

# System and Unit Test Report

PayPerRead

Jake Armendariz, Daniel Wilby, Brendan Teo, Xin Yan

March 8, 2022

## Sprint 1

- Embedded Iframe reading cookies: As a reader, I want a seamless way to verify access to articles. I'd like to be able to already be signed into my PayPerRead account upon visiting any site with PayPerRead embedded in it.
  - Iframe can read cookies and send requests via the real website
- Backend setup: mongo, create user:
  - Launch web server
  - Connects to mongo
  - Has endpoint to create user
  - After creating user with post request, can read it from a get request
- Create React project: Communicate with backend:
  - Website launches
  - Able to connect to backend
  - Display information from backend endpoint

## Sprint 2

- Reader sign up/ login: As a reader, I want a simple UI for signing up and logging in to my account to add money/manage settings.
  - Click on the sign up link at the top of the homepage
  - Click "sign in with google"
  - Choose your google account
  - Be redirected to the verify signup page if the account does not exist
    - Click the verify button
  - Get redirected to the home page
- Design User signup flow: As a user, I want a secure authorization format :
  - Login with Google Oath (properly following their protocols)
  - Allows user to feel a safe sign in experience.
- Publisher sign up/login: As a publisher, I want a secure login in to manage my account and see views/expected pay.
  - Allow publishers to login using the same procedure as for readers

- Click on the link on the signin page to change account type
- Design and implement home/welcome page:
  - Load PayPerRead website
  - Be able to login via home page
  - Click "About us" and it scrolls to about section
  - Read and learn more about PayPerRead
- User would want to navigate the site:
  - Navbar present on each of the pages
  - Has button for Home, About Us, Signup Or Manage Profile

## Sprint 3

- As a beta user, I would want a usable dummy website to demo PayPerRead.
  - Launch the web server for the dummy website
  - Go to <http://localhost:3001>
  - View the homepage for the dummy website
- Implement Iframe: As a reader, I want the iframe to read my cookies and accept my payment so that I can view the article.
  - Launch dummy website
  - Click on an article
  - Iframe says "Login to pay"
  - Login via PayPerRead
  - Go back to dummy
  - Purchase Article
  - Be able to read the article in the future
- Design/Implement API for registering articles and allowing access: As a publisher, I want to be able to specify the price and id for my articles:
  - Access the Mongo database backend
  - Select article
  - Modify price and id for articles
  - Changes will display for reader viewing Iframe
  - Changes will display for publisher account setting
- Store bought user articles: As a reader, I want the service to keep track of which articles I have already paid for so that I can go back and reread them without having to pay again.
  - Log in as a reader
  - Launch and visit the dummy site
  - Click on one of the articles
  - Purchase the article
  - Close out of the dummy site
  - Log out of the account
  - Log back in as the same reader as before

- Visit the previously bought article
- It should load the article contents without asking for payment
- Flesh out the home page, more work into design and CSS.
  - Visit PayPerRead
  - Clean CSS and information

## Sprint 4

- Manage Reader Account: As a reader, I want to be able to add money to my account or change my existing settings such as adding a new card or deleting my account.
  - Login as a reader
  - Go to manage profile
  - Click the “+” to add money to your account
  - Enter credit card information (can be fake for now)
  - Enter amount to add
  - After adding, an updated balance should be shown in the manage account
  - Navigate to dummy site
  - Purchase an article
  - Check the manage profile page to ensure the article title is showing up
- Manage Publisher Account: As a publisher, I want to be able to see my articles, their view count and my current account balance.
  - Log in as a publisher
  - Register an article using our endpoint
  - On our website, click on “manage profile”
  - View the registered article and its information
- Improved Dummy Site: As a customer, I want a demo to see how the service works.
  - Customer will visit the dummy website
  - Click on one of the articles
  - Create a new reader account
  - Purchase the article
- Mobile View: As a customer, I want to be able to visit the payperread site from my phone.
  - Boot up frontend website and dummy website
  - Switch to mobile view on desktop
  - View check each page shrink properly
- Developer Documentation: As a publisher, I want to be able to view information about how to integrate PayPerRead to my website.
  - Go into the docs folder on our github
  - Look at the README.md
  - View the instructions to integrate PayPerRead