

Lendr

Team 22: Aditya Dhingra, Rahul Pai, Yash Shiroya, Devaunsh
Sambhav, Shreyaansh Bassi

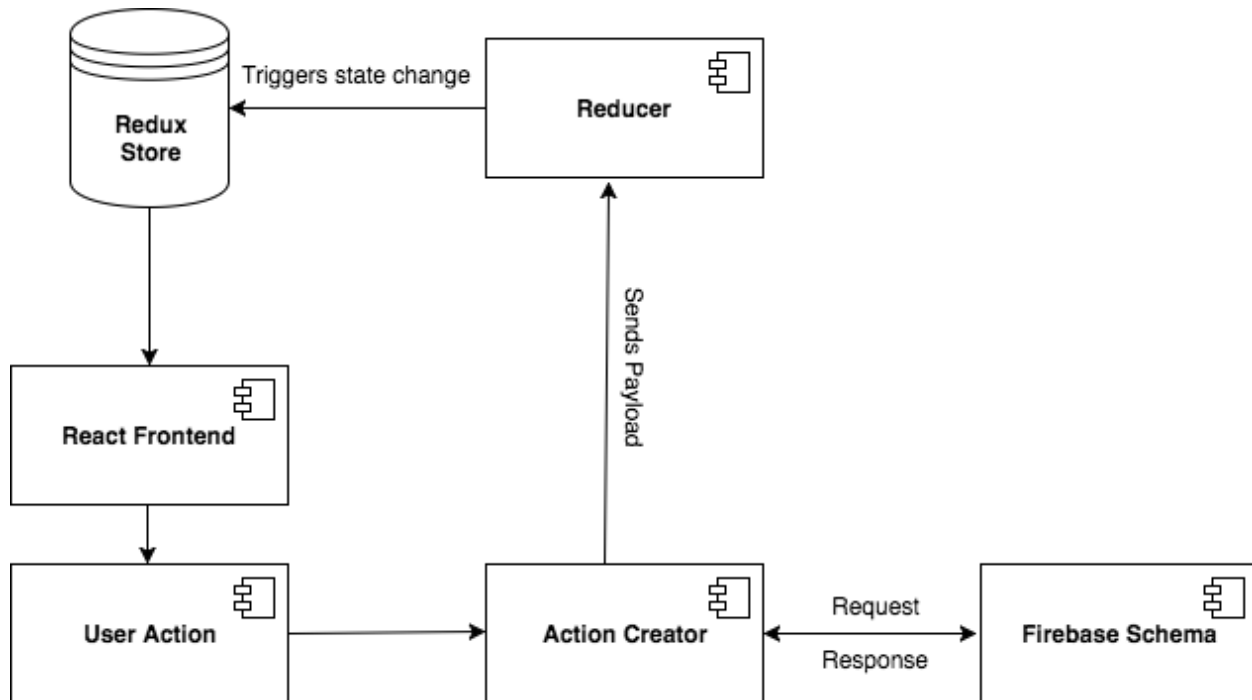
Testing

Contents

1. Components	3
1.1 Component diagram	3
1.2 Type of Incremental Testing	3
1.3 Component Definitions	4
2. Incremental and Regression testing	5
2.1 Automation	5
2.2 Defects Log	5
3. Product Backlog (Updated)	10

1 Components

1.1 Component Diagram



1.2 Type of Incremental testing

Team Lendr decided to follow the Top-Down methodology for testing. We decided to test the main module and then move on to the sub modules. Sub-modules which weren't implemented had stubs in their place to check if they integrated correctly with the main module. Using the top-down testing, we were able to find a lot of problems that occurred while integrating the modules with the "stubs" we defined in place which would've been tougher to figure out once we had actually implemented the submodules and tried to integrate them into the main module. By the time, we got to changing the "stubs" to modules we didn't encounter more than a couple of bugs and the integration went smoothly.

1.3 Component Definitions

Component	Input	Output	Dependent Components
Component A: Database	Items users want to post are saved in the database along with the status of the item.	Users looking at the website see all the items posted by the users retrieved by the database.	{Action Creator}
Component B: Redux Store	Actions sent by the action creator and the previous state of the application passed as a JavaScript object.	Mutated state as returned by the 'reducer' is stored here and is accessible via this.props using ' <i>connect</i> ' function of 'react-redux'	{Reducer, React Frontend}
Component C: React Frontend	Users interacting with the front-end, clicking on different tabs and entering information for posting a new item.	A visible change in the UI <i>reacting</i> to user inputs and other factors like other users posting items on the application.	{Redux Store, User Action}
Component D: User Action	Details of the item a user wants to post or buy as a JavaScript object.	The action calls the Action Creator.	{React Frontend, Action Creator}
Component E: Action Creator	The JavaScript object received when a user action is called.	Appropriate action is created, input for the action is set and the action once created is sent to the redux dispatcher.	{Database, User Action, Reducer}
Component F: Reducer	Takes in the action received in a format set by the action creator and sent by the redux dispatcher.	Performs the appropriate action by making the required database calls or state changes in the redux store.	{Redux Store, Action Creator}

2 Incremental and Regression Testing

2.1 Automation

Team Lendr automated most of the Incremental and Regression tests using the Jest testing framework. This allows us to test various components such as Reducers and Action Creators as separate components. Since all redux actions and reducers are plain JS functions, they can be called directly inside Jest which uses Jasmine under the core, further allowing us to automate the tests using helpers such as 'describe', 'expect' with mocked objects and inputs.

'Describe' allows us to create separate and comprehensive test suites that group similar tests to be tested at the same time.

