Experiment notes

Participant 1

First: Album playlist, no help

Second: Sleep timer, with help

First task: Starts with looking through the open program to find where the button should be visually. Then goes to the editor to start looking for where it is implemented. Identifies playlist component. Searches the repository in editor for “add album to queue”. Identifies a place where the text is written “add album to queue” and notes it (it’s the test folder). Notes some JSON files as well. Tests the other buttons in the area. Is generally overwhelmed by looking through the code.

File path: 1

Code: 1

Second task: Immediately asks an elaborate question about how to fully implement the feature. Blindly trusts the answer and essentially copy pastes the answer. Does not really look in the editor or at the program, only reads through the SPAC-B answer.

File path: 2

Code: 2

Participant 2

First: Sleep timer, no help

Second: Album playlist, with help

First task: Starts looking in the application for settings. Identifies the place in the GUI/application and opens the editor to identify where the settings is in the editor. Expresses confusion with the structure. Searches in the IDE for text from the application. Starts clicking around the different folders to get an overview of the structure. Is uncertain about the structure. Goes back to searching keywords from the application in the code. Tries to google “snap”. Expresses a desire to know where the inputs from the previously implemented Settings are handled. Has found a test file, and searches through it. Goes back to googling. Has identified elements that P2 thinks are the settings implementation from the text file and goes from there. Spends a lot of time in the test file and goes through it thinking participant should use code from it. Goes to the index.ts instead. Finds a settings.ts. Googles reducers. Participant is uncertain if participant should spend too much time googling with the limited time. Identifies settings actions from the settings.ts container.

File path: 1

Code: 1

Second task: Starts by copypasting the task description into the SPAC-B. Reads through the somewhat lengthy response. States that it’s already a help. Asks SPAC-B a follow-up question. From the SPAC-B response participant goes to the place in the code where other buttons are

implemented and tries adding one. Tries to implement stuff in the code. Has noted files by the 12 minute mark. Starts writing elaborate solutions. States that it is a lot easier with the tool than without. States that the participant hasn't been that critical towards the logic and should maybe double check if that's appropriate. Becomes uncertain with the SPAC-Bs validity. Does not have time in the end to write logical solution and therefore copy pastes SPAC-B's answer. States that the participant would implement stuff from there coupled with functions already present in the code.

File path: 5

Code: 4

Participant 3

First: Album playlist, with help

Second: Sleep timer, no help

First task: participant tries to ask about the structure of the project. Reads the summary from SPAC-B and understands some of it. Finds the Readme and reads it. Prompts for an overview of each of the packages. Goes to application to see if the button already exists, finds it does not. Asks SPAC-B for where the code is for adding to a playlist. Gets TrackPopUpContainer, finds it in the editor. Identifies a method that adds a track and notes it. Asks where albums are implemented to SPAC-B. Gets an answer that mentions both the data structure, which participant was looking for, and the GUI part, which participant was positively surprised with. Writes down notes for implementation. Asks for the triple dot button implementation. Is impressed that SPAC-B finds it. Finds the place with buttons in the editor. Tries to ask SPAC-B directly for an implementation plan, finds the plan helpful but somewhat general. States that participant feels the task would be easy had participant been proficient with React. Rushes writing solution last minute.

File path: 4

Code: 3

Second task: Start by writing a plan. Looks in the code for a place where to pause. Searches initially, but doesn't find it. Finds playerControls by looking through folders. Finds something called togglePlay, and assumes that is the correct method. States that participant feels limited by VSCode as participant is used to Writer. After searching for "Settings" and browsing through, participant identifies components/settings/index. Finally finds the place to search properly through files. Spends most of the time browsing around through the files. Repeatedly states that participant does not like web development. Starts googling ways to implement what participant wants to do. Starts inputting solution in better time.

File path: 1

Code: 2

It takes a month to onboard at the company participant works. Works as a lead and is responsible for onboarding, and participant was thinking that something like this could be helpful in their onboarding process of new employees.

Participant 4

First: Sleep timer, with help

Second: Album playlist, no help

First task: Identifies the place in the GUI where the function should be implemented. Asks where the “Autoradio craziness” is implemented. Asks specifically where the misleading test file is and finds it in the editor. Searches in the editor for autoradio craziness and finds the test file again. Notes the Test file as the place where the GUI is. Asks Spac-B for the logic to stop music. Looks for SystemMediaEvent in the code. Identifies the language files correctly. Identifies the containers as the place to add a new container called sleepTimer. Bases logic on findings from autoradio.js. Asks for the code that handles the button to start the music. SPAC-B gain returns a test answer. Continues filling in logic. Identifies the actual place holding the autoradio, as participant realizes the “autoradio craziness” is called autoradio-craziness in the code.

File path: 3

Code:3

Second task: searches for “Add album to queue” in the editor. Searches for queue. Finds the AlbumView rather quickly through this. Identifies the two popupbuttons that are in the popup by searching “queue” in this file. Searches for the addAlbumToDownloads method in the editor. Identifies AlbumViewContainer correctly from this as the place for logic. Starts looking for a place with the logic for adding songs to a playlist. Initially searches for Playlist and moves onto addToPlaylist. Identifies the method from playQueue that lets a user add these to a playlist and wants to import it. Keeps looking through the albumview/index and the queue menu to see how to combine these logics.

