Mobile User Experience

Application Description [1.0]

What is the app and what is its one main purpose? [1.1]

The application will allow users to manage their assessment items, timetable, grades and goals in a singular application.

Additionally, the aforementioned processes presented in the application will be reimagined to include the influences of agile methodology and tools to enhance user engagement and outcome.

The application's main purpose is to make managing academic workload more convenient by packaging commonly used tools in one application, increase student interest and achievement in learning by reducing common barriers (e.g. stress, unawareness of tools, bad study habits) and introduce students to industry tools and functions (Agile commonly used in various industries) .

What one problem/concern does it solve? [1.2]

Many students struggle to maintain a healthy balance between personal and academic commitments. This is often attributed to poor planning, procrastination caused by perceiving assessment items as insurmountable (mismanagement of time/task) and lack of experience directing personal progress.

This application will equip students with the tools to manage deadlines and goals in one application (reducing the complexity of navigating multiple solutions). The provided suite of tools aims to support the user learning process by teaching students how to break assessment items into achievable tasks, keep track of progress on assessment items and ensure they are holding themselves accountable to their goals.

What are the main features of the app? [1.3]

assessment manager: Users can record assessments (story) and add them to the 'Action' board (kanban board). From here, they are encouraged to break these stories down into achievable tasks and track their progress by placing tasks in appropriate columns (To-do, In-progress, complete).

Timetable manager (Stretch Goal): Users can create a timetable to record their day-to-day activities (classes, tutorials, study period). They can then set reminders to be notified of events scheduled in their timetable and easily modify their schedule as needed.

Result manager: Users can record their grades from their assessments (track academic progress) and calculate what grades they need to achieve in future assessments to reach their goals.

Goal manager: Users can record their goals at the start of a specified period. They are then periodically notified (frequency determined by the user) to reflect on their progress to ensure they stay accountable to their academic goals.

Who will use the app? [1.4]

Target users will be university students nationwide.

This group of users will be between the age of 17-25, are of any gender and have a desire to maintain/improve the way they manage their time, output and course workload.

Additionally, they must be familiar with mobile applications and have a willingness to explore new solutions.

This user group is likely to expect an engaging, simplistic and intuitive UI. For this reason, the solution will prioritise minimalism and modern illustration/colour scheme.

Additionally, the needs of this group is relatively broad and contains various niches (e.g visual impairments, language barrier, sensory barriers). For this reason, I have included accessibility features in my stretch goals.

How long would it take for you to develop the application? [1.5]

The solution should be achievable with 45 hours of work.

I will work independently throughout the development process (Assignment 1-3).

The benefit of this is that I should be able to complete tasks within estimated time frames as I am unlikely to experience any external disruption (e.g. group conflicts, scheduling conflicts).

However, the cost is the limitation of the workload I will be able to complete individually.

The accuracy of my estimation may be negatively affected by my lack of experience in developing a mobile solution.

For this reason, I intend to start development as early as possible. Allowing myself an additional 5 hours to work on the project if needed (50 hours in total).

However, if this is insufficient, I will remove the timetable component (Stretch Goal) and other minor features (e.g. help button, accessibility).

Story	**Task**	**Time (hr)**	**Total Time per story**
Get basic components set up			
	Set-up IDE	1	
	Get familiar with JS	1	2
Set up on-boarding processes			
	Set-up login	1	
	Set-up database	5	
	Set-up registration	1	
	Set-up introduction to application	1	
	Add minor components (e.g email verification, password reset)	1 (Stretch Goal)	9
Set up ‘Assignment manager’ tab			
	Let users create story (assignment)	1	
	Let users create task (actions which lead to assignment completion)	1	
	Create kanban board	1	
	Let users set/display/modify	1	

