

ToDo List: A Google Glass Application

Deep Ghosh
Georgia Tech
dghosh33@gatech.edu

Varsha Jagdale
Georgia Tech
varsha.jagdale@gatech.edu

Ashwini Khare
Georgia Tech
ashwini@gatech.edu

ABSTRACT

UPDATED—12 March 2015. This report is for the first project done towards Ubiquitous Computing course offered in Spring 2015 at Georgia Institute of Technology. The authors enhanced the existing ToDo list application for Google Glass by providing a quicker way to capture a task and a visual indication to rework on the task at a later point in time.

Author Keywords

Google Glass; ToDo List; Quick Capture

ACM Classification Keywords

Experimentation; Human Factors; Design

INTRODUCTION

A commonly observed tendency in case of ToDo tasks captured irrespective of the medium is them being very easily forgotten. “Out of sight, out of mind” principle is very relevant to the nature of ToDo tasks. The biggest problem while using any ToDo application is a long time to capture the task and lack of context-aware based reminders. Google Glass could be the most appropriate deployment platform as the user has constant & instant access to Glass. Also, the information can be constantly shown to the user literally in front of his/her eyes. This project illustrates an attempt to improve the task capture speed and addition of a constant reminder in a previously developed Google Glass ToDo application. In this text, Glass refers to Google Glass unless otherwise specified.

PREVIOUS WORK

A project [1] on similar lines was worked on in the previous offering of the same course. In their work, the authors start with studying the different types of ToDo tasks. Broadly, the tasks were classified as short term to-dos, reminders, long-term to-dos, saved information like account numbers and reference materials like links and references to external materials. The authors had discovered that time to capture is of great importance for a To-Do application. Most of the time, notes were taken in the quickest possible way on something that was at that time most handy. The notes didn’t seem to follow any structure and medium of note taking was served by the purpose of the note. For example, shopping lists are intentionally stuck on refrigerators. The

authors developed two prototypes, one being a purely web based flow and the other being an application running on Glass.

Their Glass application supported features like adding, viewing, updating, reordering and deleting ToDo tasks. The tasks could also be marked as done and undone. Finally, they conducted a user study and were able to conclude that in 9 out of 10 cases the users were comfortable with using Glass ToDo as a substitute for their conventional ToDo applications mainly because of the ubiquitous nature of Glass.

OUR WORK

The main functionalities that our team worked on for the first part of this project are improving the capture time, an option to rework on the ToDo task along with a visual indication and a constant reminder of the top most ToDo task in form of a live card.

Reducing the ToDo task capture time

We tried to employ the Glass Design principle of “Decrease the time from intent to action” in following two ways

a. Introduction of a new quick capture ToDo flow

Based on our observations, we found that the current flow of entering a ToDo task was an elaborate 8-step process “Add a task”(See Figure 1 in Appendix) We anticipate that this flow might be turn out to be a long process interrupting the natural flow of conversation that the Glass user might be having. We introduced a new flow called “ToDo” to quickly capture the ToDo task with just 3 steps (See Figure 2 in Appendix). The flow is

1. User is in conversation with some other person and needs to take a quick note
2. She/he tilts head or tap the track pad to wake up glass
3. She/he says “ok glass” to bring up menu
4. She/he says “ToDo” to add a quick ToDo task
5. She/he says the note content as briefly as possible.
6. She/he gets back to the conversation within fraction of seconds

In the background, Glass runs Google Speech-to-Text API to translate the text into speech and stores the newly added task in the Glass data storage.

• ACM copyright: ACM holds the copyright on the work.

b. Reordering the options present in the existing “Add a task” flow

In the existing “Add a task” flow there is an option for users to recapture the ToDo task by tapping for more options. The current order in which options are displayed to the user is *Save->Record->Cancel* (See Figure 3 in Appendix). We anticipate the users to use more options only when they might want to rework the ToDo. Therefore it makes sense to show the *Record* option before the *Save*. New order is *Record->Save->Cancel* (See Figure 4 in Appendix)

An option to rework on quickly captured ToDo tasks along with a visual indication

The ToDo tasks present in the task list that are captured using the new ToDo flow are distinguished from the rest in using a visual indicators (See Figure 5 and 6 in Appendix). This reminds the user to rework on these tasks whenever she/he goes through the task list next. The rework option appears under menu displayed on the detailed task screen. The quickly captured ToDo tasks that are marked done are removed from the quick capture pile. All the other interactions of the quickly captured ToDo tasks remain the same as normal tasks.

