

Part 1

A. Primary persona

Martin Clarke, 22

Martin is an astrophysics student in his final year of university in the UK. He is not an organized person by nature so he is looking for an app that makes it easy and quick to manage his tasks. He uses his current calendar app every day because he likes to review and plan his day every morning. He likes to update his calendar from his laptop while he drinks his morning tea at his bedroom desk. His schedule is packed with labs, lectures, and he spends his evenings working on his dissertation. He spends most of his time looking at his university calendar and finds it to be a hassle to switch to his personal calendar app on the occasions when he needs it.

He runs an anime club at his local leisure centre and needs to keep track of weekly activities he plans for it. He finds it difficult to decide whether to use his personal or his university calendar app for it, since it is subject that requires a similar level of commitment as university but is not as serious. He does not want to use a third calendar app to contain his club schedule because that would make staying organised too difficult and inefficient.

On the weekends he likes to backpack around the UK and France with his boyfriend and he uses a separate personal calendar app to plan their journeys, and notes down relevant train schedules, museum opening times, and restaurant bookings. He would like to add notes under some tasks to make it easier to change or decide on plans if needed. For example, a screenshot of the train schedule in case they miss their train and need to quickly know when the next one is coming.

Since he is a student and often stays up late to work on his coursework, he values applications that have dark mode to ease eye strain. He noticed that a lot of applications don't have dark mode and that upsets him. His perfect calendar app would have dark mode as its default colour scheme.

Martin is also bored of seeing the same muted colours in most major web applications and would like to use an app that has bright accent colours to make the experience of using the calendar more fun.

Goals:

- To see his personal, club, and university timetables in one place to never forget any task, and to compare them to make sure he isn't double-booked
- To be able to filter tasks based on which category they belong to, like university, travel, or club responsibilities to easily switch between each type
- To set repeating daily, weekly, and monthly tasks to save time when planning out his week
- To add notes to each task, in text or picture format, or both
- To see the tasks by day, week, and month

B.

Primary Use Case: Add A Task

Description:

Customer uses the calendar web-application to add a task to their calendar based on its category. They can add a new category during the task addition if the right category doesn't already exist, and choose to add optional information to the task.

Start Condition:

The customer has opened the calendar browser window and logged in into their calendar account.

Basic Flow:

1. The calendar opens at the homepage showing the timetable for the month with tasks across every category, a setting icon, a category sidebar, a timetable view sidebar, and a plus sign button
2. Select to add a new task
3. Choose the category (default existing categories are work, academics, personal, and other)
4. Name the task
5. Enter the start and end date and time of the task
6. Confirm task addition
7. Return to homepage to view the task you added in the calendar

Alternative flow:

- A1: add new category
- A2: proceed to optional fields (select if recurring daily/weekly/monthly, task location, add notes)

Secondary Use Case: View Task

Description:

Customer uses the calendar web-application to view a task in detail on any day on the current screen. They can choose to view tasks of a certain category and to select from a variety of calendar scopes to find their tasks of interest easier.

Start Condition

The customer has opened the calendar browser window and logged in into their calendar account.

Basic Flow:

1. The calendar opens at the homepage showing the timetable for the month with tasks across every category, a setting icon, a category sidebar, a timetable view sidebar, and a plus sign button
2. Click on a task in the calendar
3. View the task page with task details
4. Return to the calendar screen

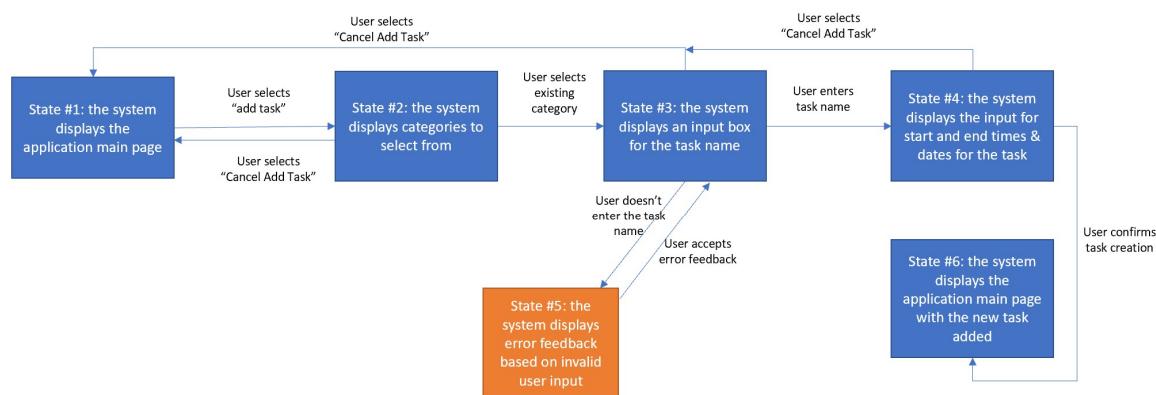
Alternative Flow:

- A1. Filter tasks by category
- A2. Change calendar scope

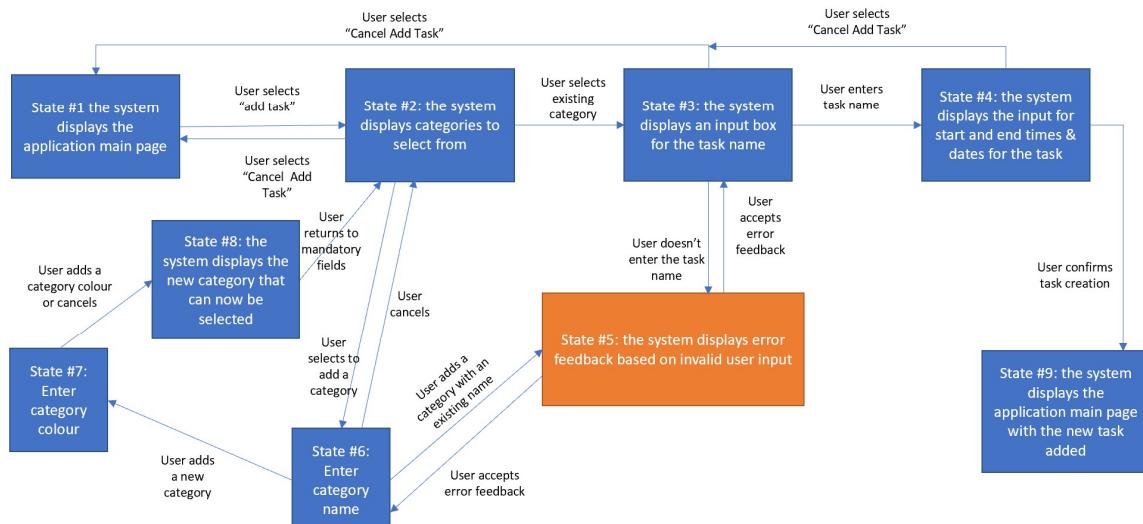
C.