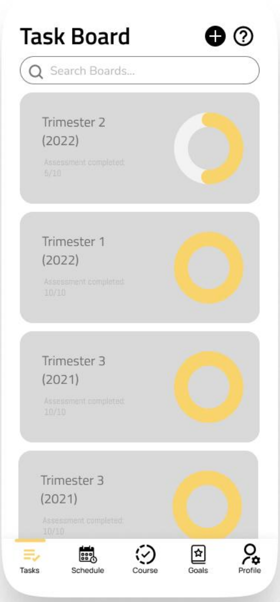
	state of task and story		
	UI Aesthetics	1	
	Create Help documentation	1 (Stretch Goal)	
	Set notification	1 (Stretch Goal)	7
Set up ‘Timetable manager’ tab (Stretch Goal)			
	Let users create fixed events (e.g classes every wednesday)	1	
	Let users create one off events (e.g group meeting)	1	
	UI Aesthetics	1	
	Let users modify existing timetable	1	
	Create Help documentation	1	
	Set notification	1	6
Set up ‘Result manager’ tab			
	Let users add assignments	1	
	Let users add grade	1	
	Let users predict/calculate future grades	1	
	Let users modify existing grades	1	
	UI Aesthetics	1	
	Create Help documentation	1 (Stretch Goal)	6

Set up ‘Goal manager’ tab			
	Let users add goal	1	
	Let users view past goals and logs	1	
	Let users log progress	1	
	Set notifications	1	
	UI Aesthetic	1	
	Create Help documentation	1 (Stretch Goal)	6
Set up ‘Manage Profile’ tab			
	Help documentation	1 (Stretch Goal)	
	Account settings (e.g change password/email, logout)	2	
	Accessibility (e.g colour, text size)	2 (Optional)	
	UI Aesthetic	1	6
TOTAL TIME (without stretch Goal)			**28**
STRETCH GOALS			**14**
DEMO RECORDING			**3**
APPLICATION+DEMO			**45**

Mobile UX Pattern [2.0]

FEATURE 1: ASSESSMENT MANAGER [2.1]

Home Screen (1)



Task Board

Search Boards...

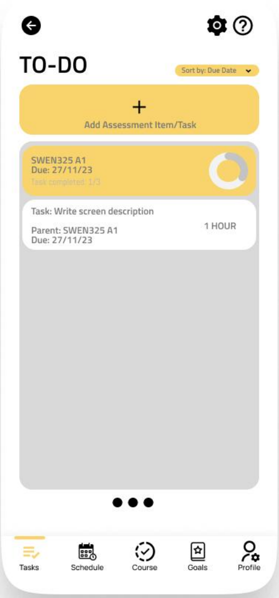
Trimester 2 (2022)
Assessment completed: 5/10

Trimester 1 (2022)
Assessment completed: 10/10

Trimester 3 (2021)
Assessment completed: 10/10

Trimester 3 (2021)
Assessment completed: 10/10

Task Board (To-do) (2)



TO-DO

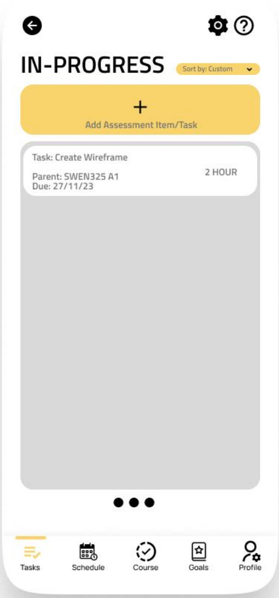
Sort by: Due Date

Add Assessment Item/Task

SWEN325 A1
Due: 27/11/23
Task completed: 1/3

Task: Write screen description
Parent: SWEN325 A1
Due: 27/11/23
1 HOUR

Task Board (In-Progress) (3)



