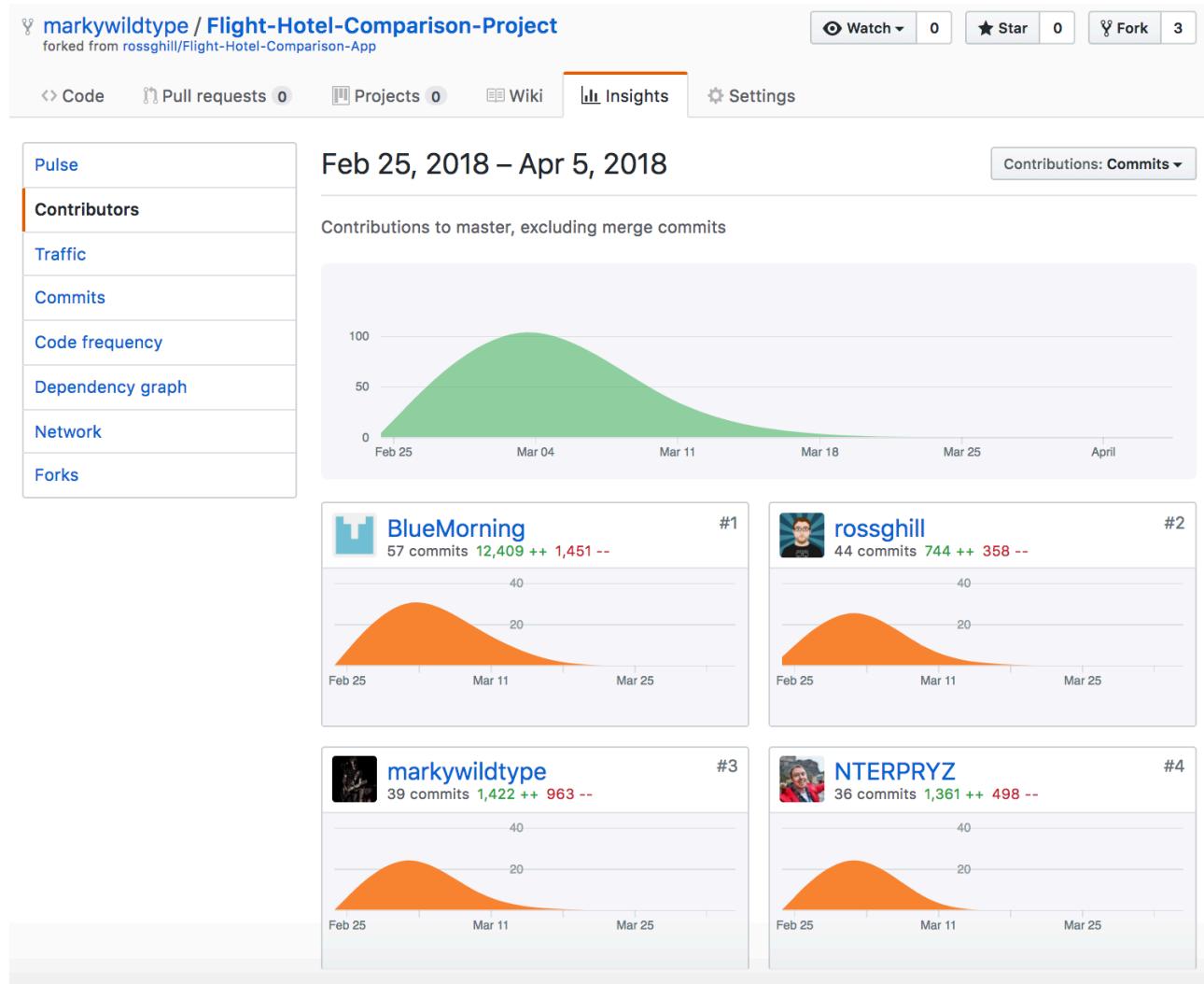


Evidence for Project Unit

Mark Blanford
Cohort E18

P.1 GitHub Contributors Page:



P.2 Project Brief:

Flightr

A new start-up Flightr, has spotted a niche in the Flight + Hotel comparison web app market. It realises it can simultaneously compare flights and hotels to offer the best package possible.

It wants a user to be able to reach our web app and input a departure date and return date and then display possible flight + hotel options. Once searched, it would calculate the available options of flights and represent the data analytically to show travel times or perhaps best value options. It will also display hotel options nearby and a total package price for return flights and hotel nights based on the user input.

They would like the MVP built as soon as possible along with any other features you can implement.

MVP

- User can search by departure and return dates
- User can view available flight and hotel options for their dates
- User can sort flight & hotel packages by price
- User can add packages to a favourites list

P.3 Use of Trello:

Initial and final MoSCoW Trello boards for To-do list project:

https://github.com/markywildtype/to_do_list_project

The Trello board is titled "PrioritEase". It features four columns: "Must have", "Should have", "Could have", and "Won't have (For now)". Each column contains several cards with red titles. A "Personal" and "Private" link is visible at the top left.

Must have	Should have	Could have	Won't have (For now)
ToDoList class	Completion status indicator	Priority (urgent/important)	Drag items from one priority list to another
CRUD Task	Delete Task button	Priority status indicator (colour?)	Complete-by date
CRUD ToDoList	Edit Task button	Multiple list display Activity by priority	Order lists by complete-by date
Launcher Acitivity	Add a card...	Limit characters for basic and detailed task descriptions	Remote database integration - sync when online for sharing lists with others
Create Task Activity			Fragments for displaying list item details
TaskList Activity			Add a card...
Individual Task details Activity			
Mark tasks with completion status			

The screenshot shows a digital Kanban board titled "PrioritEase". It has four columns: "Must have", "Should have", "Could have", and "Won't have (For now)".

- Must have:**
 - Task class
 - ToDoList class
 - CRUD Task
 - Launcher Activity
 - Create Task Activity
 - TaskList Activity
 - Individual Task details Activity
 - Mark tasks with completion status
- Should have:**
 - Completion status indicator
 - Delete Task button
 - Edit Task button
 - Limit characters for basic and detailed task descriptions
- Could have:**
 - Priority (urgent/important)
 - Priority status indicator (colour?)
 - Multiple list display Activity by priority
- Won't have (For now):**
 - Drag items from one priority list to another
 - Complete-by date
 - Order lists by complete-by date
 - Remote database integration - sync when online for sharing lists with others
 - Fragments for displaying list item details
 - Reorder by: priority, completion status, complete-by date

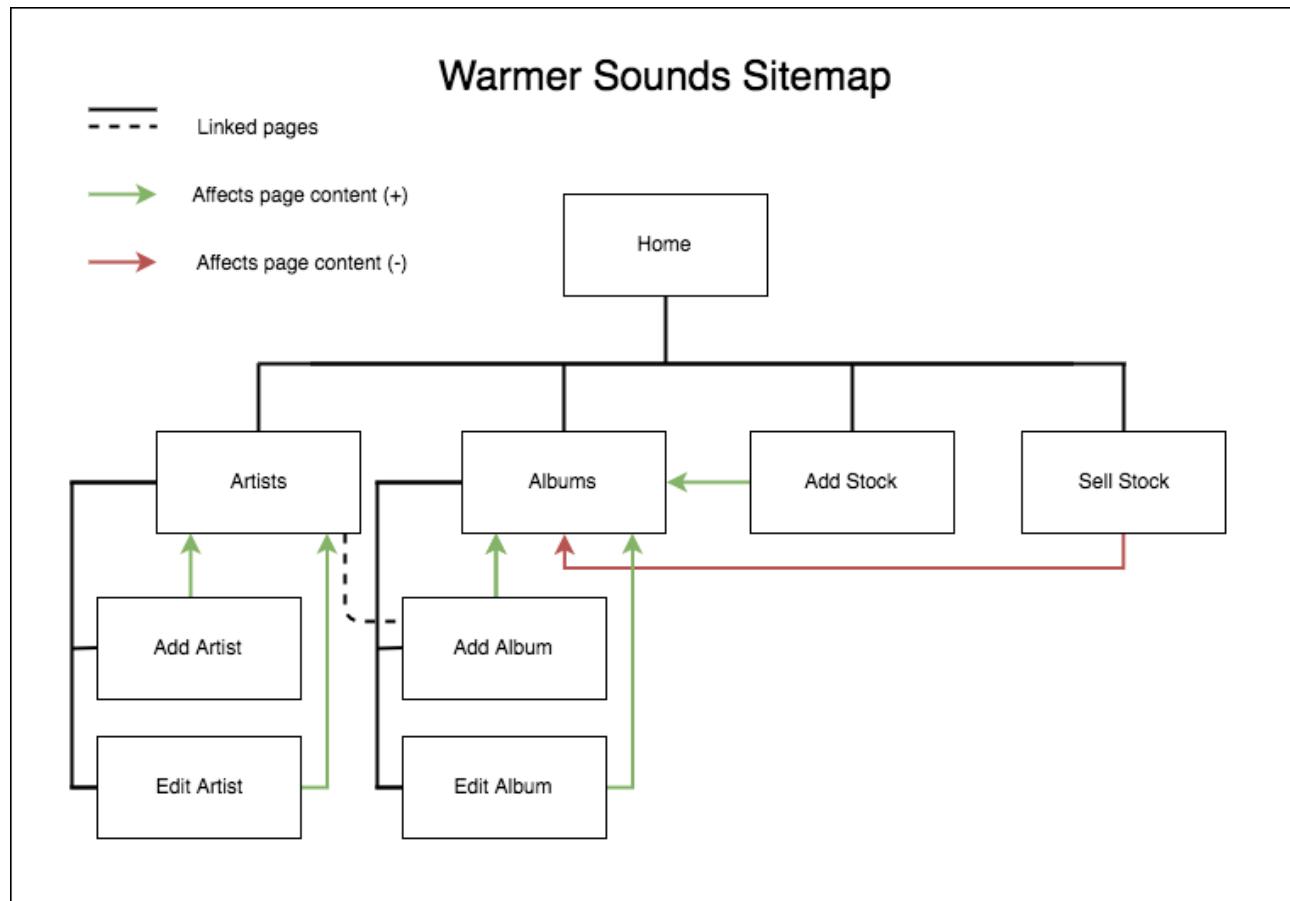
Each card includes a small icon and a "Add a card..." button at the bottom.