File path: 5

Code: 3

Participant 5

First: Album playlist, no help

Second: Sleep timer, with help

First task: Goes into the editor and finds the albumView component by the name. Goes to the program and looks for how an album page looks. Clicks the ... button and looks at the buttons in it. Reads through the code of the AlbumView component and finds the PopupButtons for the two buttons. Reads through the file names and find playlist.ts in actions and look through the code. Goes to queue.ts afterwards and reads through. Clicks through different folders. Tries to understand where the logic is that is triggered from the UI components. Returns to the actions files and look through them. Searches for the method name “addAlbumToQueue” in the editor and finds it in hooks.ts in albumviewcontainer. Starts inputting answers. States participant is thrown off by addToPlaylist being in queue, as participant is looking in the queue file, where this feature also exists.

File path: 5

Code: 3

Second task: Identifies the settings tab in the application. Asks an elaborate question for identifying the files that should be changed for implementing the feature. Gets an elaborate answer back. Starts looking through the files mentioned in the answer. States that participant has to figure out where the state management should be handled. Is uncertain if the response is correct. After reading through the files that were mentioned by SPAC-B and looking through the GUI participant settles on 2 files the participant deems the correct ones to implement the solution in. In the last minute participant queries SPAC-B again and gets an even more elaborate and potentially more helpful answer, but does not have time to use the information.

File path: 3

Code: 3

Participant 6

First: Sleep timer, no help

Second: Album playlist, with help

First task: Goes into the IDE and looks through the files. Goes to the application and looks for where to timer should be implemented in the GUI. Gets thrown off that participant cannot start the debugger. In the settings tries to clock on choosing repository. Changes language to Danish. Goes to editor and searches AutoRadio. Searches for AutoRadio again and looks through results. Searches for shuffle. Searches for .ts files containing auto. Finds a settings file, which is not the right one and then looks through autoradio.js. Keeps trying to find out where the surrounding methods are called. Identifies the language files, and states that this is where the "turn on sleeper" texts should be added. Lands on adding a separate container as sleeptimercontainer.

File path: 2

Code:1

Second task: Starts by asking SPAC-B where the playlist is implemented. States that SPAC-B has already provided the participant with more answers than participant gave in the previous task. Asks where the popup with add album to queue is implemented to SPAC-B. Searches and finds queuePopUpButton in IDE. In the application participant identifies a playlist and mistakes it for an album as the interface of it is similar, and starts ask for where the button is implemented. Searches for the files in the IDE that SPAC-B respond with. Reads SPAC-B answers and try to cook up answer from this. Realizes that participant has been looking for the wrong button in the application, as participant was on a playlist instead of an album.

File path: 1

Code: 1

Participant 7

First: Album playlist, with help

Second: Sleep timer, no help

First task: Goes into Nuclear and searches for a band, seemingly having trouble finding the album view. Asks SPAC-B where in the project add album to queue menu is implemented and

waits for the response. The answer describes the file path, popup buttons and more useful information. Asks SPAC-B to show all files which contains logic to add album to queue function. Asks where in the project add to playlist menu is implemented, and all files related to this function. Asks if SPAC-B can update the code about add to playlist functionality. Starts searching in VS code for the relevant code. Searches for utils but does not find the file the participant was looking for. Looks around in packages/main/src/utils. Tries to add a new file called trackUtils.st in this folder. Copy pastes a copy snippet for coneAndPrepareTrack into this file. Changes the name to trackUtils.ts. We inform that the participant does not need to make any changes to the code. Starts writing answers in the form. Copy pastes parts of SPAC-B responses into the form as time is running out.

File path: 1

Code: 1

Second task: Looks for a sleep timer in the code but cannot find one as it does not exist yet. Starts going through some selected files. Finds SoundContainer/autoradio.js and goes through the code. Starts filling in the form with answers but does not provide any reasoning. Finds SettingsContainer/index.tsx. Continues to look through various files. Finds app/app/actions/settings.ts. The participant writes some answers in the form. Participant finishes with 6 minutes left and does not think participant will be able to add anything else to the solution.

File path: 1

Code: 1

Participant 8

First: Sleep timer, with help

Second: Album playlist, no help

First task: Asks if the app contains both front end and back end. Asks SPAC-B where in the nuclear app the backend should be implemented for a sleep timer. Receives an answer from SPAC-B that wants the participant to inject it into the services. States that participant thinks the first part of the answer is misleading, as participant looks through the contents. Searches in the IDE for playercontroller, which is also mentioned in the SPAC-B respond and thinks this is more appropriate. Identifies some config files from the SPAC-B answer as well. Identifies player.ts as the place where participant would implement frontend, also from information from spac-b. Tries some new queries to SPAC-B and seemingly finds information that alters participant's thoughts about the answers participant has given so far. Gives some of the most elaborate answers so far. Largely bases participant's answers on SPAC-B responses, but goes into the IDE and locate most files before inputting them as answers. After inputting all answers and having a bit more time, participant queries SPAC-B again, trying to verify participant's solution. From the SPAC-B respond participant seems happy with the solution. Starts looking into the IDE, and seem a little less confident. Finishes the task without ever looking into SPAC-B