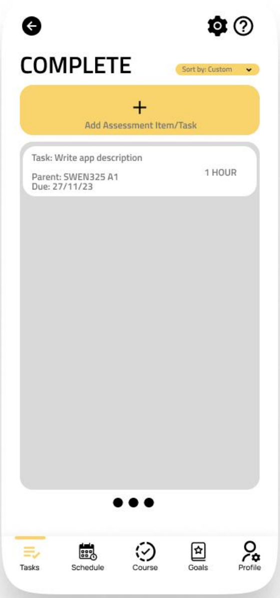
IN-PROGRESS

Sort by: Custom

Add Assessment Item/Task

Task: Create Wireframe
Parent: SWEN325 A1
Due: 27/11/23
2 HOUR

Task Board (Complete) (4)



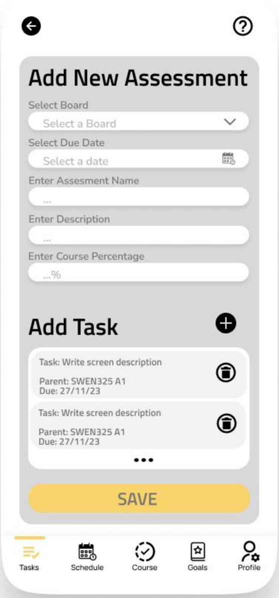
COMPLETE

Sort by: Custom

Add Assessment Item/Task

Task: Write app description
Parent: SWEN325 A1
Due: 27/11/23
1 HOUR

[ACTION] Add Assessment (5)



Add New Assessment

Select Board: Select a Board

Select Due Date: Select a date

Enter Assesment Name: ...

Enter Description: ...

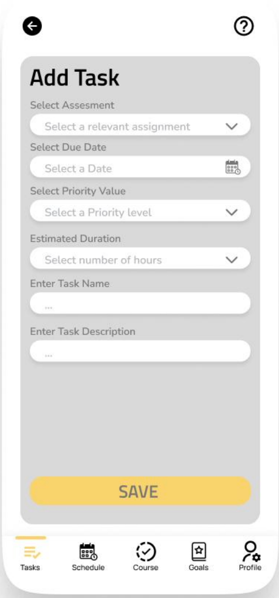
Enter Course Percentage: ...%

Add Task

Task: Write screen description
Parent: SWEN325 A1
Due: 27/11/23

Task: Write screen description
Parent: SWEN325 A1
Due: 27/11/23

[ACTION] Add Task (6)



Add Task

Select Assesment: Select a relevant assignment

Select Due Date: Select a Date

Select Priority Value: Select a Priority level

Estimated Duration: Select number of hours

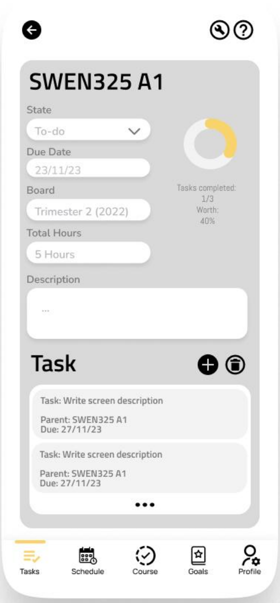
Enter Task Name: ...

Enter Task Description: ...

View Assessment (7)

View Task (8)

Action Confirmation (9)



SWEN325 A1

State: To-do

Due Date: 23/11/23

Board: Trimester 2 (2022)

Total Hours: 5 Hours

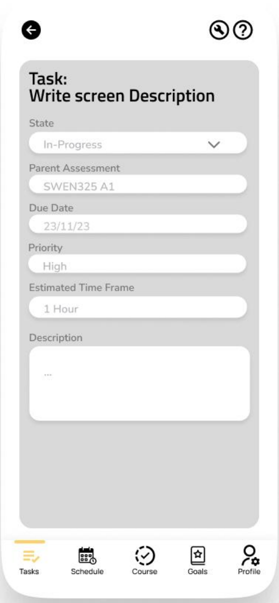
Description: ...

