

Lendr

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

Product Backlog

Contents

| | |
|---|---|
| 1. Problem Statement | 3 |
| 2. Background Information | 3 |
| 3. Environment | 4 |
| 4. Functional Requirements | 5 |
| 4.1 User | 5 |
| 4.2 Developer | 7 |
| 5. Non-Functional Requirements | 8 |
| 6. Use Cases | 9 |
| 6.1 User | |
| 6.1.1 Case 1: First time user signs up | |
| 6.1.2 Case 2: Returning user can login | |
| 6.1.3 Case 3: User searches for an item | |
| 6.1.4 Case 4: User posts an item to sell | |
| 6.1.5 Case 5: User posts an item to rent | |
| 6.1.6 Case 6: User edits a current posting | |
| 6.1.7 Case 7: User deletes a current posting | |
| 6.1.8 Case 8: User purchases an item | |
| 6.1.9 Case 9: User rents an item | |
| 6.1.10 Case 10: User gets a Push notification for Rent duration | |
| 6.1.11 Case 11: Seller/Lender get a Push notification | |
| 6.1.12 Case 12: Buyers/Renters get a Push notification | |
| 6.1.13 Case 13: User reports another user | |
| 6.1.14 Case 14: User adds an item to waitlist for renting | |

1 Problem Statement

Buying and selling items is how college students both save and make money throughout their college careers. We intend to build, **Lendr**, a buying and selling platform that allows college students to efficiently and effectively buy and sell items from others in their area. The difference between our platform and the current system of using Facebook groups is we will also incorporate a method for people to rent items for a period as well. **Lendr** provides all the of the resources to do what you need to do in one place.

2 Background Information

- Right now, students at Purdue and many other schools use Facebook buy and sell groups to buy and sell items from their peers. Many times, the groups will only let you join if you have a .edu email associated with the school. Lendr will allow people to buy and sell items in the area regardless of a .edu email and provides for more coverage of the population.
- Many times, students require certain equipment, tools, and other necessities for short periods of time to complete a school related or personal task. Lendr will offer the ability for students to rent such items for a period of time to complete the task and return it. We intend to make sure both the person renting and the person lending are equally accountable for the item as well. This is also how Lendr is different from Ebay or Craigslist, as neither provide the renting option.

- Most students use the app *Venmo*, to pay others as it is easy and free to use. We will be incorporating *Venmo* as the main payment method for buying, selling, and lending as it is already well established within most college campuses.

4 Environment

Our team is developing Lendr, a web app that allows the users to buy, sell and rent stuff. A large number of modules are required for the problem we are trying to address and Javascript is very well suited for this purpose. We intend to leverage Node Package Manager and hence, use a Node.js service that provides a functional backend to merge multiple platforms in a single window. A SQL database like MySQL works perfectly to suit our needs as it stores user information and all the items posted by the users. Additionally, the database helps with context management and facilitates smooth functioning of the web app. For the frontend, we plan to use React-Redux, HTML, and CSS. For a CRUD application React will allow smooth and fast re-renders. A React-Redux couple will allow easy state management for the frontend such that user inputs and any listenable actions can be parsed to the backend in structurally sound independent pieces, which also allows easy testing via reducer-action patterns provided by the Redux framework at the same time. For streamline payments we intend to use Venmo API, which is widely used by college students, that will concur with the agreed upon payment amount between the customers. All the environments we plan to utilize have comprehensive documentation allowing for further development.

4 Functional Requirements

4.1 User

| Backlog ID | Functional Requirement | Hours | Status |
|-------------------|--|--------------|----------------------|
| 1 | As a first-time user, I would like to be able to sign up. | 3 | Planned for Sprint 1 |
| 2 | As a user, I would like to be able to login. | 2 | Planned for 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 |
| 4 | As a user, I would like to post items that I am trying to sell. | 4 | Planned for Sprint 1 |
| 5 | As a user, I want to login using my Google credentials. | 3 | Planned for Sprint 1 |
| 6 | As a user, I would like to post pictures of items. | 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 |

| | | | |
|----|---|---|----------------------|
| 8 | As a user, I would like to post a location for pick-up of my items. | 2 | Planned for Sprint 1 |
| 9 | As a user, I want to set the price for my items. | 2 | Planned for Sprint 1 |
| 10 | As a user, I want to be able to update the all item information. | 2 | Planned for Sprint 1 |
| 11 | As a user, I would like to post items that I want to lend. | 3 | Planned for Sprint 1 |
| 12 | As a user, I would like post price per unit and duration of lending. | 2 | Planned for Sprint 1 |
| 13 | As a user, I want to purchase items that are listed. | 3 | Planned for Sprint 2 |
| 14 | As a user, I want to rent items that are listed. | 3 | Planned for Sprint 1 |
| 15 | As a user, I would like to be able to use Venmo to pay other users. | 3 | Planned for Sprint 1 |
| 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 1 |

| | | | |
|----|--|-----------|----------------------|
| 17 | As a user, I want to be able to report other users who are misusing Lendr. | 2 | Planned for Sprint 1 |
| 18 | As a user, I want to be added to a waitlist for renting an item. | 4 | 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 |
| 20 | As a user, I would like to submit requests for updates. | 2 | Planned for 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 |
| | Total | 56 | |

4.2 Developer

| Backlog ID | Functional Requirement | Hours | Status |
|------------|---|-------|--------|
| 1 | As a developer, I would like to come up with regular updates to the app. | 8 | |
| 2 | As a developer, I would like my app to handle at least 100 users at once. | 10 | |
| 3 | As a developer, I would reduce the downtime for maintenance to 2 hours per month. | 5 | |

| | | | |
|---|--|-----------|--|
| 4 | As a developer, I would like to be able to handle the UI for mobile and web appropriately. | 10 | |
| 5 | As a developer, I would like to implement further fixes and enhancements according to the usage of the user. | 8 | |
| | Total | 41 | |