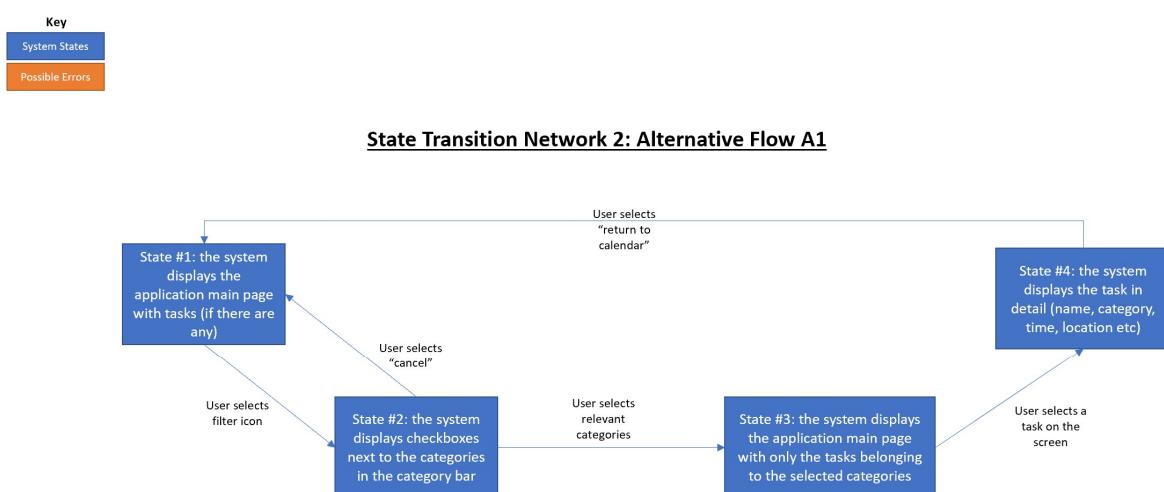
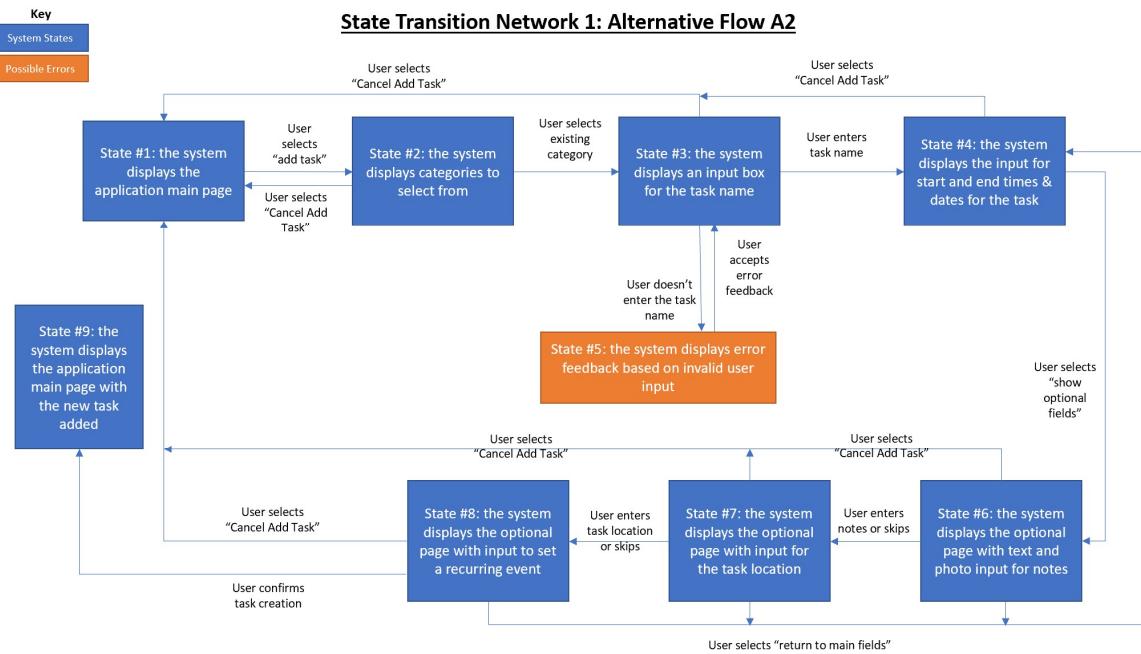


State Transition Network 1: Basic Flow



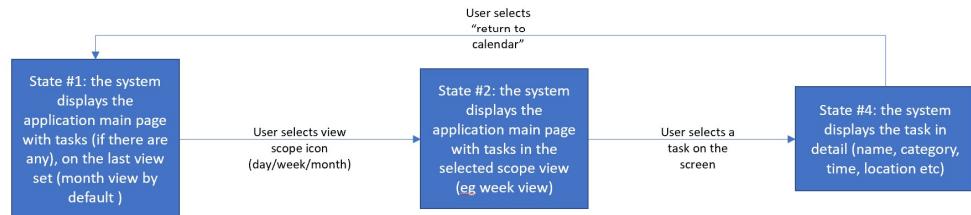
State Transition Network 1: Alternative Flow A1







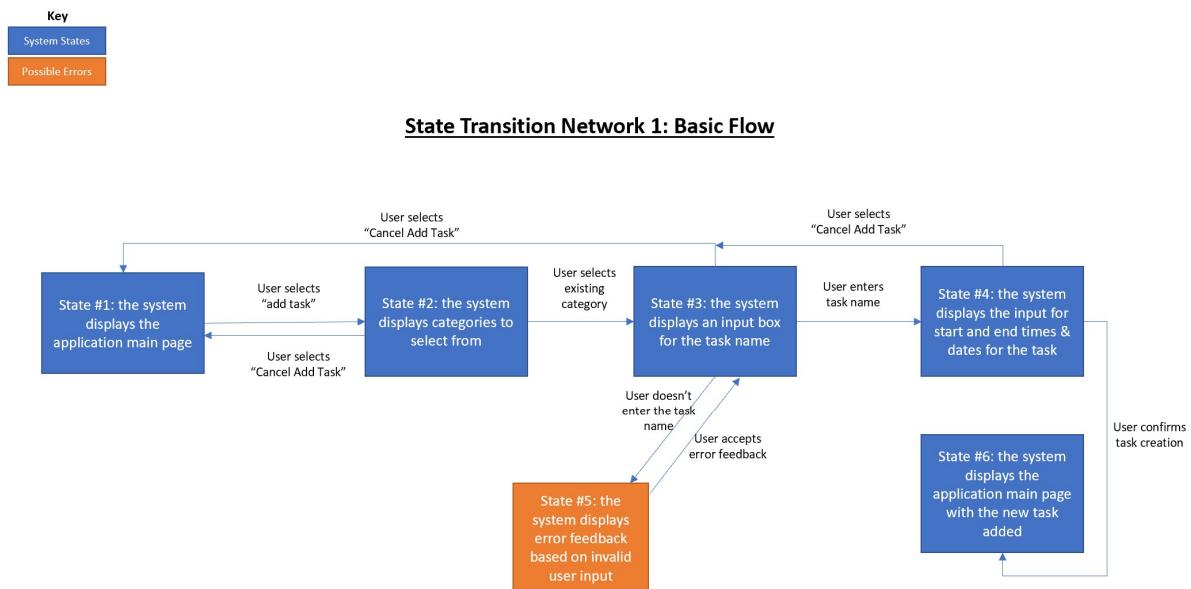
State Transition Network 2: Alternative Flow A2