File path: 1

Code: 1

Second task: searches for the button name in the IDE. Seemingly has a better overview of the program from start after using SPAC-B, i.e. quickly looks for components. Identifies the popup button location quickly by searching in the IDE. Searches for playlist in IDE, finds playlist container and notes this file. Identifies some action files as well by conducting the IDE searches. Is confused about the architecture of the program, as participant notes bot REST api and then onto "something else" . Continues browsing the folders and searching for files, seemingly comes up with very elaborate solutions for implementing the issue. Looks through REST, API, and many other backend files. Has an epiphany as participant realizes that the app is not implemented with user login and therefore the back end isn't as important, states that much of what participant has done in the previous task must therefore be incorrect, which annoys the participant. Continues with outlining participant's plan but with less focus on the backend. Continues to discuss participant's previous implementation of the timer, which participant states was missing the logic of actually pausing the music. Starts discussing that this task is actually complex as you have to choose which playlist to add to and potentially add a new

File path: 2

Code: 2

Participant 9

First: Album playlist, with help

Second: Sleep timer, no help

First task: Goes into nuclear and finds an album and finds where the button needs to be added/updated. Goes into the IDE and looks through components. Rather quickly identifies AlbumView component. Asks SPAC-B where an album overview feature can be found in the files. SPAC-B explains the two index files in AlbumView component and AlbumViewContainer. Asks two new questions regarding the Feature location to SPAC-B and seems satisfied with the answers (they are also correct). Asks SPAC-B where similar logic is implemented. Finds AlbumViewContainer from this. States that participant thinks that a popup button should be added to the AlbumView, which calls a method in AlbumViewContianer. Asks SPAC-B for where playlist actions are and finds playlist.ts from this. From being pointed to onAddToPlaylist, which is called from elsewhere, participant identifies that the correct playlist action is probably "update playlist". States that he's trying to figure out how the onAddToPlaylist works. Asks SPAC-B how AddToPlaylist can be called with multiple tracks. As He's asking regarding the wrong method, he's not impressed with the response.

File path: 5

Code: 5

Second task: Goes into the application and identifies the Settings panel. Tries to identify the location where the it should be implemented. Goes into the IDE and looks for settings by searching. Looks in the components/settings/index for the appropriate settings, but is not convinced this is the correct location. From settings container identifies settings constants. Searches anew and identifies src/settings/index.ts and is confident this is the correct place. Tries to write what participant wants in the index file. Is confident that the GUI is implemented from this, and goes to searching for player files. States that now it would have been nice to have SPAC-B, as participant is unsure how to couple the functionality. Wonders if there should simply

be a timer that starts when the timer is toggled. Starts inputting a solution that participant is not entirely confident in.

File path: 2

Code: 2

Participant 10

First: Sleep timer, no help

Second: Album playlist, with help

First task: Browses the file structure briefly to see which programming language he's working with. Opens Nuclear and finds the settings view where the sleep timer needs to be implemented. Looks around for the search functionality in VS code. Searches for playback and settings to find relevant files but seems overwhelmed with the results. Starts browsing the file structure again. Goes through app/app/app.js file. Mentions that it's a very big repository and goes back to the search functionality, looking for playback again. Expresses a need to understand more about the code base while browsing around. Finds app/app/components/settings/index.tsx and reads through the file. Compares it to how it appears in Nuclear. Continues to browse around, wondering if participant is supposed to look for a component or a container. Finds app/app/actions/settings.ts. Finds the other way to search for files in VS code, and searches for settings. Searches for shuffle as shuffle songs is an option in Nuclear settings. Feels overwhelmed with the number of files in the software project. Looks through app/app/components/playoptionscontrol/index.js. Starts filling the form but is not very confident.

File path: 1

Code: 1

Second task: Looks around in Nuclear for an album and finds it eventually. Asks SPAC-B where is the option menu on the song implemented? Realizes it's not a very good question. Asks how do I add all songs from an albim to a playlist? Gets a decent answer but seems to be missing file paths. Asks where in the codebase are tracks added to a playlist? Looks through the album in Nuclear, considering what code is reusable from adding single tracks to an album. Looks through the answer from SPAC-B. Asks if participant can add multiple tracks to a playlist at once. Where is the UI component for album? Answer contains at least one of the relevant files, albumview which participant browses to in VS code. Looks through all the SPAC-B answers again. Asks which files do I need to modify to add the functionality of adding all albums to a playlist as an option for an album. Gets a pretty elaborate answer. Asks where is the menu option located? Searches in the albumview/index.tsx file for download and queue and finds both as PopupButtons. Starts inputting an answer in the form. Finds app/app/containers/albumviewcontainer/hooks.ts. Asks where are tracks added to albums in the codebase? Gets a great answer but does not have time to follow through. Hurries to fill out the form.

File path: 3

Code: 3