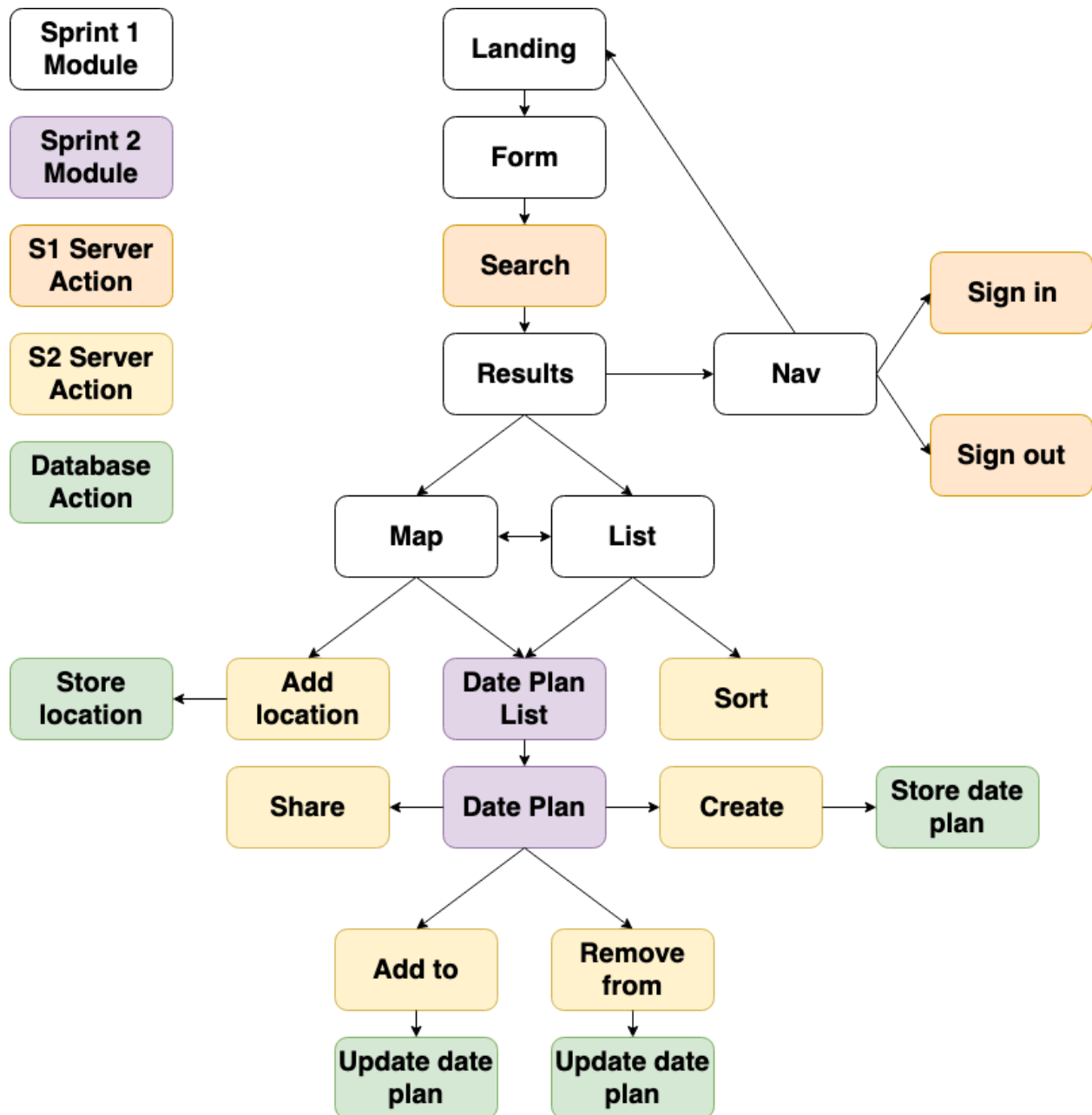


Incremental and Regression Testing - Team 2

Module Map



Modules

Landing Module

The Landing module is just a simple landing page for the application.

Inputs: None

Outputs: Page with button to navigate to Form module

Dependencies: Form, Nav

Form Module

The Form module is a form where the user fills out the information that they want to use for their search query and submits it to the search algorithm.

Inputs: Location, Price Range, Date Type, Min Time, Max Time

Outputs: Page with a form, Object that is sent to Search module

Dependencies: Form, Search

Search Module

The Search module is our main backend algorithm that takes input from the Form module and performs a custom search using the information provided.

Inputs: Object from the Form module

Outputs: Array that contains locations that match the search parameters

Dependencies: Form, Results

Results Module

The Results module is a simple container for the Nav module and either the Map module or the List module.

Inputs: Array from the Search module

Outputs: Page with Nav module and either Map or List module

Dependencies: Search, Nav, Map, List

Nav Module

The Nav module is a navbar that contains a link to the Landing module, tabs to switch between the Map and List modules, and account login/logout functionality.

Inputs: None

Outputs: Navbar with link to Landing, tabs for Map and List, button for logging in and out

Dependencies: Results, Landing

Map Module

The Map module is a Google maps component that shows users the locations and information of the results from the Search module.