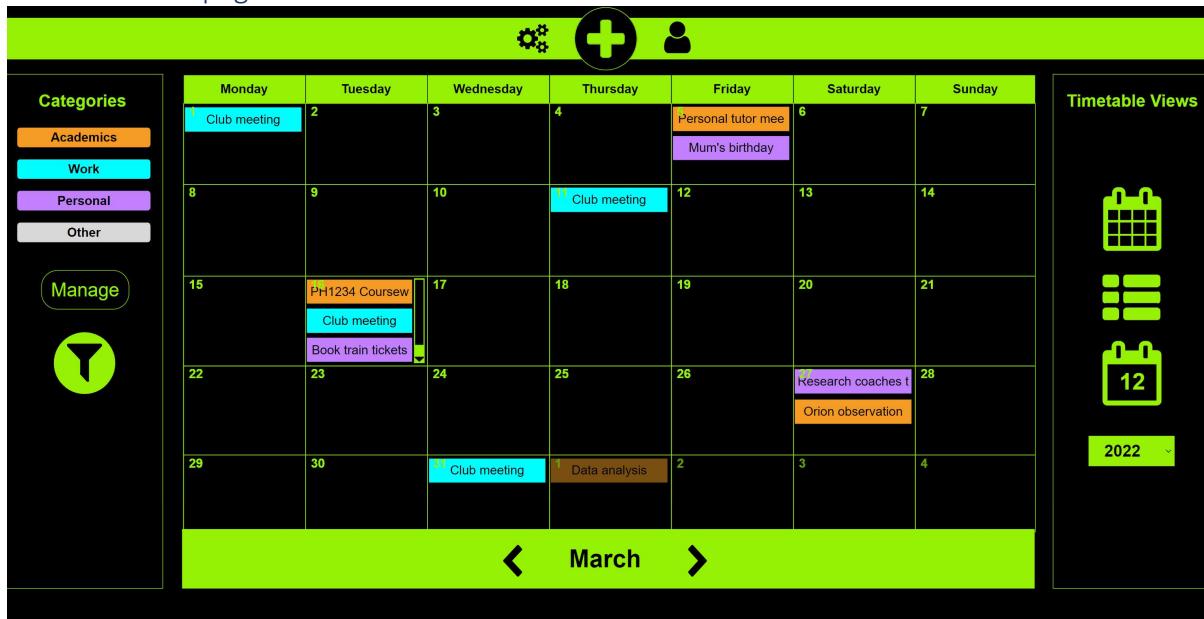
Part 2

For each STN screen state, any duplicate interface elements (such as the Add Task button) will be described once to avoid repetition.

STN1: Basic Flow



State #1: Main page



The main page is the default screen shown to user when the application is opened. It is designed to show as much calendar information as possible to make it easier for the user to identify which day/task they are interested in viewing, which makes using the calendar more efficient.

Design

The web application satisfies Nielsen's "aesthetic and minimalist design" heuristic through the use of a simple colour scheme (only 2 main colours) and progressive disclosure, where more options are revealed after selecting the relevant button (eg viewing task details after clicking on a task or seeing

checkboxes after selecting to filter tasks). This reduces the user's view of rarely needed options and keeps them focused on the most important features which they will be using all of the time.

Consistency

Every item on the app excluding category colours uses the same shade of black and green for simplicity. Primary and secondary buttons follow the same styles throughout the application, and the same front weight is used across calendar scope views. Every border on the web-app is 1px thick, solid colour, and the same shade of green as the other borders. The same font is used throughout the application, and font size is used to separate items by meaning. All buttons have rounded corners to indicate that they are buttons that can be interacted with. Screen elements are centred within their containers to provide equal spacing on opposing sides of the element.

This all provides consistency through repetition, and results in the Aesthetic Usability Effect which makes users more likely to enjoy using the calendar.

A calendar item is always referred to using only the term "Task" to ensure that the user always knows which part of the calendar they are modifying or interacting with. Also, dates and times are always show in the same format: 24 hour clock and DD/MM/YYYY. This satisfies Nielson's "consistency and standards" heuristic.

Add Task Button

The Add Task button uses the plus sign design to make it simple, which reduces the visual load on the user. It is the largest button in the navigation bar (and on the top area of the screen) which makes it easy to find, identify, and separate its meaning from the other, miscellaneous, navigation panel buttons. It's position in the centre top screen and large size makes it fastest to navigate to, following Fitts' law.

The plus sign is often commonly used to show that the button adds something to the screen. This satisfies Nielson's "Match Between System and the Real World" heuristic.

Calendar

The calendar shows the days and the month of its tasks. Days relating to the next month are darker to allow the user to focus on the tasks relating to this month.

Tasks are colour-coded based on their category to allow for an insightful overview of the balance of the user's activities.

The navigation bar at the bottom of the calendar allows the user to switch between adjacent months/days/weeks (depending on the scope).

The calendar takes up most of the screen, making each cell as large as possible. This will not only allow the user to see more information under each day, but also to move the cursor to their day of interest more efficiently. If a calendar cell contains more tasks than it can display, then a scroll bar is shown to enable the user to view all tasks in that cell. If a calendar task is of more than one category, than its colour is split in the middle to reflect it's belonging to both categories.

The days in the month scope view are a grid of equals, as they display each day in the same format and size as every other day and are arranged in a grid pattern. This is done because each day has the same importance as the other days, so they all have the same visual weight.

Categories Sidebar

This sidebar includes all information and actions regarding task categories, which utilises the Gestalt principle of Proximity. Grouping related elements in one area of the screen makes it easier to navigate to because the user knows where to expect to find what they are looking for. The sidebar used the same background as the body of the website to avoid visually overloading the user, however it also uses a 1px solid green border to visually separate the categories sidebar from other, unrelated, items on the screen.

Categories Flags

All the categories available in the calendar app (both default and user-added) are shown in the sidebar to create a visual key to help the user remember which category each task belongs to. This way, they will spend less time searching for information as they will not need to keep remembering what each colour signifies. This satisfies Nielson's "recognition rather than recall" heuristic. The user does not need to keep in mind what all the different categories are because they are all presented in the sidebar.

Filter Icon

The Filter icon is the same size as the Add Task icon for consistency, as well as to indicate that it is of a similar importance to the Add Task button, as all users will need to use this functionality often. The funnel is commonly used to indicate a filtering functionality, which satisfies Nielson's "Match Between System and the Real World" heuristic.

Manage Button

The manage button has reverse colours to the filter button, demonstrating hierarchy. As users are expected to need this function less regularly than the filter function, its button contrast is reduced to make the filter button more visible. Throughout the web-application, secondary buttons use this design to allow the primary button to draw more of the user's attention.

Timetable Views Sidebar

This sidebar groups together buttons that control how the timetable is presented, for the same reason and purpose as the categories sidebar. It allows the user to view the calendar in 3 scopes (month, week, and day) in different levels of detail.

The calendar icons were chosen in a way that resembles a physical calendar. This satisfies Nielson's "Match Between System and the Real World" heuristic.

Organisation

The tasks are organised in the calendar using 2 LATCH criteria: by Time in the calendar and by Category in the left sidebar. This makes it easy for the user to find the task they are looking for, but it's not too overwhelming. While having multiple ways to organise data, I believe that too many criteria will get confusing as the user will take too long to decide which way to organise their data

State #2,3,4: Adding a task

Buttons

The primary Add Task button is brighter than the secondary buttons on the screen to draw attention to it, as most of the time it will be the button that the user is most interested in clicking on. This difference in contrast established a button hierarchy which makes it faster for the user to complete a form because they can easier recognise which button will take them to the next state.