Tasks completed: 1/3
Worth: 40%

Task

Task: Write screen description
Parent: SWEN325 A1
Due: 27/11/23

Task: Write screen description
Parent: SWEN325 A1
Due: 27/11/23



Task:
Write screen Description

State: In-Progress

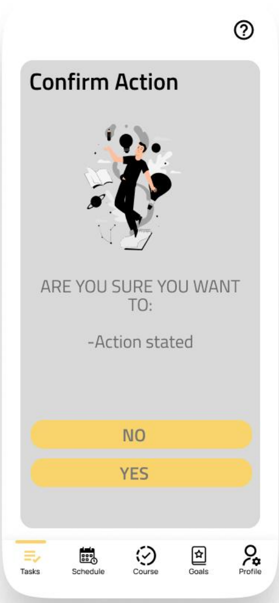
Parent Assessment: SWEN325 A1

Due Date: 23/11/23

Priority: High

Estimated Time Frame: 1 Hour

Description: ...



Confirm Action

ARE YOU SURE YOU WANT TO:

-Action stated

FEATURE 1: UI PATTERNS

FILM STRIP + LOCATION WITHIN + ON-SCREEN GESTURE (Steven Hoober's mobile patterns)

WIREFRAME: 2,3,4,5,7

The 'Film strip' pattern used in combination with the 'Location Within' pattern and 'On-screen gesture' pattern can be observed in the wireframes stated above to display the items in each state (To-do, In-progress, Complete) and display tasks relevant to an assessment item. This is visually represented by three dots.

The 'Film strip' pattern is appropriate for these functions as we want it to be convenient for users to interact with the components displayed on a page (e.g. task object).

Using the 'Film strip' pattern, we can display the entire component on one page (making it easier to select), display summarised information (reduce unnecessary interaction) and reduce the potential for error as there is a clear distinction between interactive actions (swipe to navigate, tap to select).

The objects' nature also supports the use of the 'Film strip' pattern as they are independent (information of the previous object is unrelated to future objects) and small in nature (e.g. three board states, a limited number of tasks).

The 'On-screen gesture' pattern is used to make navigating items easier as we are limited by a small screen size. Therefore, using a button or scroll bar is not appropriate. Instead, users can view different sections of the film strip by swiping left and right.

The 'Location within' pattern is used to supplement the navigation of the film strip. The dots indicate the possible navigation action from the current state (e.g. swipe left/right). As the objects being displayed are organically positioned, there is no need to use numerical values.

SEARCH-WITHIN (Steven Hoober's mobile patterns wiki)

WIREFRAME: 1

The 'Search-within' pattern can be observed in the wireframe stated above in the form of a search bar to locate relevant boards quickly.

The 'Search-within' pattern is appropriate for this function as we do not want to inconvenience users by making them scroll through an extensive list of irrelevant boards. By allowing users to search for the desired board at the top of the screen, they can explicitly filter out irrelevant boards.

The nature of how the boards are identified also supports the use of the 'Search-within' pattern as the title of the boards are represented by text; the most appropriate filtering method is the 'Search-within' pattern.

SORT AND FILTER (Steven Hoober's mobile patterns wiki)

WIREFRAME: 2,3,4

The 'Sort and Filter' pattern can be observed in the wireframe stated above to easily sort the order of assessments and tasks by specified criteria in a given board state. This is visually represented by the drop-down bar labelled 'Sort by:...'.

The 'Sort and Filter' pattern is appropriate for this function by allowing users to organise the list of tasks/assessments within each board state. This removes the inconvenience of manually selecting each task/assessment and reading its context to organise the list of tasks/assessments to our needs outside the application.

The nature of the tasks/assessments also supports the use of the 'Sort and Filter' pattern. This is because when users create tasks/assessments, we ask them to input information that may be relevant to how they would like to organise tasks/assessments (e.g. due date, time estimate, priority).

INPUT AREA + FORM SELECTION (Steven Hoober's mobile patterns wiki)

WIREFRAME: 5,6

The ‘Input area’ pattern used in combination with the ‘Form Selection’ pattern can be observed in the wireframes stated above to input relevant information when creating an assessment or task item. This is visually represented by text boxes and drop-down boxes.