'Expect' allows us to assert various outputs of the reducers and actions against expected outputs.

2.2 Defects log

Module	Component A - Database
---------------	------------------------

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	Lendr was unable to connect to Firebase database.	2	Create a Firebase object for modeling our codebase that is made in React.
2	Firebase would crash if any of the values in the schema were null.	1	Make sure that if any values weren't entered in a form, default values were set if no values were entered.

Regression Testing

Defect No.	Description	Severity	How to Correct
1	Fixing null value error in component A (Database) causes component C (React frontend) to load no 'Available items'.	1	Knowing that a value can be null at any point of time, value is initialized as <i>default=null</i> .

Module	Component B - Redux Store
---------------	---------------------------

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	Redux store didn't change states which should've led to UI changes in the application	1	Bind actions creators weren't used, just normal functions were called which led to no calls made to the store. Using bind action creators bound the actions and Redux State in the store was updated.

Regression Testing

Defect No.	Description	Severity	How to Correct
Not Applicable			

Module	Component C - React Frontend
---------------	------------------------------

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	React module wasn't initializing and showed nothing on the webpage when URL was opened.	1	Fixed the id of the div which renders the React content in main Index.js file.
2	Submitting the form for adding a new object to rent showed empty values being added in the database.	3	Event.target.value wasn't put in the firebase call. Putting it in there fixed the data going up to Firebase.

Regression Testing

Defect No.	Description	Severity	How to Correct
1	Fixing the form for sending the object to the firebase resulted in a change in the firebase structure resulting in no objects being pulled from the database.	3	Fixed the format of the objects being sent to the database so that the database call being made for a specific object, actually returned objects we wanted.
2	Changing the div to hold the react content messed up the UI.	1	The div class didn't align with the normalize.css class we needed to make the UI look good when scaled to different sizes.

Module	Component D - User Action
---------------	---------------------------

Incremental Testing

Defect No.	Description	Severity	How to Correct
Not Applicable			

Regression Testing

Defect No.	Description	Severity	How to Correct
Not Applicable			

Module	Component E - Action Creator
---------------	------------------------------

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	Action Creator handles login: When adding login functionality to action creator, did not dispatch action for Login.	1	Returned plain JS object as action instead of binding with 'connect' module using <i>'bindActionCreators'</i>
2	Action Creator also handles event 'Post Item' in adding a new item. The new item was not posted due to the same error as above.	3	Returned plain JS object as action instead of binding with 'connect' module using <i>'bindActionCreators'</i>

Regression Testing

Defect No.	Description	Severity	How to Correct
1	Upon fixing the login action creator using <i>'bindActionCreators'</i> , this.state from local react state management object was not functional since the this object was bound incorrectly.	1	Used 'ref' to store intermediate value of 'this'. The 'this' object was then accessible

			and then login functionality worked.
--	--	--	--------------------------------------

Module	Component F - Reducer
---------------	-----------------------

Incremental Testing

Defect No.	Description	Severity	How to Correct
Not Applicable			

Regression Testing

Defect No.	Description	Severity	How to Correct
Not Applicable			

3 Product Backlog (Updated)

Backlog ID	Functional Requirement	Hours	Initial Status	Status
1	As a first-time user, I would like to be able to sign up.	3	Planned for Sprint 1	Completed in Sprint 1
2	As a user, I would like to be able to login.	2	Planned for Sprint 1	Completed in Sprint 1
3	As a user, I would like to be able to search for items I have interest in.	2	Planned for Sprint 1	Incomplete: moved to Sprint 2
4	As a user, I would like to post items that I am trying to sell.	4	Planned for Sprint 1	Completed in Sprint 1
5	As a user, I want to login using my Google credentials.	3	Planned for Sprint 1	Completed in Sprint 1
6	As a user, I would like to post pictures of items.	2	Planned for Sprint 2	Planned for Sprint 2
7	As a user, I would like to post a description of the items that I have posted.	2	Planned for Sprint 1	Completed in Sprint 1
8	As a user, I would like to post a location for pick up of my items.	2	Planned for Sprint 1	Completed in Sprint 1
9	As a user, I want to set the price for my items.	2	Planned for Sprint 1	Completed in Sprint 1
10	As a user, I want to be able to update the all item information.	2	Planned for Sprint 1	In Progress: moved to Sprint 2

11	As a user, I would like to post items that I want to lend.	3	Planned for Sprint 1	Completed in Sprint 1
12	As a user, I would like post price per unit and duration of lending.	2	Planned for Sprint 1	Completed in Sprint 1
13	As a user, I want to purchase items that are listed.	3	Planned for Sprint 2	Planned for Sprint 2
14	As a user, I want to rent items that are listed.	3	Planned for Sprint 1	Incomplete: moved to Sprint 2
15	As a user, I would like to be able to use Venmo to pay other users.	3	Planned for Sprint 1	Incomplete: moved to Sprint 2
16	As a user, I want to receive push notifications notifying me of potential buyers and when to return rented items.	5	Planned for Sprint 2	Planned for Sprint 2
17	As a user, I want to be able to report other users who are misusing Lendr.	2	Planned for Sprint 1	Completed in Sprint 1
18	As a user, I want to be added to a waitlist for renting an item.	4	Planned for Sprint 2	Planned for Sprint 2
19	As a user, I would like to access an FAQ page when I have questions.	2	Planned for Sprint 1	Completed in Sprint 1
20	As a user, I would like to submit requests for updates.	2	Planned for Sprint 1	Completed in Sprint 1
21	As a user, I would like to be able to request contacting sellers in order to discuss logistics of purchases.	3	Planned for Sprint 1	Completed in Sprint 1