There is a visible “Cancel Add Task” button which tells the user how to quit the Add Task screen in case they change their mind. This satisfies Nielsen’s “user control and freedom” heuristic because the user has control over the system state.

Layout

The input fields are divided into two columns to maximise the use of space on the screen. Since the two columns are not separated by semantics, there are no divider lines. The vertical spacing between groups of inputs is the same for consistency. The buttons below the columns and the input fields are the same width to further solidify this visual separation, following the Gestalt Law of Closure. The user will see the two columns are two rectangles, facilitating the visual separation of the items. The Cancel Add Task button unites the two columns by stretching from the middle point of the left button to the middle point of the right button, showing that the two columns have the same semantic meaning despite being separate visual items.

Every form in the web-app uses a column for labels and a column for inputs. The labels are right-aligned and the inputs are left-aligned, creating a sense of continuity and connectedness. By aligning them this way, there is less distance between the labels and the inputs, making it easier to follow the line along which they are aligned, and connect which label belongs to which input.

The input-label columns are read from left to right and top to bottom, which is intuitive for users in western countries, including the UK. This satisfies Nielsen’s “match between system and real world” heuristic.

List Box

The list box is used for category selection because it allows the user to select one or more pre-defined categories for the task. Using a list box helps save space on the screen as only one option is visible at any one time, but can be changed through scrolling. A list box was chosen over a text box because it removes the need to remember the available categories, making the labelling process faster. This satisfies Nielson’s “recognition rather than recall” heuristic. It also makes error-checking easier because the user is unable to input an invalid (eg non-existing) category name. however, should the user want to add a new category, they can still do it using either the Add Category button or in the Categories Sidebar by entering the Manage menu.

Dropdown Menu

The input method for the date and time of the task was chosen for similar reasons to the category input. The user can only choose from a list of existing days, months, years, hours, and minutes, as they are immutable and therefore the list will never need to be updated. There is also no need for the user to provide their own original input, unless they can govern the laws of space-time (unlikely).

The input date follows the UK format of DD/MM/YYYY, and the times follow the military time format, both of which follow satisfy Nielsen's "match between system and real world" heuristic for the UK user.

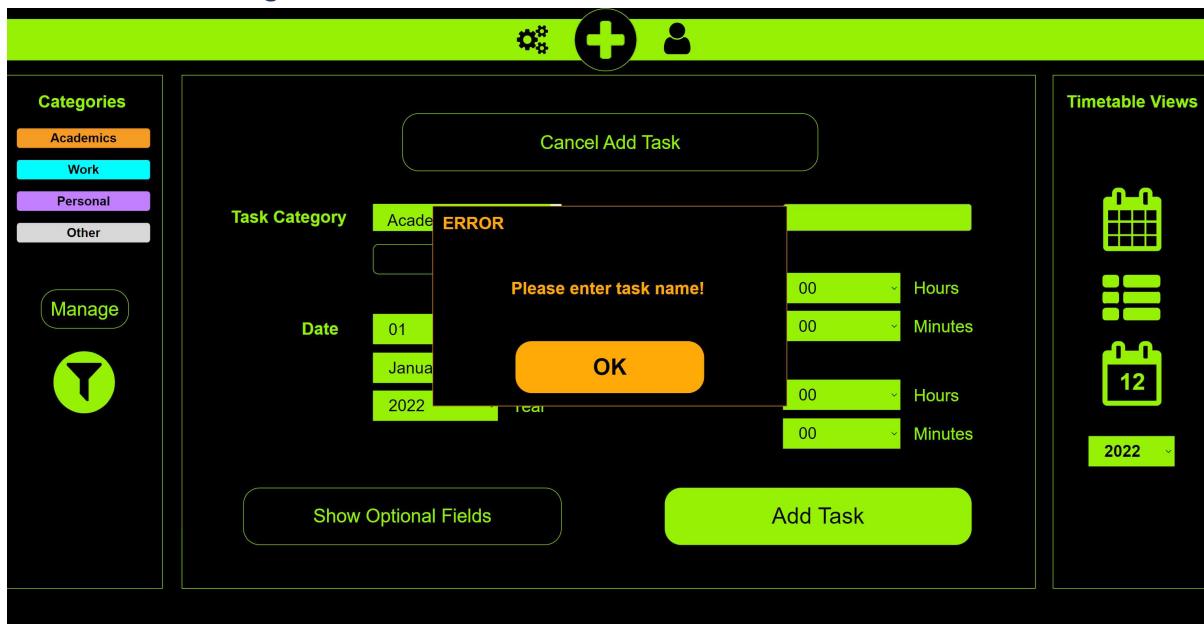
Text Box

The task name can be any string of characters so a text box is used for its input to allow the user to be flexible with their naming. Limiting the name input to a more restrictive method, for example, a list, will limit the level of description that the user will be able to give to their task with little benefit to them.

Error prevention

To satisfy Nielson's "help users recognise, diagnose, and recover from errors" heuristic, the number of fields which can cause an error is minimised. The date and time inputs default to the current date and time, so that if the user forgets to set a date the task can still be created without issues. The category is also set to "Other" by default. Only the mandatory text field inputs can result in an error from a missing input. Having fewer possible errors means that the user is less likely to encounter them, which improves user experience.

State #5: Error dialogue box



This pop-up box is shown if the user does not enter the task name.

The error messages clearly state what caused the error and how it can be solved. This satisfies Nielsen's "help users recognise, diagnose, and recover from errors" heuristic.

Colour Choice

The error box is designed to be of a contrasting colour to the main accent colour of the calendar theme to show the user that it is not part of the regular functionality and so an action is needed to

return to it. The colour orange is also associated with alertness and alarm, and as such it is often used in application error messages, so the user will be familiar with it. To further show that this box is an error box, the word "ERROR" is placed at the top left corner to remind the user that they have done something wrong.

Button

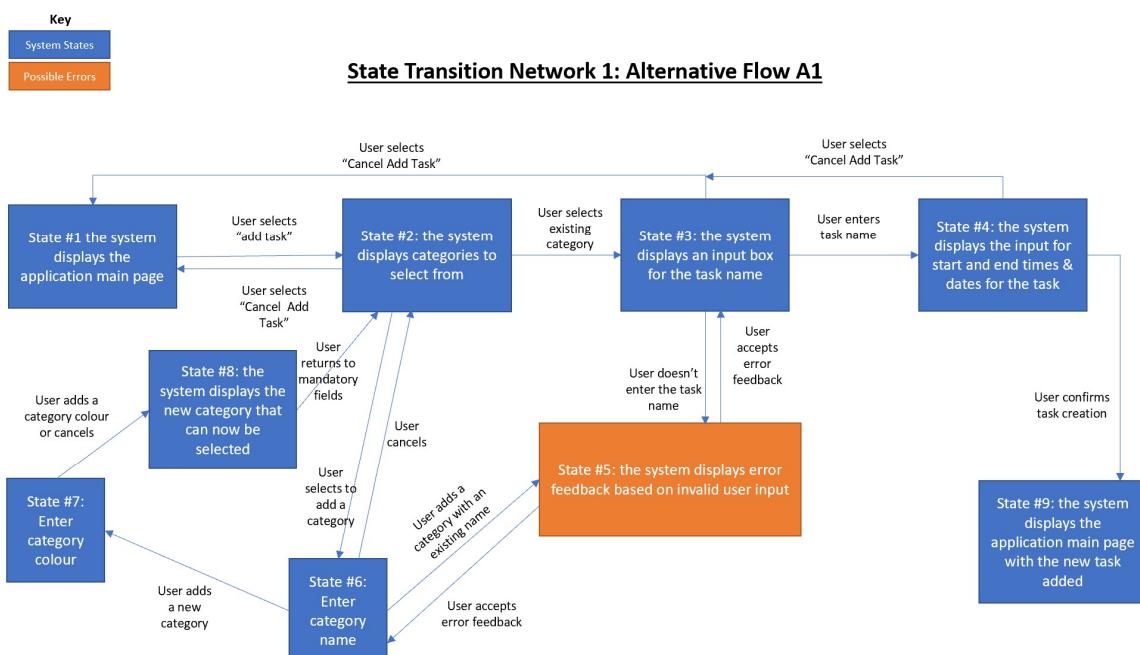
The "OK" button uses the same style as the green primary button style, just with a different accent colour. This means that it will still be recognisable to the user as the button that they need to press to exit the error box.