The ‘Input area’ pattern is used explicitly for fields which require custom text or specific tasks (e.g. name, description, percentage) - used sparingly to limit user error. The ‘Form Selection’ pattern is used for information with categorical values (e.g. parent assessment, priority level, time (hours)).

The ‘Input area’ pattern is appropriate for this function as we require custom user input which we cannot predict - using this pattern for custom fields means the object will more closely reflect users’ needs.

The ‘Form Selection’ pattern is appropriate for this function as we can make categorical inputs more convenient for users, align inputs with other system functions (e.g. Sort and Filter) and reduce the possibility of users inputting incorrect information.

BUTTONS + CONFIRMATION (Steven Hoober’s mobile patterns wiki)

WIREFRAME: 5,6,9

The ‘Button’ pattern used in combination with the ‘Confirmation’ pattern can be observed in the save button displayed at the bottom of the wireframe stated above. When a user presses the button to indicate that they want to save the item they created, the input for creating an Assessment/Task item is evaluated. If found to be valid, the user is prompted with a confirmation screen that conveys the result of submitting the action to confirm they want to submit the action.

The ‘Button’ pattern is appropriate for this function as we want to make it as convenient and clear as possible to submit an action. This reduces the possibility of user error and barriers to use.

The ‘Confirmation’ pattern is appropriate for this function as we want to ensure that users are aware of the result of the action they submit. They can proceed with the action if they are confident or return to the form at this stage.

ICONS (Steven Hoober’s mobile patterns wiki)

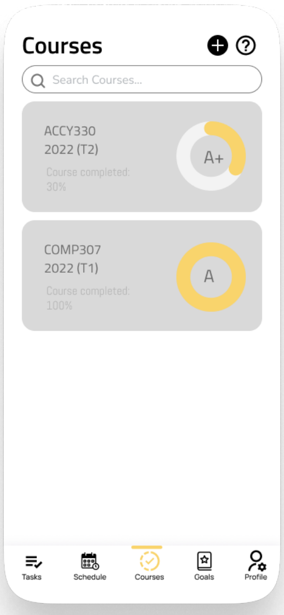
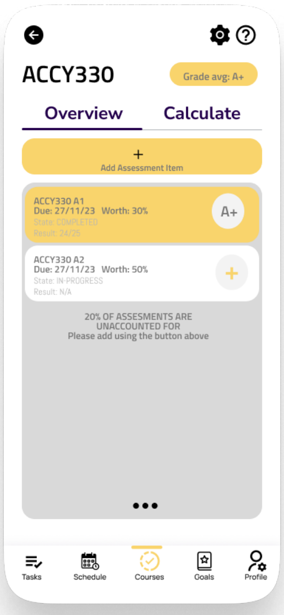
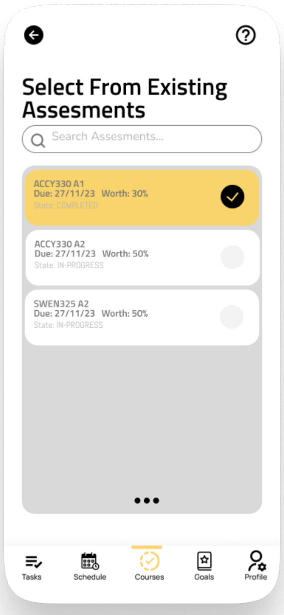
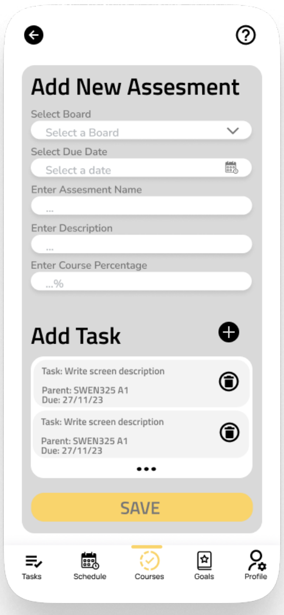
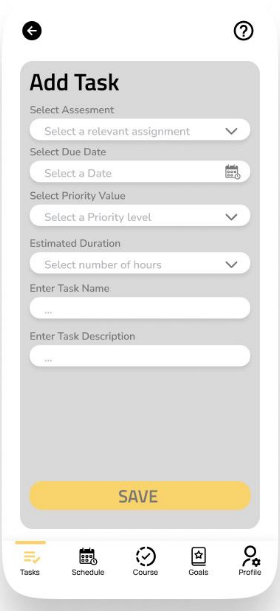
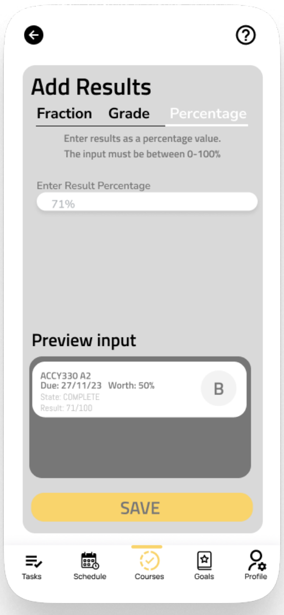
WIREFRAME: 1,2,3,4,5,6,7,8,9