P.4 Acceptance Criteria:

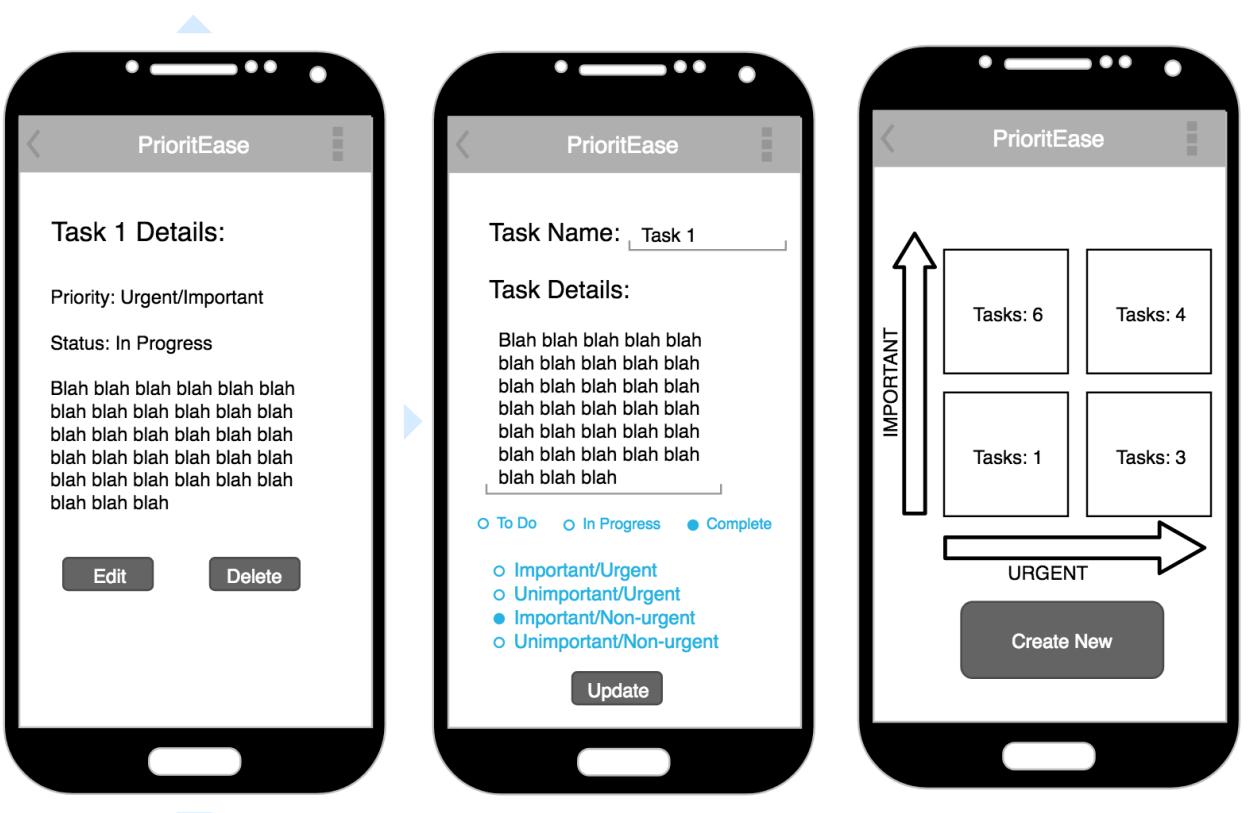
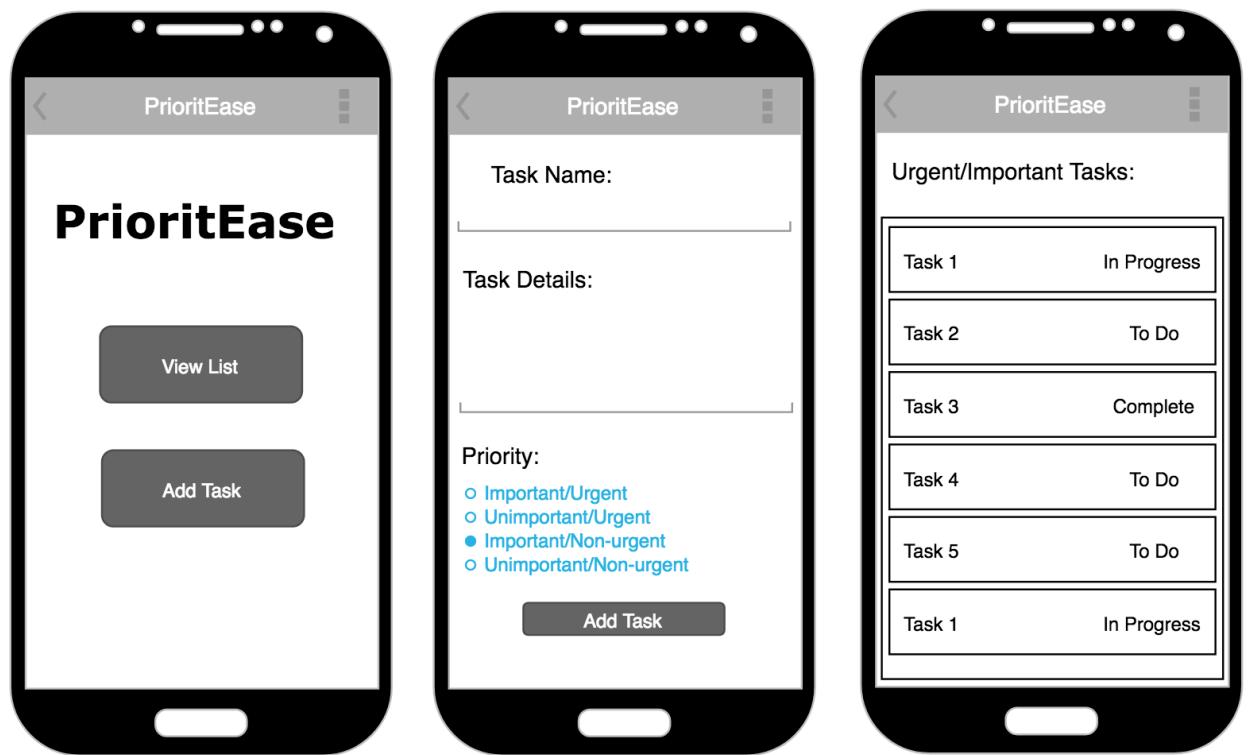
Acceptance Criteria	Expected Result/Output	Pass/Fail
A user can search for flight/hotel packages with search criteria of their choice	A list of packages matching search criteria are displayed	Pass
A user can filter results according to various criteria	Filter options are displayed and when selected, decrease the results accordingly	Pass
A user can toggle a map version of results instead of a list	A google map is displayed showing all the hotel options in graphical format	Pass
The site should be easy to navigate from a user perspective	The page navigation utilises familiar design elements and page structure consistently	Pass
A user can save result packages using a button displayed on the package listing	The package is saved to a packages modal, which can be accessed by an easy to locate favourites button	Pass
A user can delete packages from their favourites	The 'save package' button is replaced by a 'delete' button in the favourites list. Clicking this removes the package from favourites	Pass
The site should indicate that activity is ongoing if the APIs accessed are slow to respond	After the search button is clicked, a modal with a loading gif is presented, closing when results are received and loaded	Pass