State #6: Main Page with the task added



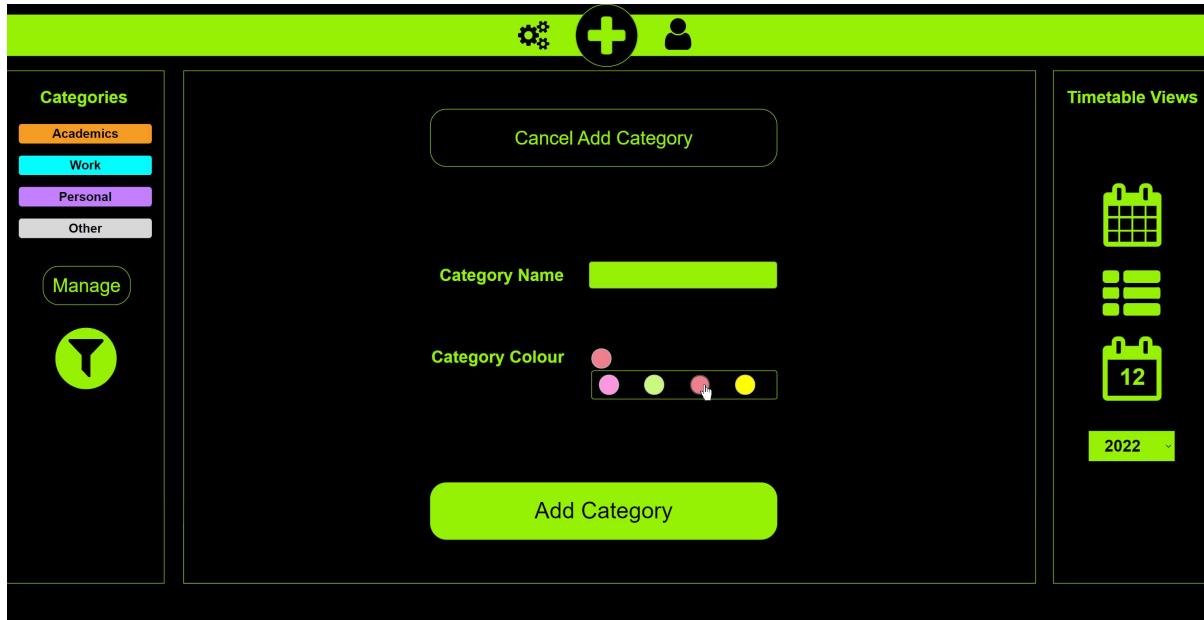
The design for the newly created task follows the design of the main page.

STN1: Alternative Flow A1



The designs for states 1,2,3,4, and 9 have already been showcased and explained in the section above.

State #6,7: Adding a new category name and colour



The alternative flows use Progressive Disclosure to avoid overwhelming the user with a range of optional choices and to limit the amount of information showing on the screen at any one time. The Add Category screen is displayed only when the user chooses to click on the Add Category button. The buttons are designed with the same reasons as on the previous screen.

Layout

The page is laid out in one column as there are few enough elements to not require separation. The buttons, labels, and input form a visual rectangle following the Gestalt principle of Closure, grouping the Add Category items and separating them from the other items on the screen, such as the sidebars.

Text Box

The user is free to name their new category anything that does not equal to an already existing category, and a text box provides them this freedom and flexibility.

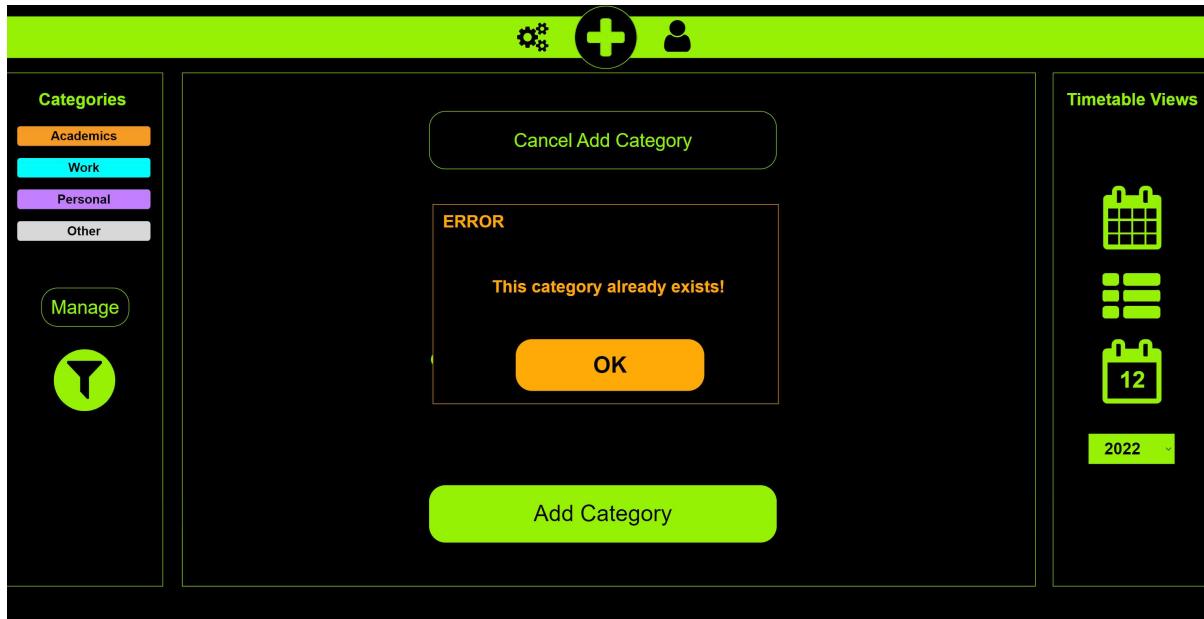
Colour Picker

The colour picker is shown when the user clicks on the coloured circle. The colour picker suggests a selection of colours that are visually different to the colours used in existing categories to ensure that the new category colour is visually distinct. This makes it faster for the user to select a unique colour than a more sophisticated gradient-style colour picker as they don't need to adjust the shade to be more distinct or think about what colour to use themselves.

The word “colour” is the British spelling, which satisfies Nielsen’s “match between system and real world” heuristic for the UK user.