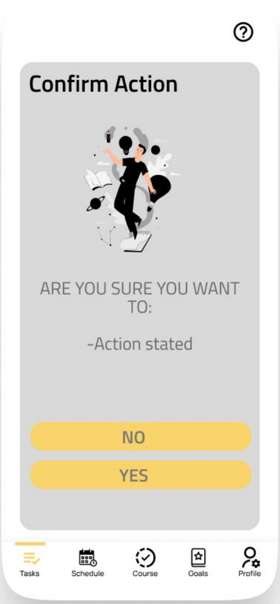
The ‘Icon’ pattern is used to represent actions users can take visually. This is demonstrated with the add, remove, help, settings, and edit icons present in the wireframes stated above. When users select an icon, it allows them to interact with the solution in actions represented by the icons.

The ‘Icon’ pattern is appropriate for this function as we want to concisely communicate potential actions due to a mobile application’s screen size limitation. As we cannot always provide text with icons, we choose only to represent common actions with icons.

OTHER IMPORTANT PATTERNS: [VERTICAL LIST + ON-SCREEN GESTURE], [TITLES]

FEATURE 2: COURSE MANAGER [2.2]

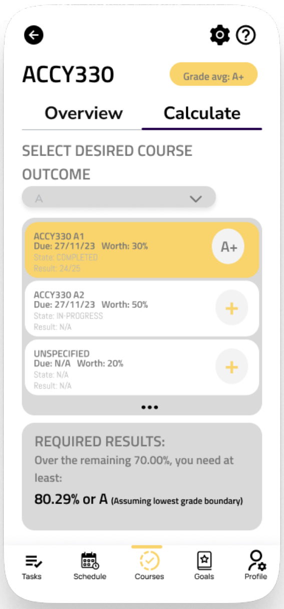
Home Screen (1)	**Course Overview (2)**	**[Action] Add Using Existing Assessment (3)**
 Courses Search Courses... ACCY330 2022 (T2) Course completed: 30% A+ COMP307 2022 (T1) Course completed: 100% A 	 ACCY330 Grade avg: A+ Overview Calculate + Add Assessment Item ACCY330 A1 Due: 27/11/23 Worth: 30% State: COMPLETED Result: 24/25 A+ ACCY330 A2 Due: 27/11/23 Worth: 50% State: IN-PROGRESS Result: N/A 20% OF ASSESMENTS ARE UNACCOUNTED FOR Please add using the button above 	 Select From Existing Assesments Search Assesments... ACCY330 A1 Due: 27/11/23 Worth: 30% State: COMPLETED ACCY330 A2 Due: 27/11/23 Worth: 50% State: IN-PROGRESS SWEN325 A2 Due: 27/11/23 Worth: 50% State: IN-PROGRESS
[Action] Add By Creating New Assessment (4)	**[Action] Add Task To New Assessment (5)**	**[Action] Add Result For Assessment (6)**
 Add New Assesment Select Board Select a Board Select Due Date Select a date Enter Assesment Name ... Enter Description ... Enter Course Percentage ...% Add Task Task: Write screen description Parent: SWEN325 A1 Due: 27/11/23 Task: Write screen description Parent: SWEN325 A1 Due: 27/11/23 SAVE 	 Add Task Select Assesment Select a relevant assignment Select Due Date Select a Date Select Priority Value Select a Priority level Estimated Duration Select number of hours Enter Task Name ... Enter Task Description ... SAVE 	 Add Results Fraction Grade Percentage Enter results as a percentage value. The input must be between 0-100% Enter Result Percentage 71% Preview input ACCY330 A2 Due: 27/11/23 Worth: 50% State: COMPLETE Result: 71/100 B SAVE
Action Confirmation (7)	**Calculate Outcome (8)**	**Create New Course (9)**



