Novel Interaction: Voice-Based Kanban Software

Abdul-matin Adebayo
M00869707 | 2023

INTRODUCTION

A voice-based interface is a type of user interface that lets people use spoken commands to control a device. Voice-based interfaces can be used to support computer science students in many ways. One efficient way is to use a voice-based interface to help manage computer science students' tasks and projects (a voice-based project management software). Overall, voice-based interfaces have the potential to improve the learning experience for computer science students by providing a more approachable and natural method to interact with technology.

DESIGN CONCEPT

The voice-based project management software would be a Kanban board which would help students visualize tasks, allow group members to view the status of tasks, update tasks across the board, and rename tasks with voice commands. An example of a typical kanban board for project management is Trello.

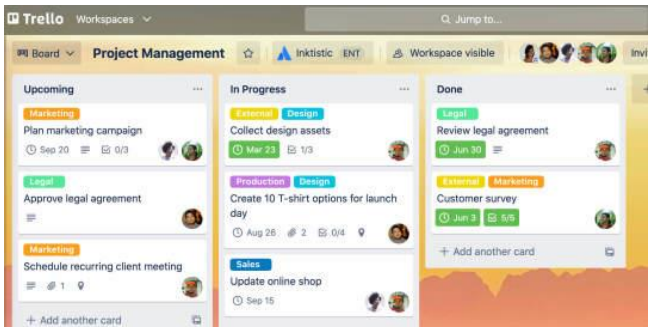

Figure 1. Design - Trello Project Management Board

Features and Command Phrases

Some of the major features and activating commands of the voice-based project management system expected by the system as input are listed below.

- **Updating a Task**:
  To start an update operation of a task, the following command phrases act as a trigger:
  - update, set, change, alter, modify, edit, correct, make.
- **Updating a Status of a Task**
  To update a status of a task, the update operation would have already been triggered, and updating the status would act as a follow-up event, the following phrases act as a trigger for updating a task:
  - inprogress, progress, pending, completed.
- **Valid Task ID**:
  In updating a task, a task ID would be used to identify a task, and a valid task ID acts as a trigger to identify a task. Examples to be used in this project are:
  - M22, M16, M19

The system would be able to engage in dynamic conversations listening for trigger commands in a particular stage of a conversation between a user and the system and the user does not need to restart every conversation with the system. This would be done by using stages for each conversation where a particular stage would be listening for a certain set of commands to activate a function. If in any case, the system encounters an error, the system would be able to resume its initial stage from a previous conversation and carry on with the user creating a more natural conversation between the user and the system.

Dialogue

Since the trigger words and dialogue are dynamic, below are possible statement structures the system can respond to in different stages of a dialogue.

Scenario 1:

Listening command from the user:

'{***updating a task command phrases***} task {***valid task ID***} to {***updating a status of a task command phrases***}'

Action:

A task with the specified task ID gets updated from its initial status to the specified status from the command.

System Response:

'task {**valid task ID**} has been updated to {**updating a status of a task command phrases**}'.

A more dynamic conversation and actions may be as follows.

Scenario 2:

Listening command from the user:

'{***updating a task command phrases***} task.'

System Response:

'what is the ID of the task to be updated'

Listening command from the user: '{***valid task ID***} task'

Action:

If the task ID is invalid, the system asks the user 'what is the ID of the task to be updated' and saves the stage of the conversation.

A user can then respond with either a valid task ID or ask the system where the task ID can be located with is determined against a 'question command phrase' which are:

- what, think, maybe, do not, tell

A sample listening command for this would be:

'I **do not** know what the ID of the task is', or '**Tell** me where the ID of the task is located'. In this case, the system says 'the ID is located on the top left section of the cards'. The system then returns to the previous stage of this conversation which again asks the user 'what is the ID of the task to be updated.'

System Response:

'what should the task be updated to?'

Listening command from the user:

'{***updating a status of a task command phrases***} task'

Action:

A task with the specified task ID gets updated from its initial status to the specified status from the command.

System Response:

'task {***valid task ID***} has been updated to {***updating a status of a task command phrases***}'

Scenario 3:

Updating a title of a task

Listening command from the user:

'{***updating a task command phrases***} task {***header, title, head, label, subject, info***} of {***valid task ID***} to {***value of new title to assign to task***}'

Action:

The title of the specified task ID gets updated to the newly given title input.

Some other conversational scenarios to engage users are scenarios where a user appreciates the system after a task has been performed or calls out to the system like a typical 'a Siri' command. A conversation for this scenario may be as follows:

Listening command from the user:

'{**thank you, thanks, nice, appreciate**} for the support.'

System Response: 'You are welcome'

PROTOTYPE

The developed prototype is targeted toward computer science students and assists them in their day-to-day activities. I would be developing a voice-based project workflow management to allow for easy access to manage and control a project workflow.

Some features implemented in the prototype are.

- Updating the status of a task across the Kanban board using voice commands
- Updating a title of a task using voice commands
- Conversations with the application

Implementation

In developing the application, I used the Web speech API which is a JavaScript framework for building voice-based interfaces on the web.