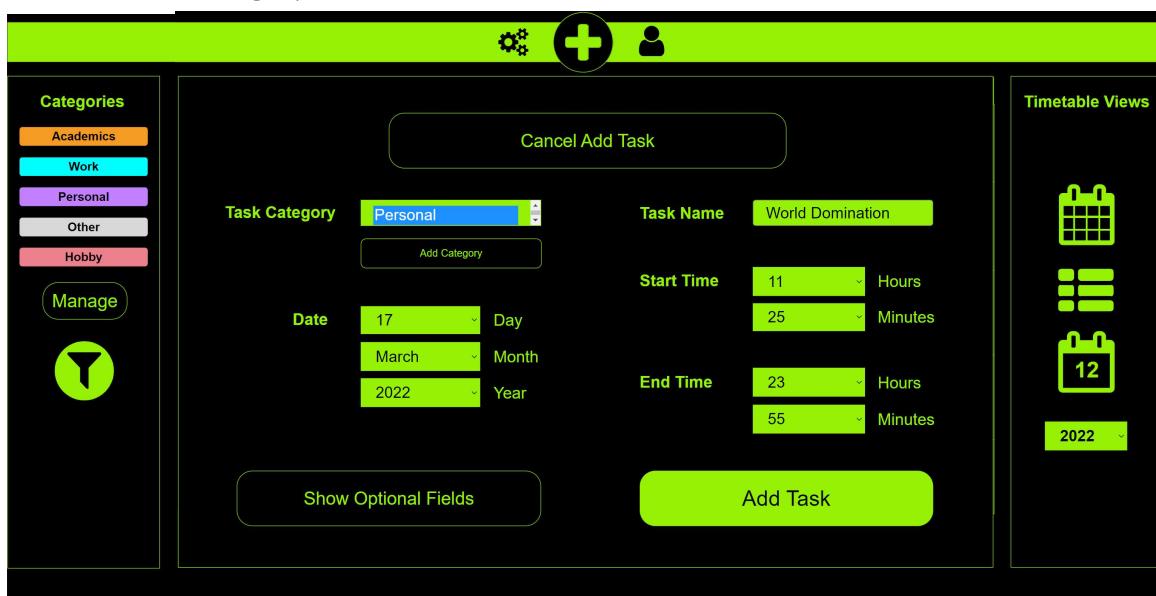
C2041844

State #5: Error dialogue box



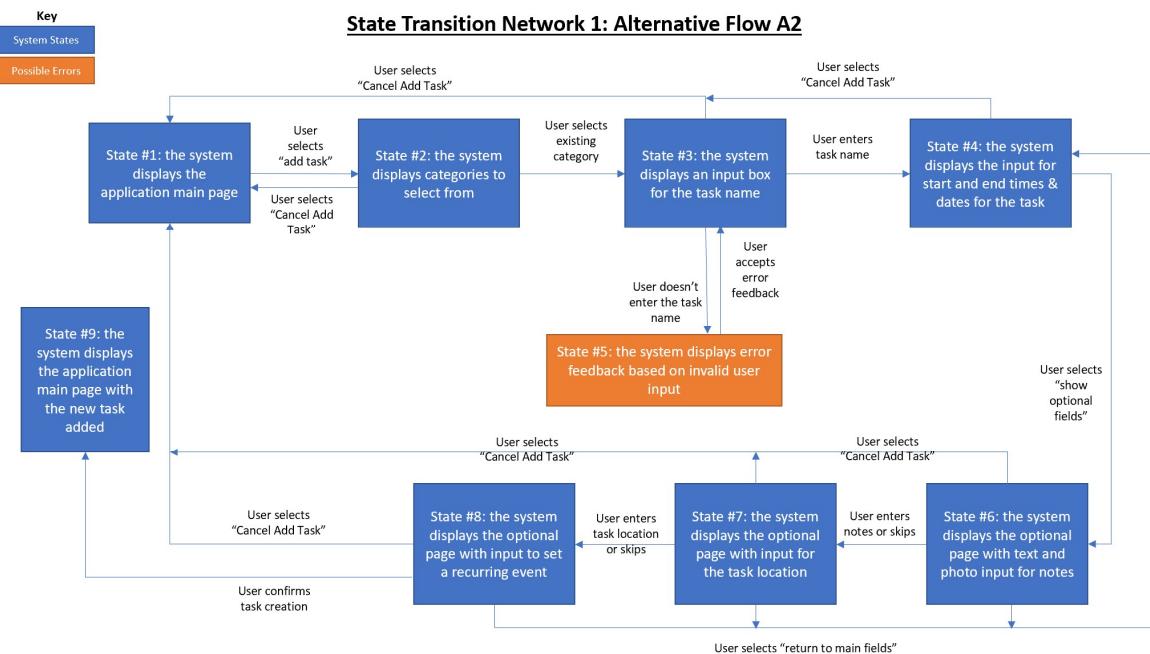
This pop-up box is shown if the category name that the user entered already exists. The program checks for lower- and upper-case variations of this category.

State #8: New category shows in the sidebar



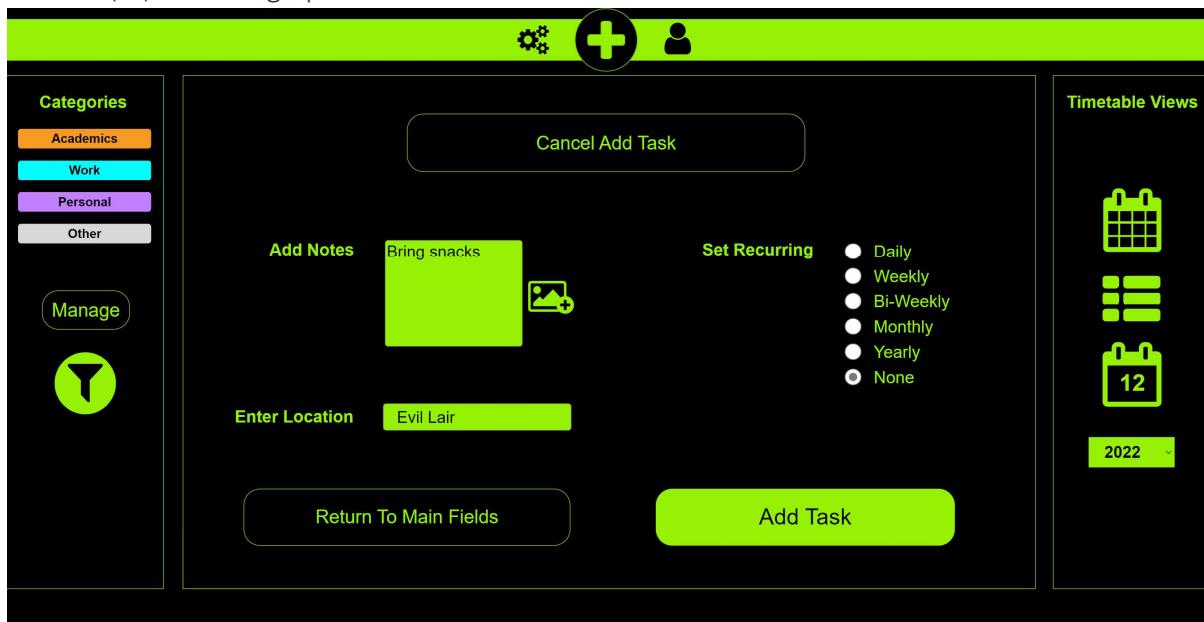
The sidebar is updated with the newly added category. The Manage and Filter buttons are moved down when each new category is added to make space for more categories. If a lot of categories are created, a scrollbar appears to allow the user to view all the categories within the categories sidebar.