Confirm Action

ARE YOU SURE YOU WANT TO:

-Action stated



ACCY330

Grade avg: A+

SELECT DESIRED COURSE OUTCOME

A

ACCY330 A1
Due: 27/11/23 Worth: 30%
State: COMPLETED
Result: 24/25

A+

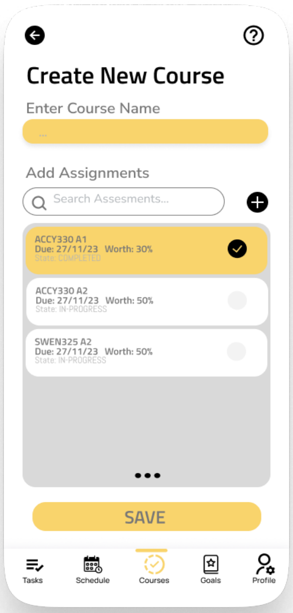
ACCY330 A2
Due: 27/11/23 Worth: 50%
State: IN-PROGRESS
Result: N/A

UNSPECIFIED
Due: N/A Worth: 20%
State: N/A
Result: N/A

REQUIRED RESULTS:

Over the remaining 70.00%, you need at least:

80.29% or A (Assuming lowest grade boundary)



Create New Course

Enter Course Name

...

Add Assignments

Search Assesments...

ACCY330 A1
Due: 27/11/23 Worth: 30%
State: COMPLETED

ACCY330 A2
Due: 27/11/23 Worth: 50%
State: IN-PROGRESS

SWEN325 A2
Due: 27/11/23 Worth: 50%
State: IN-PROGRESS

FEATURE 2: UI PATTERNS

ICONS (Steven Hoober's mobile patterns wiki)

WIREFRAME: 1,2,3,4,5,6,7,8,9

The 'Icon' pattern is used to represent actions users can take visually. This is demonstrated with the add, remove, help, settings, and edit icons present in the wireframes stated above. When users select an icon, it allows them to interact with the solution in actions represented by the icons.

The 'Icon' pattern is appropriate for this function as we want to concisely communicate potential actions due to a mobile application's screen size limitation. As we cannot always provide text with icons, we choose only to represent common actions with icons.

BUTTONS + CONFIRMATION (Steven Hoober's mobile patterns wiki)

WIREFRAME: 5,6,7,8

The 'Button' pattern used in combination with the 'Confirmation' pattern can be observed in the save button displayed at the bottom of the wireframe stated above. When a user presses the button as an indicator that they want to submit an action, the action is evaluated. If found to be valid, the user is prompted with a confirmation screen that conveys the result of submitting the action to confirm they want to submit the action.

The 'Button' pattern is appropriate for this function as we want to make it as convenient and clear as possible to submit an action. This reduces the possibility of user error and barriers to use.