P.5 User Sitemap:

User sitemap from Record Shop Stock Tracker Project:
https://github.com/markywildtype/record_shop_project

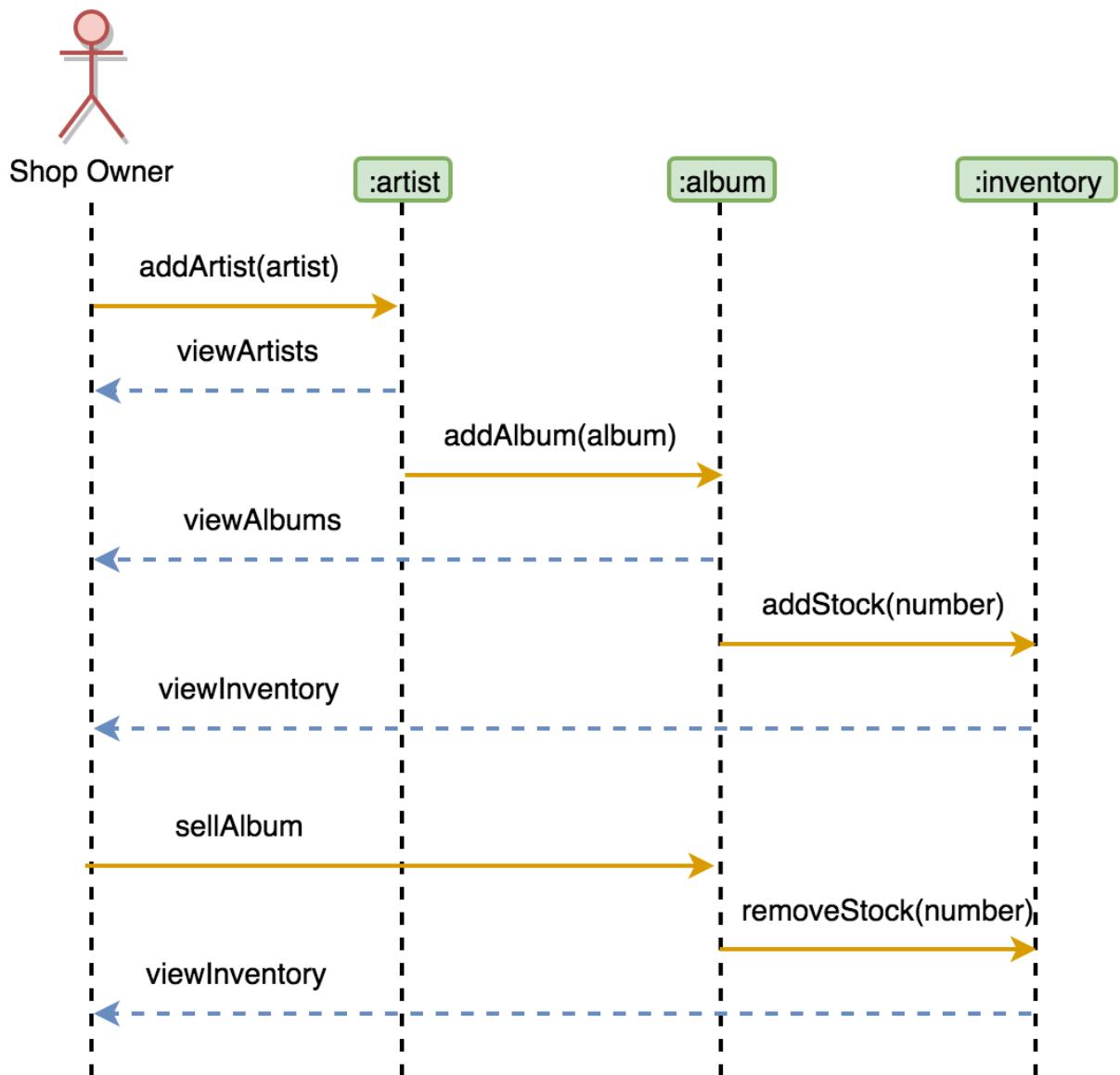


P.6 Wireframe Designs (To-do list project):

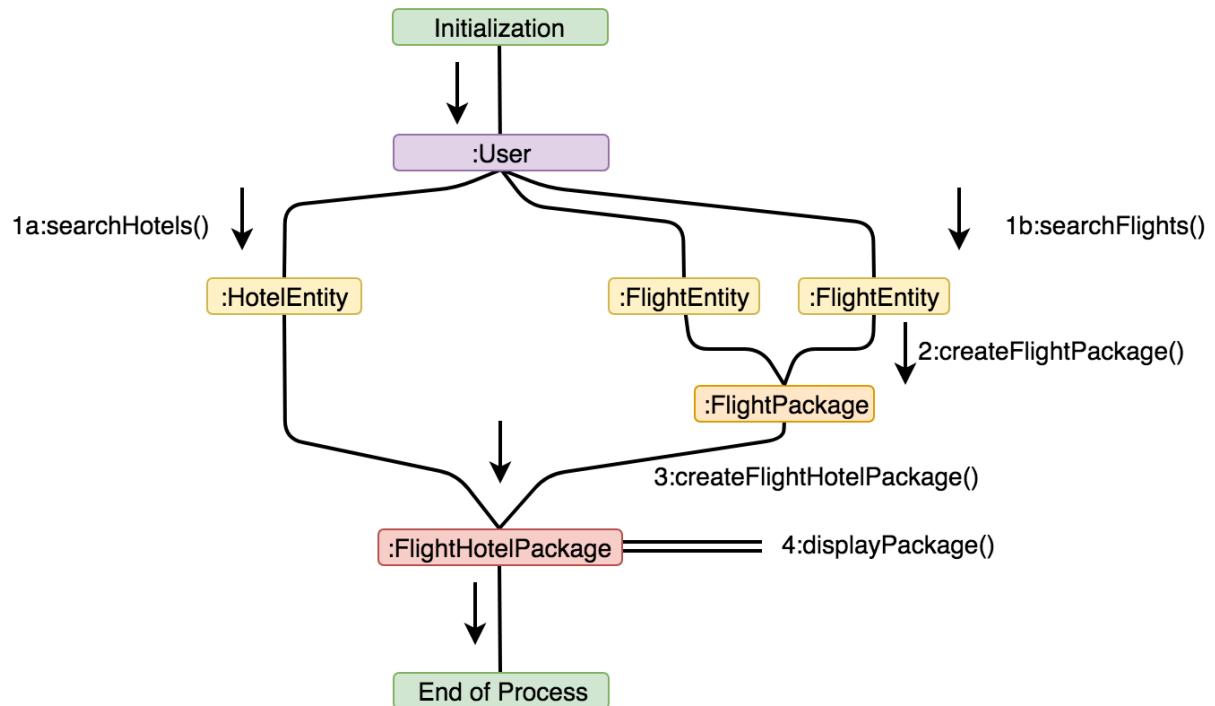


P.7 System Interaction Diagrams:

Sequence diagram (Record Shop Stock Tracker Project):