Inputs: None

Outputs: Map with pins, Info popups when pin is clicked

Dependencies: Results, List, Date Plan List

List Module

The List module is an alternate view to the Map module that just shows information in a simple list instead.

Inputs: None

Outputs: List with cards for each result, Information on each card about the result

Dependencies: Results, Map, Date Plan List

Date Plan List Module

The Date Plan List module is a container that holds a list of all date plans that a user has created.

Inputs: Date Plans from account

Outputs: Page with list of Date Plans

Dependencies: Map, List, Date Plan

Date Plan

The Date Plan module is an individual date plan that a user has created and can update or delete.

Inputs: Date Plan from account

Outputs: Card with information of the date plan

Dependencies: Date Plan List

Form of Incremental Testing

We followed top-down testing for this project because it made the most sense for our development cycle. We wanted to test everything as we made it, and it made the most sense for us to start developing at the top of the application.

Incremental Testing and Regression Testing

ANY MODULES NOT INCLUDED WERE COMPLETED IN THE LAST SPRINT

Module	Map - TESTS FROM LAST SPRINT AND CURRENT		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Submitting a crowdsourced event would not save the event	1	Changed what data was sent from the frontend of the application to the backend endpoint
2	Movies were not being displayed on the frontend.	2	Changed the post request to return movie data if there is a movie theatre.
3	Searching in a location that had crowdsourced events would not show them	2	Changed the getVenues API to properly flatten the rows returned from the database
4	Crowdsourced events and default events were searched with different distance values	1	Changed the SQL database handler to convert degrees of latitude and longitude to radians before querying
Regression Testing			
Defect #	Description	Severity	How to correct
1	Map was not centered around the location that the user searched	2	Added logic that centered the map on the location that the user searched for

2	User could mess up flow of crowdsourcing submission	2	Added logic that disables the button so that a user is required to click a location on the map
3	Crowdsourced location was not saved	1	Corrected the algorithm that sends data from the frontend to match what was being expected on the backend
4	International Showtimes API would return movies and cinemas without ids or titles and would cause the program to crash	1	Added logic to existing tests to see if any fields are undefined after being processed

Module	List - TESTS FROM LAST SPRINT AND CURRENT		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	A feature to sort by price was added	1	Add a test to check for the selected sort and check the order of the cards
Regression Testing			
Defect #	Description	Severity	How to correct
1	Currently there is no indication that there are no results in the list	2	Add a simple message that says there were no results
2	Currently there is no error checking to see if the reviews exist	2	Add logic to check if the reviews are undefined

3	Currently there is no error checking to see if the contact information exist	2	Add logic to check if the contact information is undefined
4	Sorting by price level was broken when no price was given	2	Add a check in the comparator function for the sorting

Module	Date Plan List - TESTS FROM CURRENT SPRINT		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Automated testing was impossible with the previous file structure	1	Rewrote the component structure so that all of the fetch requests were done in a container component and then passed to the Plans component
2	Clicking on the share button didn't work as expected	2	Rewrote the logic of the share button to change the implementation so that the link gets copied to the clipboard
3	Requesting a user's dateplan ids did not show all of their plans when requesting from the frontend	1	Changed the location where user data is retrieved from the api. Req.user to req.user.sub
4	Sometimes not all plans rendered for a user	1	Write a check to ensure that the proper number of plans are rendered on the front end

Regression Testing			
Defect #	Description	Severity	How to correct
1	Share button copied wrong link	1	Changed the logic so that the share button copied the correct link instead of an undefined link

Module	Date Plan - TESTS FROM CURRENT SPRINT		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	User was unable to navigate to the date plan details page	1	Changed the link structure to use query strings instead of route parameters
2	Clicking on the details button from the list page did not display correct information	1	Fixed the link on the list page so that it routed the user to the correct page using the new query strings method
3	Clicking start dateplan1 button did not open the modal to input name	1	Add a toggle method to enable the modal
4	View plan was clickable when a name for dateplan had not been created yet	2	Disabled the view dateplan modal and enable it when there is a non empty dateplan name

5	An empty dateplan name could be entered and accepted	1	Added a constraint that the modal would only close if the input has a length > 1 when clicking create.
6	Adding activities to dateplan were not viewable	1	Added a bulleted list to the view plan modal that displays the activities the user added
7	Dateplan buttons were visible when user has not signed in	1	Hid buttons until user is logged in and authenticated
8	Null values were not accepted in cases when they should have been upon inserting into the db	1	Changed logic to let certain null value fields pass the check
9	Activity insertion usually failed because some phone numbers were differently formatted than others	1	Added library to validate phone numbers in many formats before insertion.
Regression Testing			
Defect #	Description	Severity	How to correct
1	The page crashed when the user tried to view the details of a date plan with an invalid id	1	Wrote logic so that if the user sends a request for an invalid id, they are shown a 404 page instead of crashing

2	The page displayed an inaccurate rating if the owner of the date plan had not rated the plan yet	2	Wrote logic that checked if the date rating was null, and if it was displayed a special message instead
3	If the venue name had an apostrophe in the title it would not be inserted into the db	2	Wrote logic that delimits apostrophes before inserting into the db.