STN1: Alternative Flow A2



The designs for states 1,2,3,4, 5 and 9 have already been showcased and explained in the section above.

State #6, 7, 8: Adding optional information



Once again, the inputs are arranged in two columns to make better use of the window space, like in the Basic Flow.

Text Box

The notes and location for the task can be any string of characters, so a text box is used for their input.

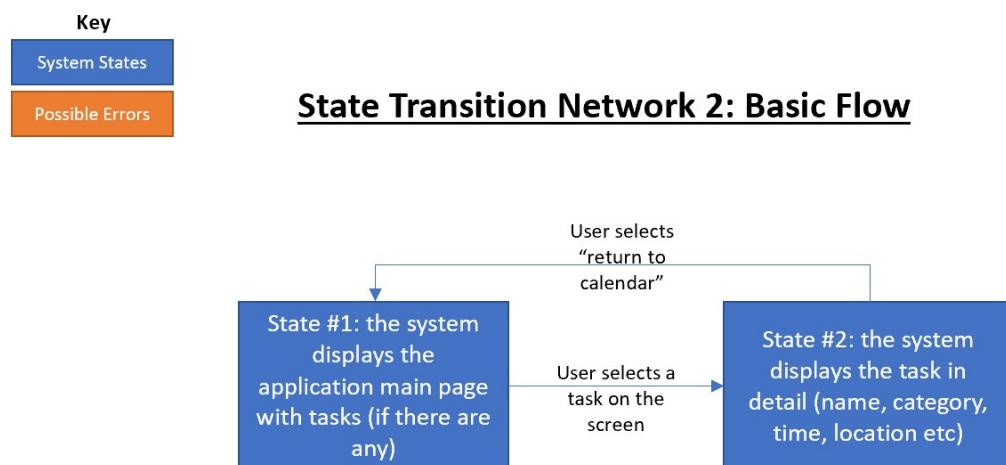
I have considered using a filtered drop-down list (where the list of matches becomes shorter as you type more characters in) like in the Google Maps location input. However, not every task location will be a geographical location that is indexed in a map database (eg. 42 Wallaby Way), because

sometimes the user may need to enter a room or stall number for the event that they are attending (eg. Abcws/1.39). So in this case it is best to use a text box to allow the user to enter a location that is not a formally mapped place.

Radio Buttons

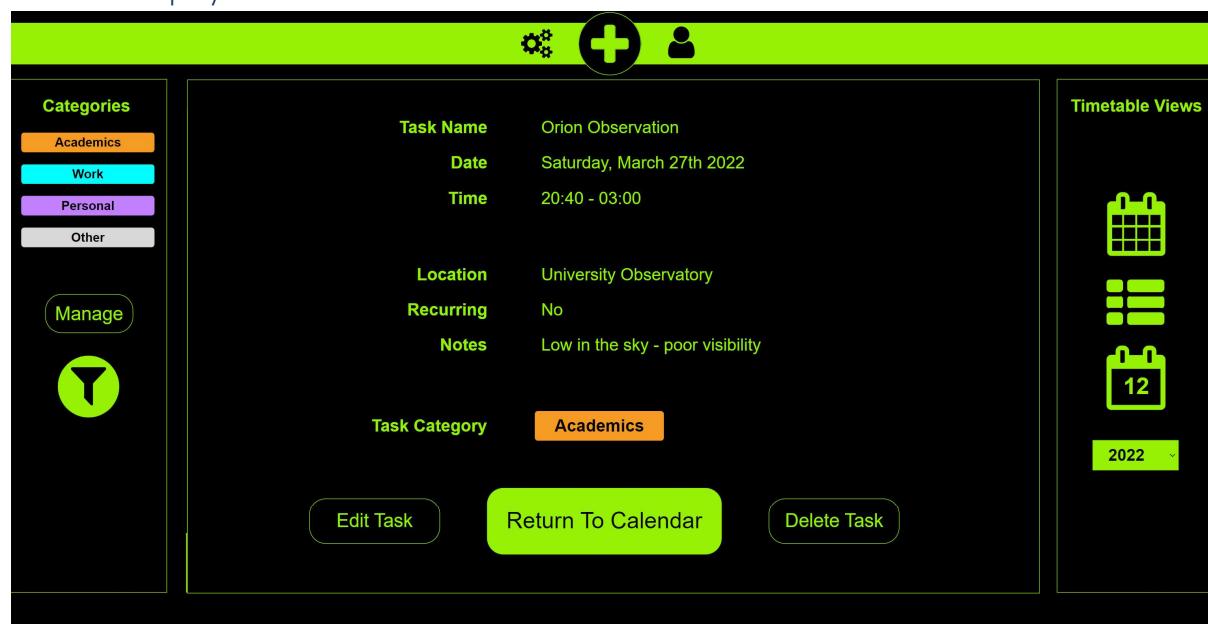
The task can be set to recur at certain common time intervals. Recurring calendar events commonly do not happen in more than one recurrence pattern, so a set or radio buttons is the best input method in this case. The user can only select one of the given recurrence patterns per task.

STN 2: Basic Flow



State #1 has already been showcased and explained in the section above.

State #2: display the task in detail



Layout

The task details are separated in semantic groups. The first group is the mandatory details, without which the task cannot be created. The second group is the optional details, which can be absent

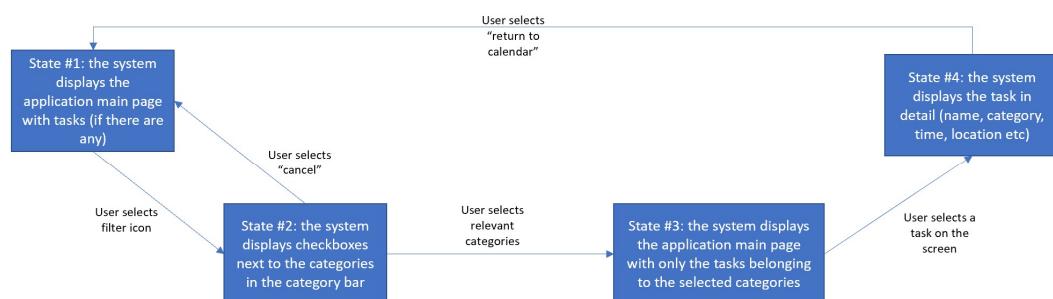
from a task with no effect on its creation and behaviour within the calendar. The last group is the task's category. It's placed last because it is the least important detail in this page, because the user will have already seen the task's category when they were selecting it, so there is no need to remind them of it before the rest of the task details.