Metadata of the ToDo tasks

We anticipate that by providing contextual data like the place and time where the quick ToDo task was captured might help the user to remember the ToDo details. Thus enabling she/he to effectively re-work on the ToDo tasks. (See Figure 7 in Appendix)

Persistent display of the oldest task

We constantly want to remind the user of his/her most important ToDo task. That is why we created a live card that is constantly displayed as the first card the left of the home card. (See Figure 8 in Appendix) As of now we are displaying the top most task of the ToDo list on the live card.

DISCUSSION

We would like to run user studies to determine whether any of our above design decisions are effective. Some of the directions, which we are interested in exploring in the user studies, are follows

1. Is the new quick “ToDo” flow better and faster than the previous “Add a flow”?
2. How many users would religiously go back to rework on the quickly captured ToDo tasks?
3. What are the implications in case of the user not reworking on the quickly captured ToDo? Especially while accessing the ToDo task later.

4. Which mediums (For e.g. Desktops, Laptops and Mobile) are users most likely to use for reworking on the quick ToDo tasks?
5. In “Add a task” workflow, do the users tap for more options only to recapture the ToDo task?
6. Does the metadata (time and place) captured along with “ToDo” actually help the user remember facts and effectively work of the quick ToDo tasks?
7. Should the metadata be shown on the properly captured ToDo task? In this case, will it help the user to carry out the ToDo task effectively?
8. How many of ToDo tasks, which ones and with how much detail should be they shown on the live card?
9. Would the users like to be reminded about the tasks to be performed? If so, in what ways would they like to be reminded e.g. time, location or person?

Answers to these questions will help us derive the design decisions for the next iteration. For example, for facilitating external data entry should we create an Android app or a web interface? How should the database design look like?

Going by “Access drives capture” rule in context aware systems, if we are able to understand the likely ways in which the user might access the ToDo tasks, we can capture the context data accordingly.

FUTURE WORK

First and foremost, we need to conduct user studies to evaluate some / most of our above questions. Other possible directions in which we could take our work forward is:

Reducing the ToDo task capture time even more

Removing one more step from the quick “ToDo” flow by adding a “hot word”, listening to which the Glass wakes up and starts the add a to-do operation. One possible alternative to activate Glass is by using an external hardware trigger. Another alternative could be run an application as a service, which keeps on listening the voice inputs for the hot word and activating it accordingly.

Context aware access

The ToDo list could be made context aware by adding a component of location, time or people. Use-cases of location and time are fairly common and have been implemented before. For the person scenario, we assume both the user of this application and the person the task is being associated to, maintain their public calendars, a few jobs running in the background can determine whether if both of the persons will be in a common location. If so, the user can be prompted with the task associated.

CONCLUSION

In the first part of this project we tried to focus on finding various ways in which we could improve the task capture speed, an option to rework on the ToDo task along with a visual indication and a constant reminder of the top most ToDo task in form of a live card. We need to conduct user studies in order to verify if our anticipated ways actually work. Hopefully, the data from the user studies help us derive direction and design decisions for the next part of this project.

REFERENCES

1. Google Glass To-Do
<http://www.harshath.com/glass-tester/>
2. Visual Contextual Awareness in Wearable Computing. Thad Starner, Bernt Schiele, Alex Pentland
3. Wearable Computing and Contextual Awareness. Thad Starner
4. Towards a better understanding of Context and Context Awareness. Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, Pete Steggles
5. Ubiquitous location tracking for context-specific information delivery on construction sites. Amir H. Behzadan, Zeeshan Aziz, Chimay J. Anumba, Vineet R. Kamat
6. Everyday Wearable Computer Use: A Case Study of an Expert User. Kent Lyons
7. Practices for Capturing Short Important Thoughts. Gillian Hayes, Jeffery S. Pierce and Gregory D. Abowd
8. Information Scraps: How and Why Information Eludes Our Personal Information Management Tools. Michael Bernstein, Max Van Kleek, and David Karger