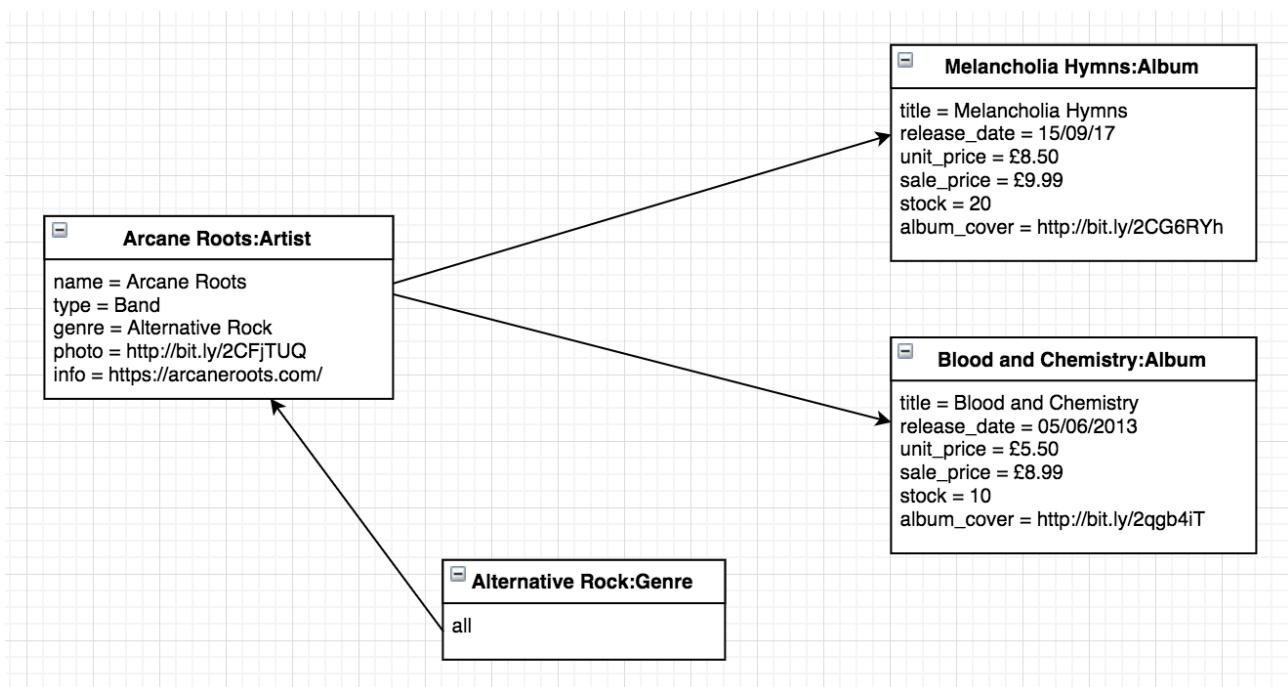


Collaboration diagram (Flight Hotel Comparison Project)

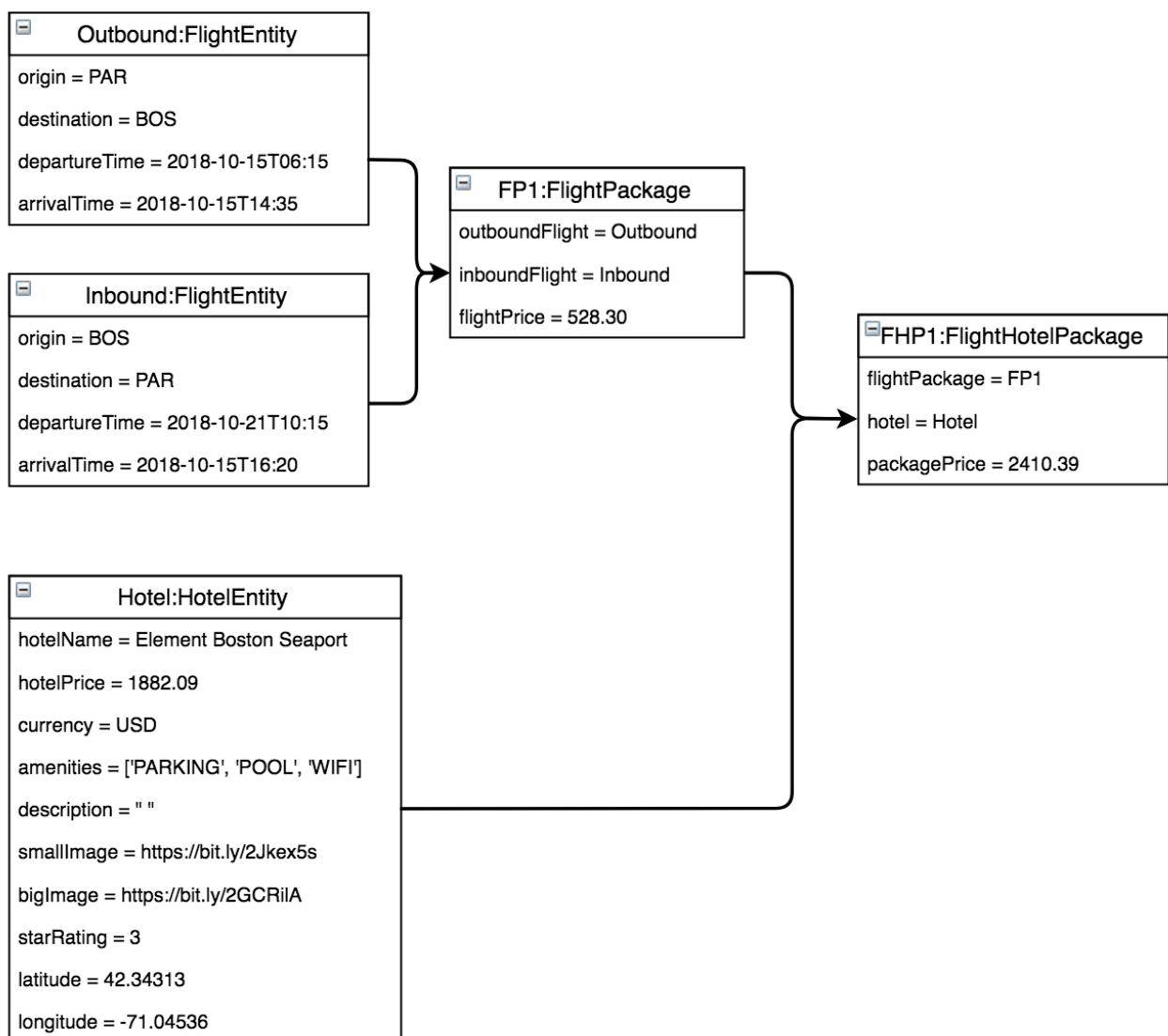


P.8 Object Diagrams:

From Record Shop Stock Tracker Project:



From Flight Hotel Comparison Project:



P.9 Examples of Algorithms:

This is an algorithm to determine the largest and smallest integers in an array. It is looping through the array comparing the first value to each of the other values, reassigning the min/max variables if the value of the current number is smaller/larger than the initial value of those variables respectively. As the array being assessed becomes larger, this algorithm will be less efficient, so it should only be used for smaller arrays:

```
1  const numbers = [99, 108, 564, 2, 77, 236, 1898]
2
3  const findMaxAndMin = function(){
4
5    let min = numbers[0];
6    let max = numbers[0];
7
8    for(const number of numbers){
9      if(min > number){
10        min = number;
11      }
12
13      if (max < number) {
14        max = number;
15      }
16    }
17
18    console.log("Smallest number = " + min + ", Largest number = " + max);
19  }
20
21  findMaxAndMin();
22
```

This algorithm adds together all the integers in the array by using the forEach enumerator to loop through the array and add each value to a 'runningTotal' variable, which starts at 0. Again, it may be less efficient for larger arrays:

```
23 const numbers = [99, 108, 564, 2, 77, 236, 1898]
24
25 const addAllNumbers = function(){
26
27  let runningTotal = 0;
28
29  numbers.forEach(function(number){
30    runningTotal += number;
31  });
32
33  console.log("The sum of all numbers in the array is: " + runningTotal);
34 }
35
36 addAllNumbers();
```

P.10 Example of Pseudocode:

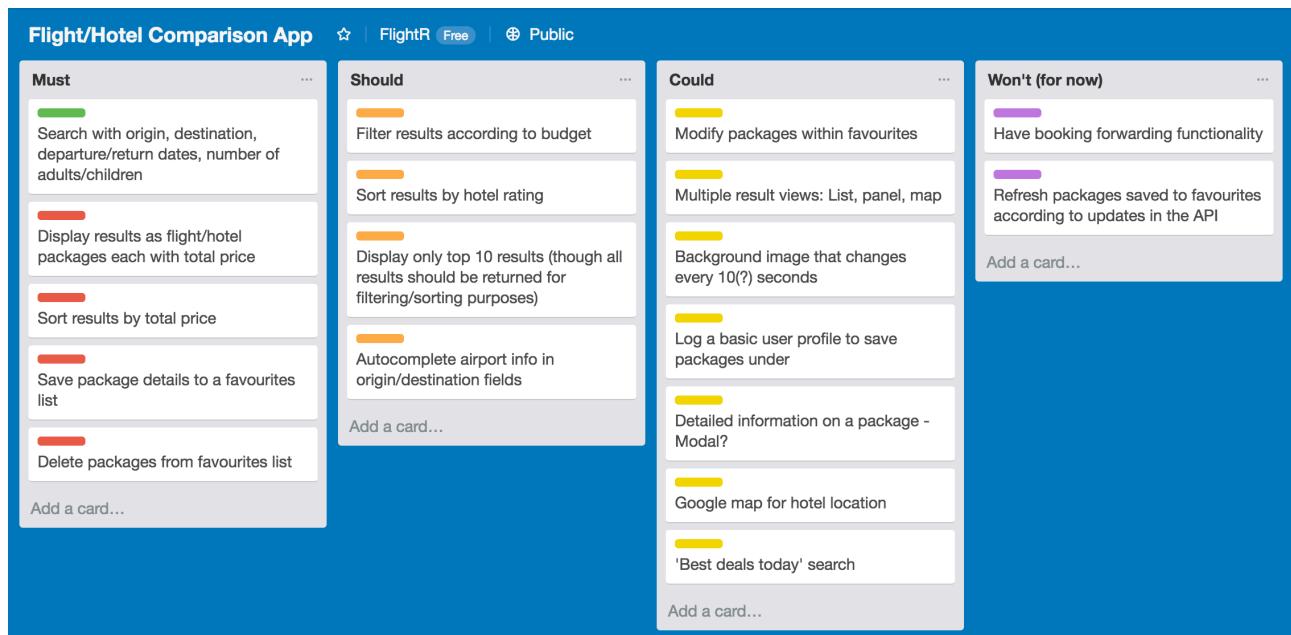
```
3
4 post('/albums/sell_stock') do
5 # find the album to be updated
6 # if the number of copies to be sold is less than current stock:
7 # - decrease the stock level of that album using the sell_stock function
8 # - save the new amount to the database using the album update function
9 # - return to the album information page
10 # if the number of copies exceeds the current stock:
11 # - redirect to an error page without affecting the stock level
12 end
13
```

P.11 GitHub link (To-do list Project):

https://github.com/markywildtype/to_do_list_project

P.12 Project Planning and Development Evidence (Flight Hotel Comparison Project):

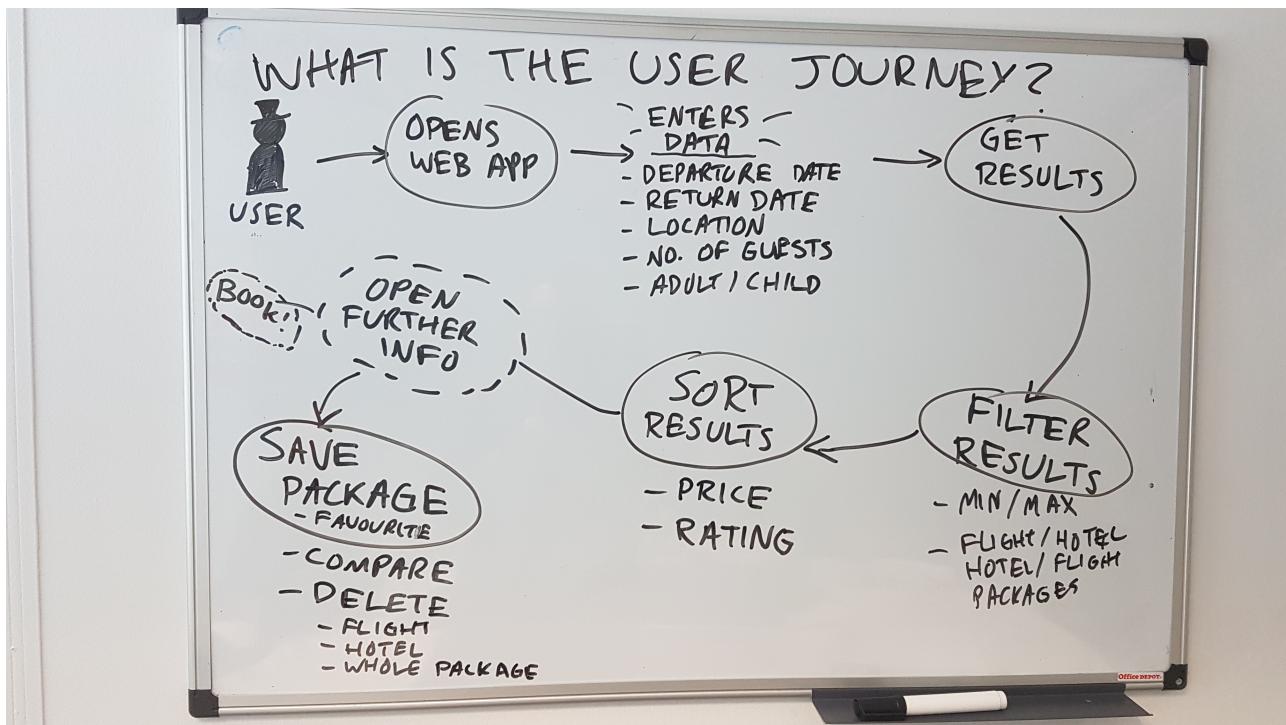
Initial MoSCoW Trello board:



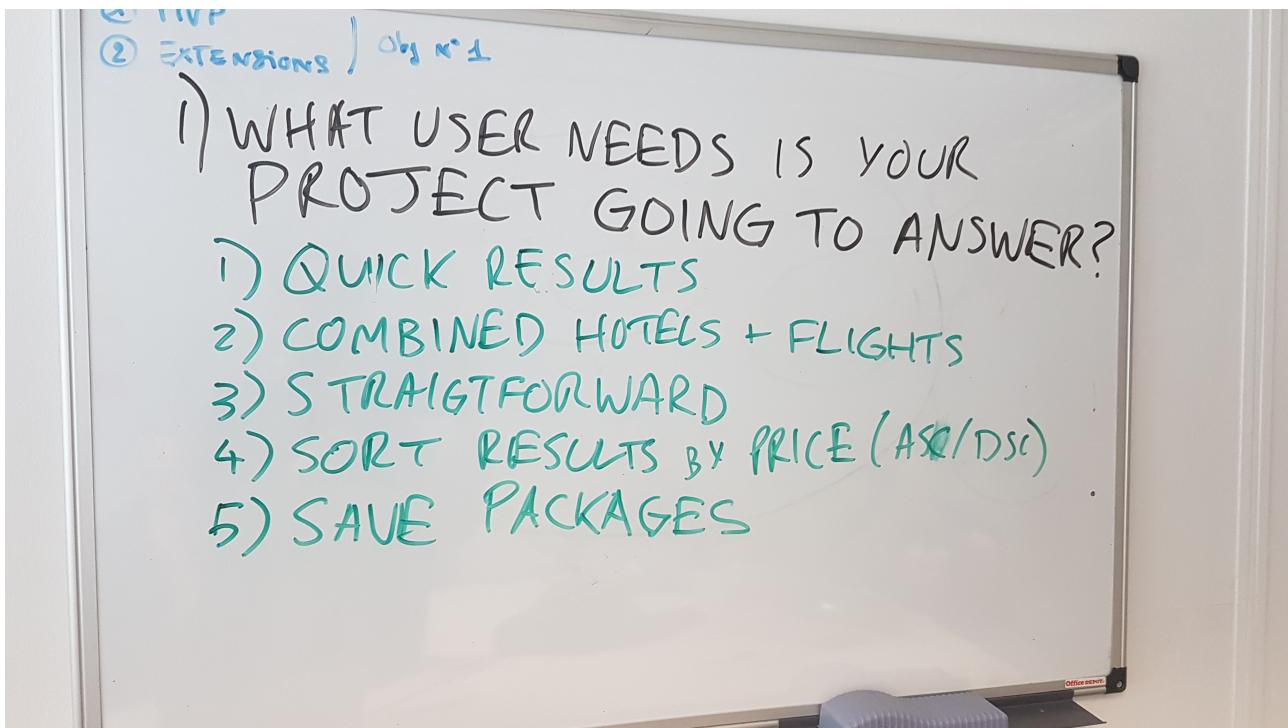
Final MoSCoW Trello board:

Must	Should	Could	Won't (for now)
Search with origin, destination, departure/return dates, number of adults/children	Filter results according to budget	Modify packages within favourites	Have booking forwarding functionality
Display results as flight/hotel packages each with total price	Sort results by hotel rating	Multiple result views: List, panel, map	Refresh packages saved to favourites according to updates in the API
Sort results by total price	Display only top 10 results (though all results should be returned for filtering/sorting purposes)	Background image that changes every 10(?) seconds	Add a card...
Save package details to a favourites list	Autocomplete airport info in origin/destination fields	Log a basic user profile to save packages under	
Delete packages from favourites list	Add a card...	Detailed information on a package - Modal?	
Add a card...		Google map for hotel location	
		'Best deals today' search	
		Add a card...	

