

# Lendr

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

# Testing – Sprint#2

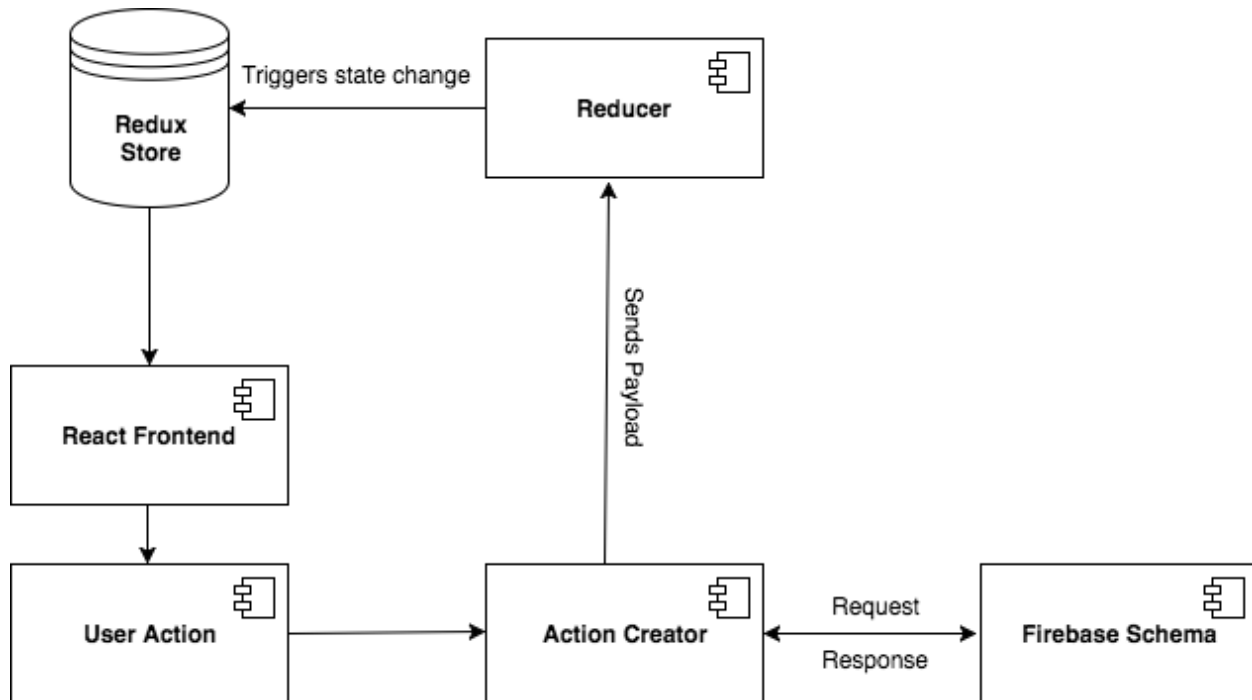
---

## 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
<b>Component A:</b> 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}
<b>Component B:</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}
<b>Component C:</b> 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}
<b>Component D:</b> 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}
<b>Component E:</b> 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}
<b>Component F:</b> 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

<b>Module</b>	Component A - Database
---------------	------------------------

#### Incremental Testing

Defect No.	Description	Severity	How to Correct
1	Firebase was not updating information when user update existing item info	1	Every time a user submitted new item info, send query to Firebase and change ALL existing item info.
2	Users were not being saved to the waitlist.	3	When a user clicks, "WAITLIST" they are placed in the "waitlist" key in Firebase.

#### Regression Testing

Defect No.	Description	Severity	How to Correct
------------	-------------	----------	----------------

1	Since the waitlisted users were not being saved in the database we could not notify them when the product was made available again.	2	The users with the “waitlist” key are extracted from the database, after which they receive an email when the item status changes.
---	---	---	--

<b>Module</b>	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			

<b>Module</b>	Component C - React Frontend
---------------	------------------------------

### Incremental Testing

Defect No.	Description	Severity	How to Correct
1	Button text for rented items was not being updated to “WAITLIST”	1	On user update of item information, provide condition to check if the user updated the status. If status is “RENT” show that on button, if status is “WAITLIST” change button text to waitlist.

2	Pictures were not being shown as part of the item information.	3	Pictures needed to be reformatted into a correct size for placing on the item information page.
3	Purchased items were still being shown on the item's page.	1	Make sure to check flag condition if the item has already been purchased. If it has been purchased do not display item. Only display item to that user who has purchased it under the "Your Items" tab. Do not change for items that were rented.

### Regression Testing

Defect No.	Description	Severity	How to Correct
1	Updated the form to be able to add the picture in the object which is sent to firebase, which resulted in a change in 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 returns the object we want.
2	The waitlist button was not replacing the rent button so we had to change the UI in order to make it work.	1	We checked for the item in database and if it was listed as rented, the button text was changed to RENT.

<b>Module</b>	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

<b>Module</b>	Component E - Action Creator
---------------	------------------------------

### Incremental Testing

Defect No.	Description	Severity	How to Correct
1	Action Creator handles update: When adding update functionality to action creator, did not dispatch action for Update.	1	Returned plain JS object as action instead of binding with 'connect' module using ' <b>bindActionCreators</b> '
2	Action Creator handles waitlist When adding update functionality to action creator, did not dispatch action for Waitlist.	3	Make sure to check condition for status of item and add user email to database.
3	Action Creator handles payment When adding update functionality to action creator, did not dispatch action for Payment.	1	Set flag to "purchased" and add the item to that user's items in "Your Items". Make sure to remove item from being displayed.

### Regression Testing

Defect No.	Description	Severity	How to Correct
1	While fixing the payment action creator using ' <b>bindActionCreators</b> ', this.state from local react state management object was not fully functional as "this" object was not bound correctly.	2	We used 'ref' to store the intermediate value of "this". The 'this' object was then accessible and the payment functionality worked.



<b>Module</b>	Component F - Reducer
---------------	-----------------------

#### Incremental Testing

<b>Defect No.</b>	<b>Description</b>	<b>Severity</b>	<b>How to Correct</b>
Not Applicable			

#### Regression Testing

<b>Defect No.</b>	<b>Description</b>	<b>Severity</b>	<b>How to Correct</b>
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	Overambitious for the time period
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	Completed in 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 an item's information.	2	Planned for Sprint 1	Completed in Sprint 2

<b>11</b>	As a user, I would like to post items that I want to lend.	3	Planned for Sprint 1	Completed in Sprint 1
<b>12</b>	As a user, I would like post price per unit and duration of lending.	2	Planned for Sprint 1	Completed in Sprint 1
<b>13</b>	As a user, I want to purchase items that are listed.	3	Planned for Sprint 2	Completed in Sprint 2
<b>14</b>	As a user, I want to rent items that are listed.	3	Planned for Sprint 1	Completed in Sprint 2
<b>15</b>	As a user, I would like to be able to use Venmo to pay other users.	3	Planned for Sprint 1	Completed in Sprint 2
<b>16</b>	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
<b>17</b>	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
<b>18</b>	As a user, I want to be added to a waitlist for renting an item.	4	Planned for Sprint 2	Completed in Sprint 2
<b>19</b>	As a user, I would like to access an FAQ page when I have questions.	2	Planned for Sprint 1	Completed in Sprint 1
<b>20</b>	As a user, I would like to submit requests for updates.	2	Planned for Sprint 1	Completed in Sprint 1
<b>21</b>	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