The 'Confirmation' pattern is appropriate for this function as we want to ensure that users are aware of the result of the action they submit. They can proceed with the action if they are confident or return to the form at this stage.

TABS (Steven Hoober's mobile patterns wiki)

WIREFRAME: 2,6,8

The 'Tab' pattern is used to allow users to access relevant screens in a feature. This can be observed in the wireframe stated above to allow users to switch between different ways of adding an assessment outcome, viewing a summary of all grades, and calculating potential grades. The active tab is distinguished by a line and highlighting the tab title.

The 'Tab' pattern is appropriate for this function as we want users to be able to access functions quickly, know that these functions are always available and better organise the components on our screen as users can consume information and take action on the same page without scrolling (e.g. can view existing result when calculating marks - easier to decide the desired grade for the course).

VERTICAL LIST + ON-SCREEN GESTURE (Steven Hoober's mobile patterns wiki)

WIREFRAME: 1

The 'Vertical list' pattern, in combination with the 'On-screen gesture' pattern, is used to display all possible courses users can access. This can be observed in the wireframe stated above. Users can scroll through all courses created by swiping up and down to view how much of the course they have completed and select courses to reveal more information about the course on a separate screen.

The 'Vertical' list pattern is appropriate for this function as we want users to view all possible courses in an easily consumable way. Moreover, as the courses aren't related, and we have only a few components to display (a large proportion of the screen used to present courses), we can use a vertical list without worrying about obscuring information or blocking the desired action by a user.

The 'On-screen gesture' pattern is appropriate to support the navigation of the vertical list. However, as it is inconvenient to press a button or interact with a small scroll bar (screen size limitation) when navigating through a large number of courses, we have decided it is best to use a scrolling gesture to aid users in finding their desired courses. This is implemented using gestures where users swipe up and down.

TITLES (Steven Hoober's mobile patterns wiki)

WIREFRAME: 1,2,3,4,5,6,7,8,9

The 'Titles' pattern is used to indicate the overall function of a screen and assist users in navigating to desired information and action. This is implemented in the wireframes stated above using large and bold text to illustrate a screen's title.

The 'Title' pattern is appropriate for this function as we want users to be confident in exploring our application and reassured that actions taken result in their desired outcome. In addition, by using titles, we provide users with extra information, which decreases barriers and inconvenience created by making our users make assumptions when interacting with functions and features in the solution.

SEARCH-WITHIN (Steven Hoober's mobile patterns wiki)

WIREFRAME: 1,3,9

The 'Search-within' pattern can be observed in the wireframe stated above in the form of a search bar to locate relevant boards easily.

The 'Search-within' pattern is appropriate for this function as we do not want to inconvenience users by making them scroll through an extensive list of irrelevant courses. By allowing users to search for the desired course at the top of the screen, they can explicitly filter out irrelevant courses.

The nature of how the courses are identified also supports the use of the 'Search-within' pattern as the title of the boards are represented by text; the most appropriate filtering method is the 'Search-within' pattern.

INPUT AREA + FORM SELECTION (Steven Hoober's mobile patterns wiki)

WIREFRAME: 4,5,6,8,9

The 'Input area' pattern used in combination with the 'Form Selection' pattern can be observed in the wireframes stated above to input relevant information when creating/selecting an assessment/task item or creating a course. The 'Input area' pattern is specifically used for fields which require custom text or specific values (e.g. name, description, percentage) - used sparingly to limit user error. The 'Form Selection' pattern is used for information with categorical values (e.g. parent assessment, priority level, time (hours)).

The 'Input area' pattern is appropriate for this function as we require custom user input which we cannot predict - using this pattern for custom fields means the object will more closely reflect users' needs.

The 'Form Selection' pattern is appropriate for this function as we can make categorical inputs more convenient for users (using information already inputted), align inputs with other system functions (e.g. Sort and Filter) and reduce the possibility of users inputting incorrect information.

OTHER IMPORTANT PATTERNS: [FILM STRIP + LOCATION WITHIN + ON-SCREEN GESTURE]