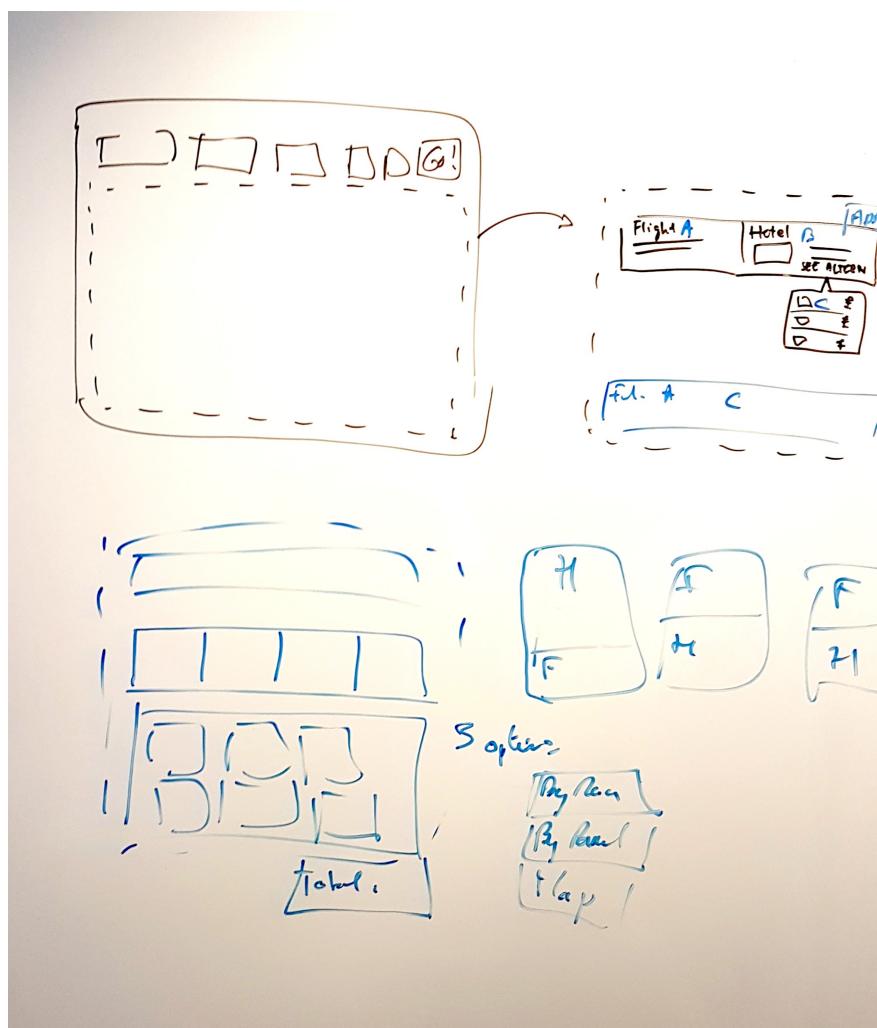
User journey whiteboard sketch:



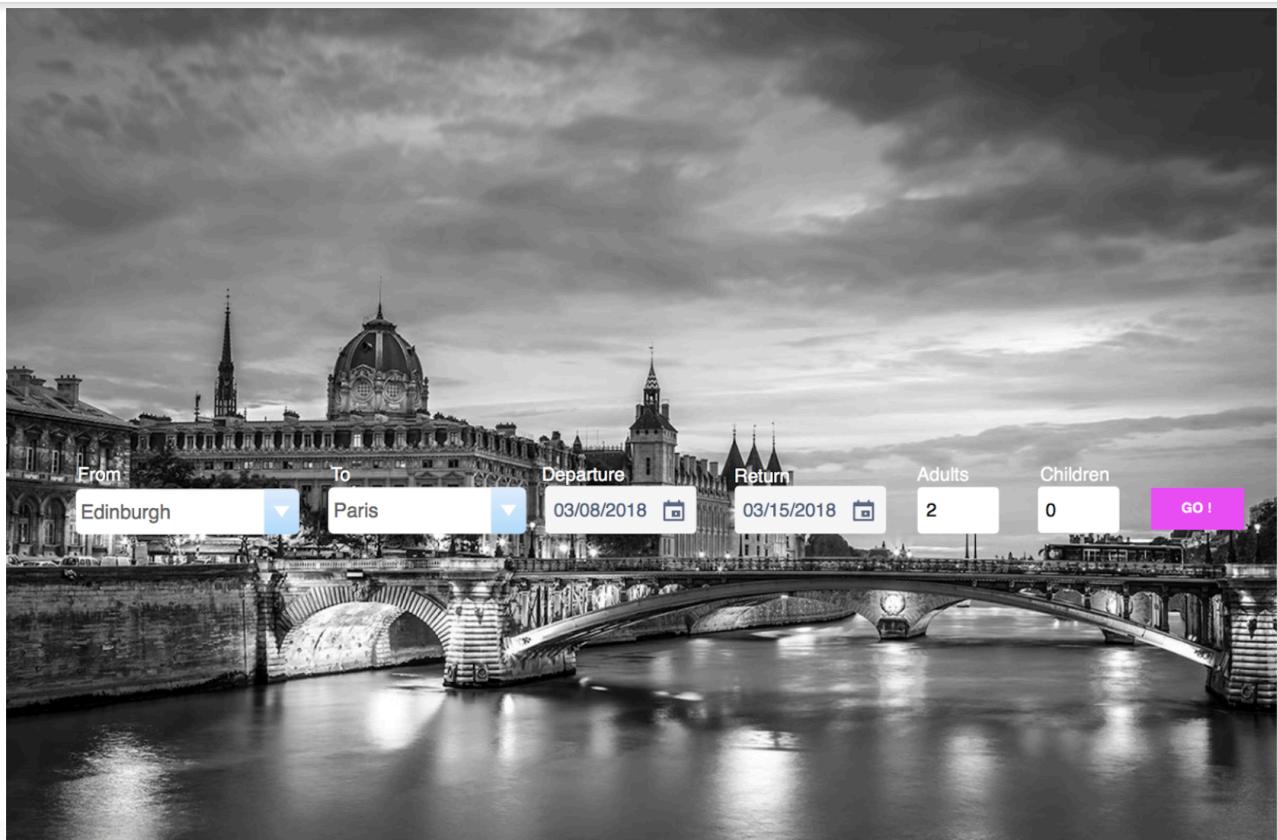
UX planning:



Interface whiteboard sketch:



Interface wireframes:



From: Edinburgh To: Berlin Departure: 08/03/2018 Return: 14/03/2018 Adults: 2 Children: 0 GO!

700 results out of 1256 SHOW ALL Map List SEE FAVOURITES : 0

Sort by: Price - => +

£ 500 **£ 3000**

Stars:

Rating:

Hotel name: Even

Flight Type	Flight Details	Duration	Hotel Name	Rating	Description	Map	Total Cost	Action
Outbound	EDI 07:00 - 08:20 LTN EDINBURGH - LONDON	1h20	Even Hotel	★★★★★	This stunning 4-star boutique hotel, just a few-minute walk from Paddington station, offers elegant rooms with fluffy bathrobes, Egyptian linen and luxurious toiletries.		Total : £ 1000	ADD
Return	LTN 17:00 - 18:20 EDI LONDON - EDINBURGH	1h20						
Outbound	EDI 07:00 - 08:20 LTN EDINBURGH - LONDON	1h20	Even Hotel	★★★★★	This stunning 4-star boutique hotel, just a few-minute walk from Paddington station, offers elegant rooms with fluffy bathrobes, Egyptian linen and luxurious toiletries.		Total : £ 800	ADD
Return	LTN 17:00 - 18:20 EDI LONDON - EDINBURGH	1h20						
Outbound	EDI 07:00 - 08:20 LTN EDINBURGH - LONDON	1h20	Even Hotel	★★★★★	This stunning 4-star boutique hotel, just a few-minute walk from Paddington station, offers elegant rooms with fluffy bathrobes, Egyptian linen and luxurious toiletries.		Total : £ 700	ADD
Return	LTN 17:00 - 18:20 EDI LONDON - EDINBURGH	1h20						

From: Edinburgh To: Berlin Departure: 08/03/2018 Return: 14/03/2018 Adults: 2 Children: 0 GO!