The project is a web-based solution and can be accessed from the link below:

https://github.com/matinayo/voice-based-kanban-board

The web interface has a play audio button to give the user an option to start talking. On click of the start button, an audio listening button appears indicating the system is waiting for a voice command from the user and when receiving feedback from the system, an icon indicating the system is talking is displayed, enabling the user to identify the different states of the application.

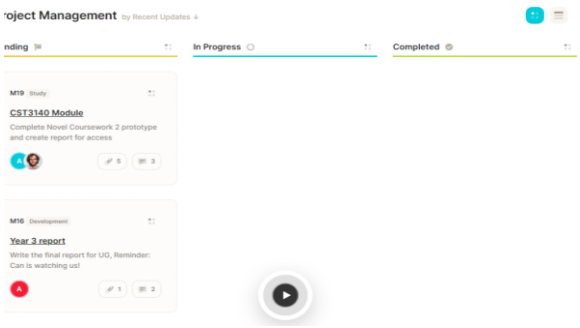

Figure 2. Prototype – A play button giving the user an option to start talking.

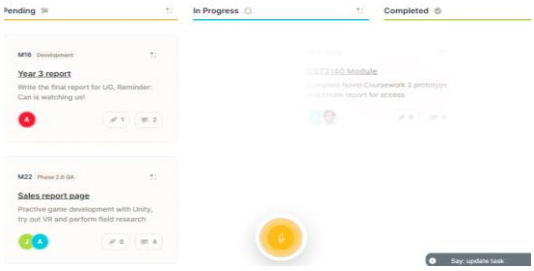

Figure 3. Prototype – A speaker button indicating the system is listening for a user voice command.

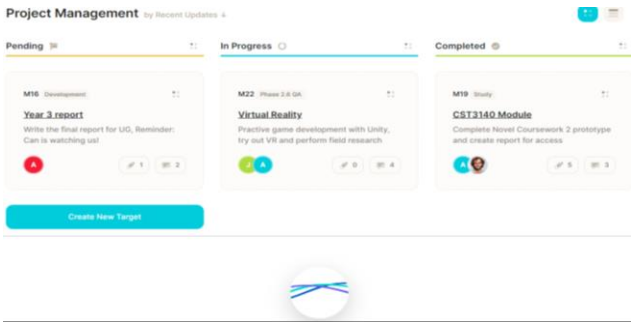

Figure 4. Prototype – An icon indicating the system is giving feedback to the user with updated tasks.

From the images above, after a user specifies a command to the system from Figure 3, the system updates the task and gives feedback on the command as shown in Figure 4.

EVALUATION

AIM

An evaluation was conducted upon completion of the prototype. The purpose of the evaluation was to determine if the project prototype meets the design concepts, if it serves the purpose of assisting computer science students in managing their tasks, and to identify any area that might have any flaws or identify areas that could be improved on to better make the system more reliable.

Method / Process of Evaluation

The type of evaluation used is the Formative Evaluation. This type of evaluation is conducted during the development phase (the prototype) to identify any problems and make improvements before a final product is released.

The method of evaluation used is Observation which involved watching how my target audience interacts with the project in a dynamic setting. The type of data collected is Qualitative data, using a list of tasks for users to follow during observation and was used to gain an understanding of user's perceptions of the application.

Data

Below is a tabular format of the evaluation process with questions, the table outlines the experience of two different users using the application.

SN	Task	Observation
1	Update a status of a task	User found it easy to update the status of the task without any intervention
2	Change a title of a task	User used a speech pattern that was different from what the system was expecting. The user said just the title name instead of 'change title to'
3	Are you able to identify the ID of the task to update?	The user was able to easily identify the ID of a task
4	Action user after completing a task	User said thank you and system responded
SN	Task	Observation
1	Update a status of a task	User said 'complete' instead of a valid task status which is 'completed', and system was not able to respond
2	Change a title of a task	User was not able to follow up with system during a dialogue because prompt for identifying a task was not clear
3	Action after completing a task	No other action from user

Table 1 & 2. Evaluation – Observations obtained from users carrying out tasks.

Discovery

During the evaluation process, some problems identified are:

- When users try to specify an ID of a task to be updated, the prompts and instructions were confusing and unclear which lead to difficulty in completing tasks listed in the evaluation process.
- Another interesting discovery is that users may have different accents or speech patterns that may be difficult for the system to understand and in some cases, were not recognizable by the system.

Implications

The discoveries made during the evaluation process may have implications for the project going forward. Some of the possible solutions that can be implemented to better achieve a better, more interactive, and efficient solution are:

- Asking clearer and more specific questions to users before they perform an action.
- Display possible valid commands that can be used in different stages in a conversation across the application to enable the system to carry out specific actions and give the user more knowledge in using the application.
- Enabling more dynamic patterns during command recognition to better understand different speech patterns from users.
- Continuous conduction of system evaluation, to identify and accommodate more keywords that might be used in a particular stage of the application, allowing the system to understand a wider range of conversation and identify more commands.