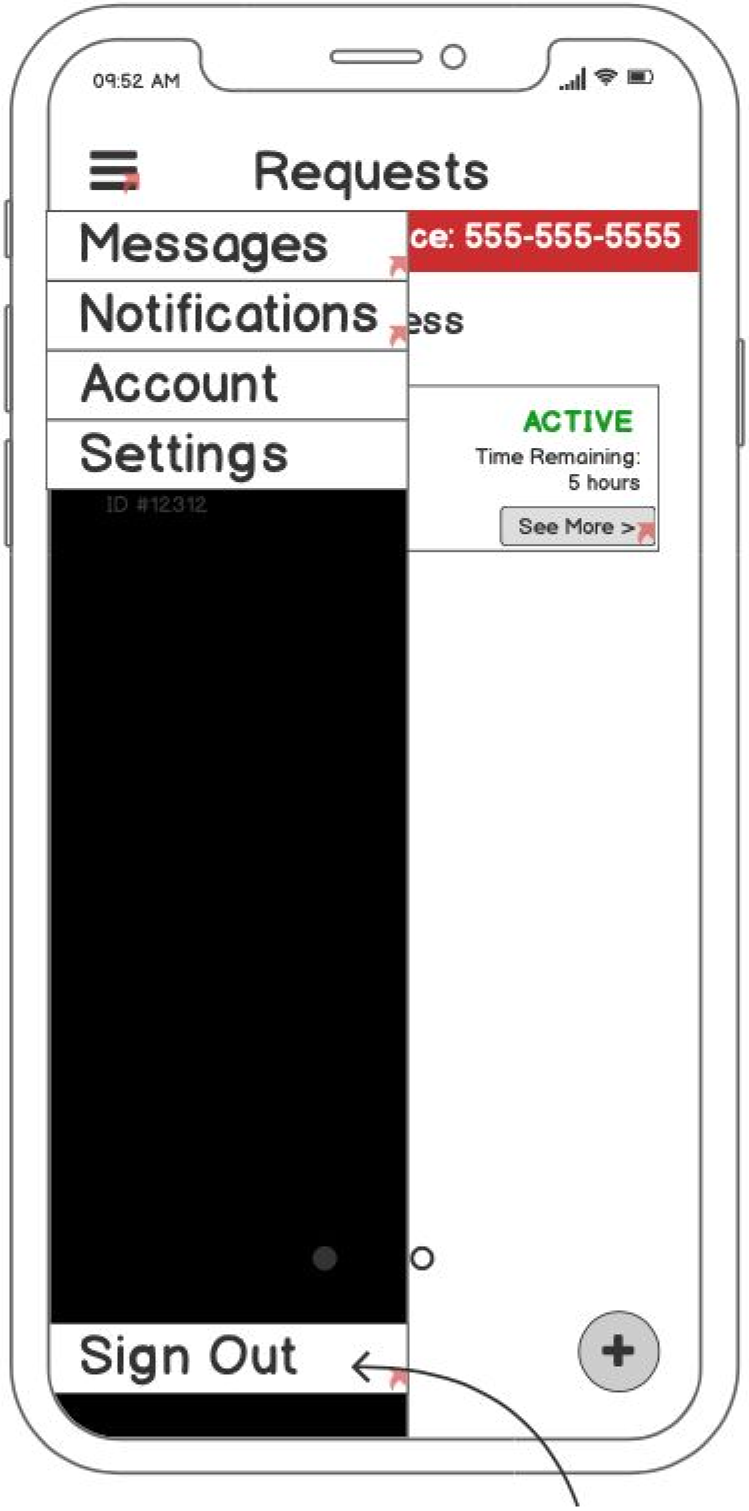
Justification (Frequency, Impact, Persistence):
The flashing of Emergency Maintenance number right at the top would certainly able the users to call the number right away for their emergency needs. This saves them from all the searching and navigating through the application to get this number.

Possible solution and/or trade-offs:
N/A

Relationships:
This UAR is not related to any other UAR.

UAR 13

No. **Abhimanyu - 13**	**Good Aspect 3**
Name: “Notifications”,“Messaging” features for the users	
Evidence: The slider screen on the click of the hamburger menu option presents users with notification and messaging features. **Interface aspect:**	



Heuristic:
Navigation and control
Standard and consistency.

Explanation:
With these features on the application the users can be kept updated on their requests real time and with the messaging feature users can sort out their queries and contingency situations directly with the service provider instead of doing through typical call centre offices.

Severity or Benefit:

Rating:
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

With these features on the application the users can remain updated on their requests real time and with the messaging feature users can sort out their queries and contingency situations directly with the service provider instead of doing through typical call centre offices.

Possible solution and/or trade-offs:
N/A

Relationships:
This UAR is not related to any other UAR.

Final Conclusion:

The problems were regards to ambiguous labels being presented to the user like "Time Remaining" which might not give user the exact idea what system wants to convey.Other issues such as isolated placing of "signout" option, using setting symbol to signify message deletion option, "+" sign to let users know that they can add pictures were also some minor issues found which if solved would certainly make the use lot intuitive.Also problems such as not marking fields as "mandatory" were also found, not symbolising the call option on the "emergency number flash", not providing a way to go directly to the the requests thread in my view were little more significant. Apart from this one more minor issue of providing option in the checkboxes form instead of radio buttons was also found.

To solve these problems few very basic and subtle changes/ additions were suggested.
For example to solve "Time Remaining" issue, it can be improved by simply rephrasing the label to make that more explicit/clear to the user.Others issues could be overcome by just replacing the existing "+" and "settings" symbol with more prevalent and typical symbol of "Gallery" and "Trash Bin" symbols. To solve problems pertaining to "emergency number flash"
 adding a phone call symbol was suggested, also linking notification to request on the "Requests" page was also suggested which would save users from having to remember the IDs of the request and then checking exactly which particular request got approved.Also marking fields in the "Maintenance Form" and "Account Creation" form as mandatory is suggested. Once these suggestions are acted upon,I guess the application would become a lot more intuitive and efficient for the users to use.

Reflection:

In total I could find 10 problems. There were couple of more problems that I didn't report as in my view they were not exactly usability problems but more of a personal viewpoint.These were
:

1) providing the messaging feature from the "requests details" page also
2) The "+" to make new requests is too isolated at the right bottom corner.

To evaluate and finalise the UARs it took me around 3.5 hrs and using the nielsen's heuristics I could categorise all my UARs.So Nielsen's heuristics proved to be a useful tool for me.
The UAR's in my view gave me a pretty comprehensive and solid framework to describe all my good and problem aspects and as to what can be improvements in doing this task all over again, to me this process seemed pretty comprehensive and encompassing.