700 results out of 1256 SHOW ALL Map List SEE FAVOURITES : 4

Sort by: Price - => +

£ 500 £ 3000

Stars: 5 stars 4 stars 3 stars 2 stars 1 star

Rating:

Hotel name: Even

Even Hotel ****
This stunning 4-star boutique hotel, just a few-minute walk from Paddington station, offers elegant rooms with fluffy bathrobes, Egyptian linen and luxurious toiletries.

Outbound: EDI 07:00 - 08:20 LTN 1h20 EDINBURGH - LONDON
Return: LTN 17:00 - 18:20 EDI 1h20 LONDON - EDINBURGH

Total: £ 700 ADD

P.13 Example of User Input (Record Shop Stock Tracker Project):

'Add Artist' button clicked:

HOME

ARTISTS

ALBUMS

ADD STOCK

SALES

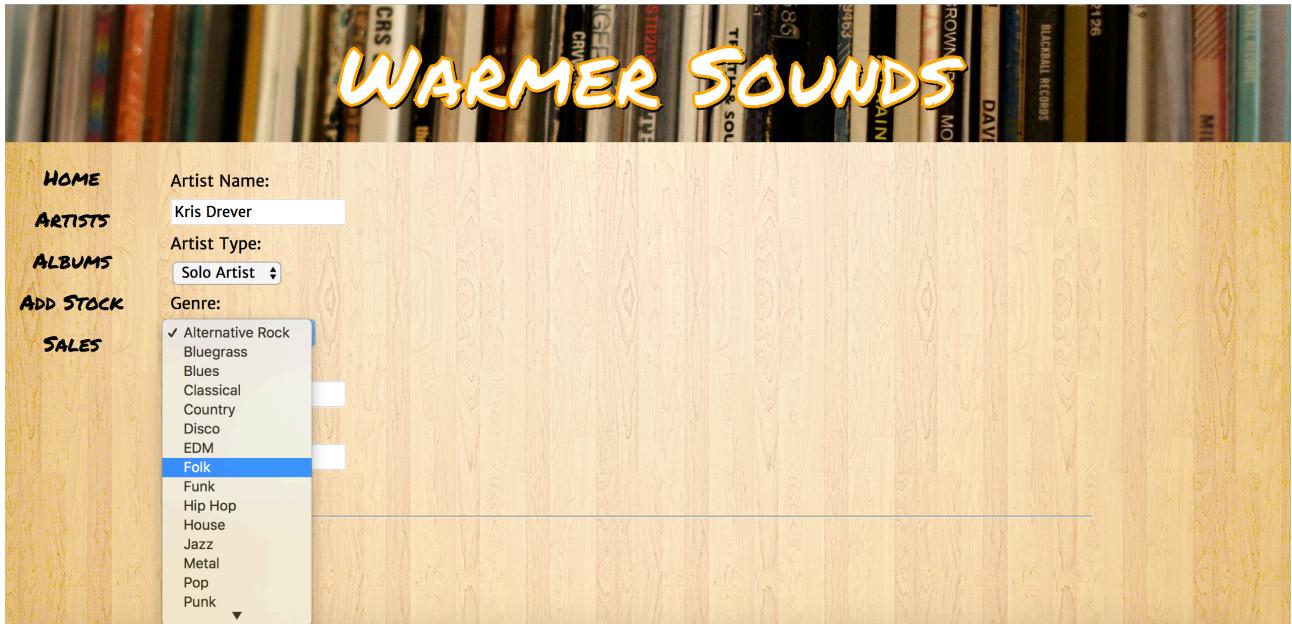
WARMER SOUNDS

ALL ARTISTS:

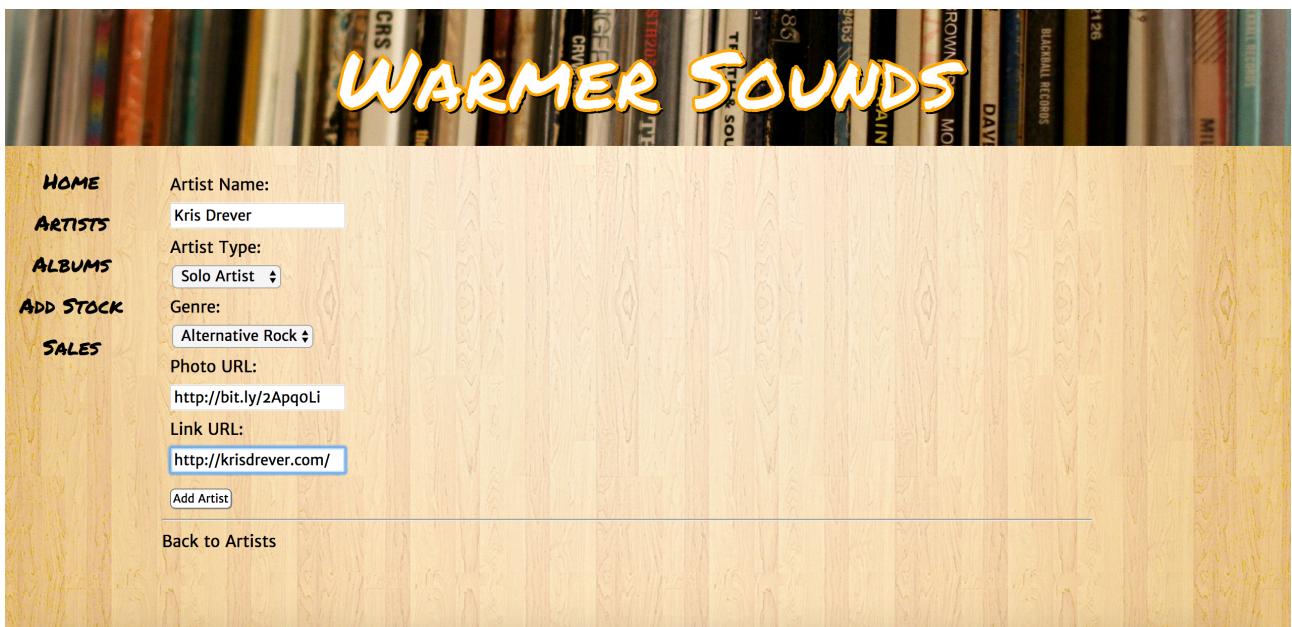
- Arcane Roots || Albums: (2)
 - Add Album
 - Edit Artist
 - Delete Artist
- Biffy Clyro || Albums: (1)
 - Add Album
 - Edit Artist
 - Delete Artist
- Big Thief || Albums: (1)
 - Add Album
 - Edit Artist
 - Delete Artist
- Black Peaks || Albums: (1)
 - Add Album
 - Edit Artist
 - Delete Artist
- Rachel Sermanni || Albums: (2)
 - Add Album
 - Edit Artist
 - Delete Artist

localhost:4567/artists/new

Artist information added:



Further information added, 'Add Artist' button clicked:



Artist information page created from user-input information:

The screenshot shows a website for "Warmer Sounds". The header features the word "WARMER SOUNDS" in large, yellow, stylized letters against a background of book spines. On the left, a vertical navigation menu lists "HOME", "ARTISTS", "ALBUMS", "ADD STOCK", and "SALES". Below the menu is a black and white portrait of a man with long hair and a beard. To the right of the portrait, the name "Kris Drever" is displayed. Below the name are several details: "Type: Solo Artist", "Genre: Alternative Rock", a link to "More information", and "Albums: (0)". At the bottom of this section are three buttons: "Add Album", "Edit Artist", and "Delete Artist". A "Back to Artists" link is located at the very bottom of the page.

P.14 Interaction with Data Persistence:

Artist selected from database-generated list of saved artists:

The screenshot shows a "ADD STOCK" page for "Warmer Sounds". The header and navigation bar are identical to the artist information page. The main form has fields for "Artist:" (with a dropdown menu listing "Arcane Roots", "Biffy Clyro", "Big Thief", "Black Peaks", "Kris Drever" which is highlighted with a blue selection bar), "Stock:", "Unit Price:", "Sale Price:", and "Album Cover URL:". At the bottom of the form is a single button labeled "Add Album".