5 Non-Functional Requirements

| Backlog ID | Non-Functional Requirement | Hours | Status |
|-------------------|--|--------------|----------------------------|
| 1 | As a user of Lendr, I expect my information to be secure from other users. | 3 | Planned for sprint 2 |
| 2 | As a developer, I expect Lendr to be scalable. The performance shouldn't be affected with increasing user base. | 10 | Planned for sprint 2 |
| 3 | As a developer, I would like the model wrapper build around React and Venmo API's to be reusable and extensible. | 15 | Planned for sprint 1 and 2 |
| 4 | As a developer, I would like to securely store user usage and telemetry for further understanding of user requirements without storing user credentials. | 15 | Planned for sprint 2 |

| | | | |
|---|---|-----------|----------------------|
| 5 | Build a database that is computationally cheaper to run and maintain. | 15 | Planned for sprint 1 |
| | Total | 58 | |

6 Use Cases

6.1 User

6.1.1 Case 1: First time user signs up

| Action | System Response |
|------------------------------|---|
| 1) Enter email | |
| 2) Enter password | |
| 3) Enter first and last name | 4) Save information to database |
| | 5) Encrypt password using hashing algorithm |

6.1.2 Case 2: Returning user can login

| Action | System Response |
|-------------------|---------------------------------|
| 1) Enter Email | |
| 2) Enter Password | |
| 3) Click Login | 4) Access Database |
| | 5) Check password with database |

| | |
|--|---|
| | 6) If password is correct allow access, if not prompt for entering password again |
|--|---|

6.1.3 Case 3: User searches for an item

| Action | System Response |
|-------------------|---|
| 1) Enter keywords | 2) Check database using keywords |
| | 3) Collect all items that relate to the keyword |
| | 4) Return items with all pictures, description, and information |

6.1.4 Case 4: User posts an item to sell

| Action | System Response |
|----------------------|--|
| 1) Click Sell | |
| 2) Upload Pictures | |
| 3) Enter Description | |
| 4) Enter Price | |
| 5) Enter Location | |
| 6) Click Post | 7) Everything gets stored in the database |
| | 8) UI makes a call to the database to fetch the data |

| | |
|--|--|
| | 9) Database returns the required information to the UI and the post is made. |
|--|--|

6.1.5 Case 5: User posts an item to rent

| Action | System Response |
|-----------------------------|--|
| 1) Click Rent | |
| 2) Upload Pictures | |
| 3) Enter Description | |
| 4) Enter Duration and Price | |
| 5) Enter Location | |
| 6) Click Post | 7) Everything gets stored in the database |
| | 8) UI makes a call to the database to fetch data |
| | 9) Database returns the required data to the UI and the post is made |

6.1.6 Case 6: User edits a current posting

| Action | System Response |
|------------------------|-----------------|
| 1) Go to “My Posts” | |
| 2) Open a current post | |

| | |
|-----------------|--|
| 3) Click Edit | 4) A new edit window opens where user can make changes |
| 5) Make changes | |
| 6) Click Save | 7) The new changes are saved in the Database |
| | 8) The UI loads the edited post |

6.1.7 Case 7: User deletes a current posting

| Action | System Response |
|------------------------|---|
| 1) Go to “My Posts” | |
| 2) Open a current post | |
| 3) Click Delete | 4) An alert shows up with two buttons to confirm action |
| 5) Click Yes | 6) The post gets deleted from the database |
| | 7) The UI reloads and the post is not there anymore |

6.1.8 Case 8: User purchases an item

| Action | System Response |
|--------------------------------------|--|
| 1) Open the post | |
| 2) Click Buy | 3) UI loads the checkout page |
| 4) Agree to the terms and conditions | 5) The pay using venmo button is enabled |

| | |
|--------------------------|---|
| 5) Click Pay using venmo | 6) Make a call to the Venmo API and completes the payment |
| | 7) A call to the database is made and the post is marked sold |

6.1.9 Case 9: User rents an item

| Action | System Response |
|--|---|
| 1) Open the post | |
| 2) Click Rent | 3) UI loads the checkout page |
| 4) Select the duration | |
| 5) Agree to the terms and conditions | |
| 6) The pay using venmo button is enabled | 7) Make a call to the Venmo API and completes the payment |
| | 7) A call to the database is made and the post is marked rented |

6.1.10 Case 10: User gets a Push notification for Rent duration

| Action | System Response |
|--------|--|
| | 1. Rent duration is a certain amount over |
| | 2) Send a reminder email to the user account to return |

6.1.11 Case 11: Seller/Lender get a Push notification

| Action | System Response |
|--|--|
| 1) User offers to purchase or rent an item | 2) System changes status of item |
| | 3) Stops showing item to other users |
| | 4) Send notification to seller that status has changed |

6.1.12 Case 12: Buyers/Renters get a Push notification

| Action | System Response |
|---------------------------------|--|
| 1) Seller/ Lender accepts offer | 2) Buyer receives notification for payment |
| | 3) Link to venmo for payment |

6.1.13 Case 13: User reports another user

| Action | System Response |
|-----------------------|--|
| 1) User submits claim | |
| 2) Submits claim | 3) Sends email to account |
| 5) User exits prompt | 4) Shows user prompt with confirmation email |

6.1.14 Case 14: User adds an item to waitlist for renting

| Action | System Response |
|---------------|------------------------|
|---------------|------------------------|

| | |
|-----------------------------|---|
| 1) Open the Post | |
| 2) Click on Add to waitlist | 3) Save it to the database mapped to the item |