The screen is separated into labels and details. The labels are right-aligned and the details are left-aligned, creating a sense of continuity and connectedness. The details and labels are shifted slightly to the left of the centre line to draw the user's focus to the details, which are the most important part of the page

STN 2: Alternative Flow A1

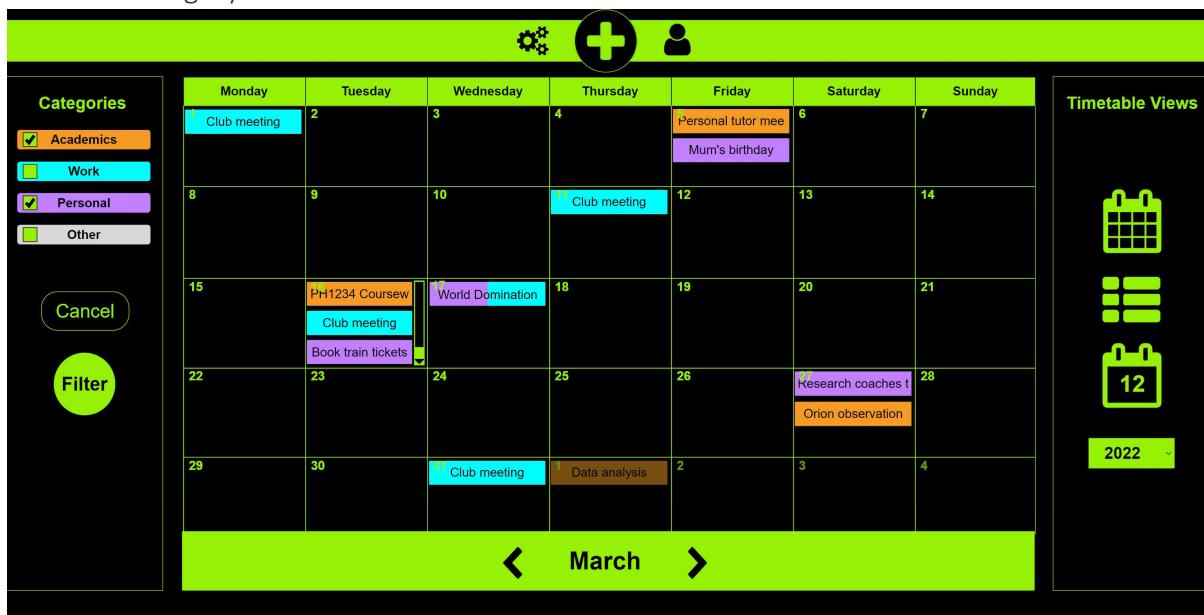


State Transition Network 2: Alternative Flow A1



States #1 and #4 have already been showcased and explained in the sections above.

State #2: category filter checkboxes



When the user clicks on the filter icon, a set of check mark boxes appear next to the category labels to show which categories will be displayed on the calendar after the filter has been applied.

When the filter icon has been clicked, it changes to the text "Filter" to provide visual feedback that the state of the sidebar has changed. The "Manage" button becomes "Cancel" because there is no need to edit the categories while applying a filter.

Buttons

There is a visible “Cancel” button which tells the user how to quit the filtering screen in case they change their mind. This satisfies Nielsen’s “user control and freedom” heuristic because the user has control over the system state.

Check Boxes

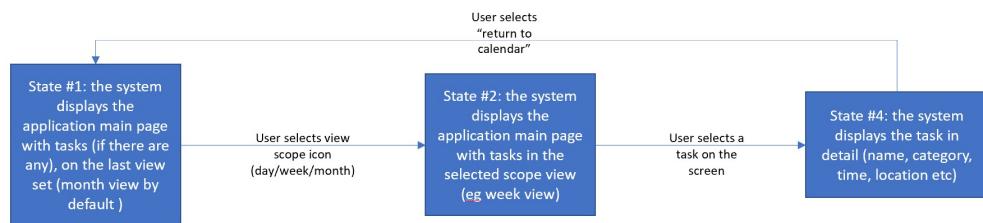
Check boxes are used to filter results because the user is allowed to select multiple categories to filter, which cannot be done using radio buttons. There is also no need for selection input types such as a drop-down list because the category labels are already present in the sidebar and so can be used as a basis for the filter input without over-complicating the process. The filtering function utilises the existing sidebar and category flags to satisfy Nielson’s “recognition rather than recall” heuristic. The user does not need to input what categories to enter from memory (eg using a text field) to make filtering faster and easier.

State #3: display the filtered results

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1				4	Personal tutor mee Mum's birthday	6	7
8		9	10	11	12	13	14
15		PH1234 Coursew	17	18	19	20	21
22		23	24	25	26	Research coaches t Orion observation	28
29		30	31	1 Data analysis	2	3	4

Once the results have been filtered, the tasks with the categories that were not selected are removed from the calendar view

STN 2: Alternative Flow A2

State Transition Network 2: Alternative Flow A2

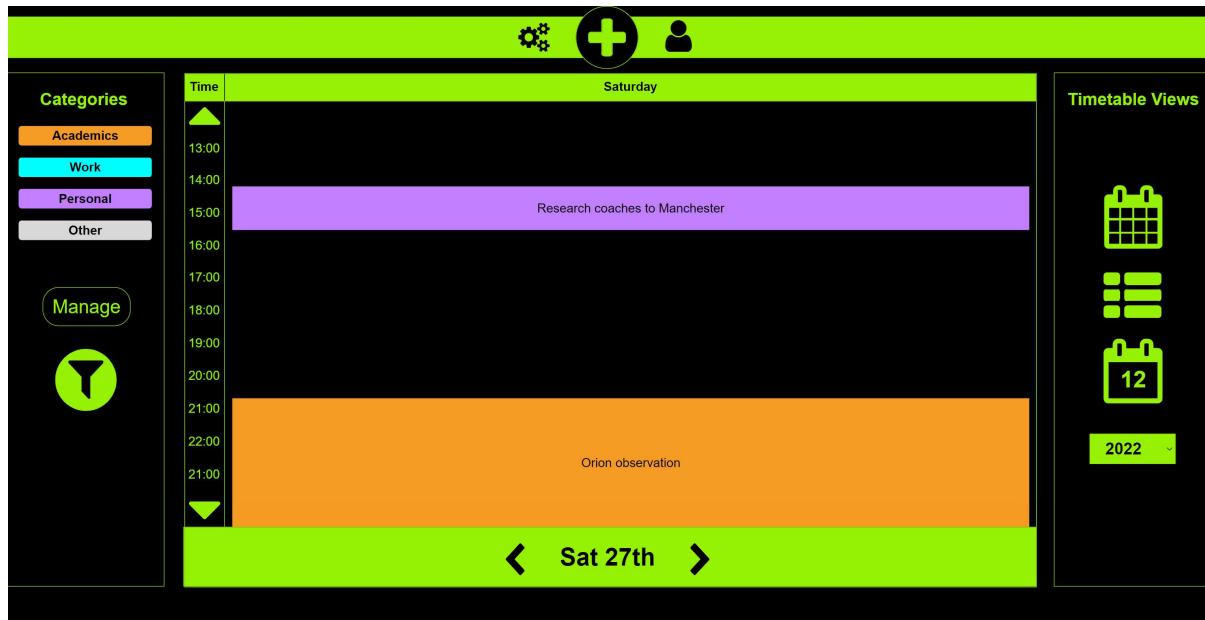
States #1 and #4 have already been showcased and explained in the sections above.

State #2: display selected scope

Week view:



Day view:



Both the scopes follow a similar design.

Unlike the month scope, they show task sizes proportionate to the tasks' lengths, to allow the user to visualise how big each task is. This is often a better representation of how much one's day is "packed" than the equal sized month view, and allows the user to make better decisions of when to schedule new tasks.

Both scopes do not show the full 24 hours of the day at the same time. This is because the task lengths have precision of 5 minutes, and splitting 24 hours into 288 5-minute segments will make it hard to see quick tasks that take 5-15 minutes. Therefore the screens allow the user to scroll up and down within the calendar window to see the task lengths in higher resolution.