Album information added and saved for the selected artist:

HOME Artist: Kris Drever

ARTISTS Album Title: If Wishes Were Horses

ALBUMS

ADD STOCK

SALES Release Date: 25/03/2106

Stock: 7

Unit Price: 5.00

Sale Price: 7.99

Album Cover URL: <http://bit.ly/2CC6tMN>

[Back to Albums](#)

Site design by Mark Blanford © 2017

Artist/album displayed on database-generated inventory page:

ARTISTS **ALL ALBUMS:**

	Artist	Album	Stock Level	Actions		
ALBUMS	Arcane Roots	Blood and Chemistry	Low	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>
ADD STOCK	Arcane Roots	Melancholia Hymns	High	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>
SALES	Biffy Clyro	Only Revolutions	Low	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>
	Big Thief	Masterpiece	Medium	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>
	Black Peaks	Statues	High	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>
	Kris Drever	If Wishes Were Horses	Medium	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>
	Rachel Sermanni	Under Mountains	Low	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>
	Rachel Sermanni	Tied to the Moon	High	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>
	Terrestria	Things You'll Never Know	High	<input type="button" value="View Details"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Album"/>

Site design by Mark Blanford © 2017

Individual album details as persisted in the database, accessed from inventory page by clicking 'View Details' button:

The screenshot shows a sidebar with navigation links: HOME, ARTISTS, ALBUMS, ADD STOCK, and SALES. The main content area is titled 'ALBUM DETAILS:' and displays the cover art for 'Kris Drever IF WISHES WERE HORSES'. Below the cover, the following information is listed:
Artist: Kris Drever
Album: If Wishes Were Horses
Release Date: 2106-03-25
Unit Price/Sale Price: £5.0/£7.99
Stock: 7
Current Inventory Profit Margin: £20.93
At the bottom are two buttons: 'Edit Album' and 'Delete Album'.

P.15 User Output Result:

User clicks 'Edit Album' button:

The screenshot shows a sidebar with navigation links: HOME, ARTISTS, ALBUMS, ADD STOCK, and SALES. The main content area is titled 'EDIT ALBUM DETAILS' and contains the following form fields:
Artist Name: Kris Drever
Album Title: If Wishes Were Horses
Release Date: 25/03/2106
Stock: 7
Sale Price: 7.99
Album Cover URL: <http://bit.ly/2CC6tMN>
At the bottom are two buttons: 'Save Changes' and 'Back to Albums'.
A footer at the bottom of the page reads 'Site design by Mark Blanford © 2017'.

User amends Sale Price and Stock level of album:

The screenshot shows a web application interface for managing album details. On the left is a vertical navigation menu with links: HOME, ARTISTS, ALBUMS, ADD STOCK, and SALES. The main area is titled "EDIT ALBUM DETAILS". It contains the following form fields:

- Artist Name: Kris Drever
- Album Title: If Wishes Were Horses
- Release Date: 25/03/2106
- Stock: 15
- Sale Price: 6.99
- Album Cover URL: <http://bit.ly/2CC6tMN>

A "Save Changes" button is at the bottom, and a "Back to Albums" link is below it. The background features a wooden panel texture and a row of book spines.

User saves changes and returns to Album Details view, where the updated information is displayed:

The screenshot shows the "ALBUM DETAILS" page after changes were saved. The left navigation menu is identical to the previous screen. The main content area displays the following information:

ALBUM DETAILS:

KRIS DREVER
IF WISHES WERE HORSES

Artist: Kris Drever
Album: If Wishes Were Horses
Release Date: 2106-03-25
Unit Price/Sale Price: £5.0/£6.99
Stock: 15
Current Inventory Profit Margin: £29.85

Buttons for "Edit Album" and "Delete Album" are at the bottom.

The updated information is also referenced on the main inventory display page ('Stock Level' indicator has changed from 'Medium' (<10) to 'High' (>10)):

ARTISTS	ALL ARTISTS					
ALBUMS	Artist	Album	Stock Level	Actions		
ADD STOCK	Arcane Roots	Blood and Chemistry	Low	View Details	Update	Delete Album
SALES	Arcane Roots	Melancholia Hymns	High	View Details	Update	Delete Album
	Biffy Clyro	Only Revolutions	Low	View Details	Update	Delete Album
	Big Thief	Masterpiece	Medium	View Details	Update	Delete Album
	Black Peaks	Statues	High	View Details	Update	Delete Album
	Kris Drever	If Wishes Were Horses	High	View Details	Update	Delete Album
	Rachel Sermanni	Under Mountains	Low	View Details	Update	Delete Album
	Rachel Sermanni	Tied to the Moon	High	View Details	Update	Delete Album
	Terrestria	Things You'll Never Know	High	View Details	Update	Delete Album

Site design by Mark Blanford © 2017

P.16 Bug Tracking Report (Flight Hotel Comparison Project):

Bug Description	Status	Bug Fix	Updated Status
Result loading from multiple API calls is slow	FAIL	Parallel calls to the different APIs instead of sequential, triggering request completion booleans when package display becomes possible	PASS
No activity onscreen while waiting for API response/package creation	FAIL	Animation modal added to indicate loading and discourage user action	PASS
Loading modal disappears too soon or too early	FAIL	Modal visibility lifespan linked to API request completion booleans	PASS
Departure date cannot be in the past	FAIL	Add current date validation to date picker	IN PROGRESS
Return date cannot be before the departure date	FAIL	Departure date validation to return date picker	IN PROGRESS
App is responsive to different screen sizes	FAIL	CSS reconfiguration to improve responsiveness	IN PROGRESS
Deleting a favourite doesn't automatically remove it from the favourites list	FAIL	Deleting a favourite triggers a re-render of the favourites list	PASS

P.17 Evidence of Testing:

Test code:

The screenshot shows a code editor with a sidebar labeled "Project". The project structure includes a folder "heroes_and_rats" containing ".git", "node_modules", and a "specs" folder which contains "hero_spec.js" and "task_spec.js". The "task_spec.js" file is open in the main editor area. The code in "task_spec.js" is as follows:

```
1 const assert = require('assert');
2 const Task = require('../task.js');
3
4 describe('Task', function(){
5
6     let task1;
7
8     beforeEach(function(){
9         task1 = new Task('Prove the existence of aliens', 10, 2, 'I Want To Believe poster');
10    });
11
12    it('should have a description', function(){
13        const actual = task1.description;
14        assert.strictEqual(actual, 'Prove the existence of aliens');
15    });
16
17    it('should have a difficulty level', function(){
18        const actual = task1.difficulty;
19        assert.strictEqual(actual, 10);
20    });
21
22    it('should have an urgency level', function(){
23        const actual = task1.urgency;
24        assert.strictEqual(actual, 2);
25    });
26
27    it('should have a reward', function(){
28        const actual = task1.reward;
29        assert.strictEqual(actual, 'I Want To Believe poster')
30    })
31
32
33 })
```

Failing test:

The screenshot shows a terminal window with the command "node task_spec.js" run. The output shows the test results:

```
Task
  ✓ should have a description
  ✓ should have a difficulty level
  ✓ should have an urgency level
  1) should have a reward

  8 passing (10ms)
  1 failing

  1) Task
      should have a reward:
      AssertionError [ERR_ASSERTION]: undefined
      ed === 'I Want To Believe poster'
          at Context.<anonymous> (specs/task_spec.js:29:12)

npm ERR! Test failed. See above for more details
```

Failing code:

```
task_spec.js
```

```
task.js
```

```
1 const Task = function(description, difficulty, urgency){  
2     this.description = description;  
3     this.difficulty = difficulty;  
4     this.urgency = urgency;  
5 }  
6  
7 module.exports = Task;  
8
```

Corrected code:

```
task_spec.js
```

```
task.js
```

```
1 const Task = function(description, difficulty, urgency, reward){  
2     this.description = description;  
3     this.difficulty = difficulty;  
4     this.urgency = urgency;  
5     this.reward = reward;  
6 }  
7  
8 module.exports = Task;  
9 |
```

Passing test:

```
Task  
  ✓ should have a description  
  ✓ should have a difficulty level  
  ✓ should have an urgency level  
  ✓ should have a reward  
  
  9 passing (7ms)  
  
→ heroes_and_rats git:(master) ✘
```