

Good Deeds Anonymous

Ty Richardson

April 24, 2018 | Version 1.0

Document Objectives

The purpose of this document is to provide detailed documentation about the application I am creating for my Prime Digital Academy Solo Project. It includes details such as:

1. A description of all Features (how the app works)
2. The technology stack used (Frameworks, Libraries), including any 3rd party APIs/web services
3. Milestones and a rough schedule
4. Assumptions made during the scoping process

Application Overview

A little emotional uplift! There are good people, living among us, doing good deeds for others simply because they can. And they are us! Not looking for credit, not looking for the limelight, simply wanting share our stories. We can do that on Good Deeds Anonymous. By sharing our stories and reading those of others, we can create a quiet feedback loop that benefits ourselves, others, and the communities we share.

Application Features

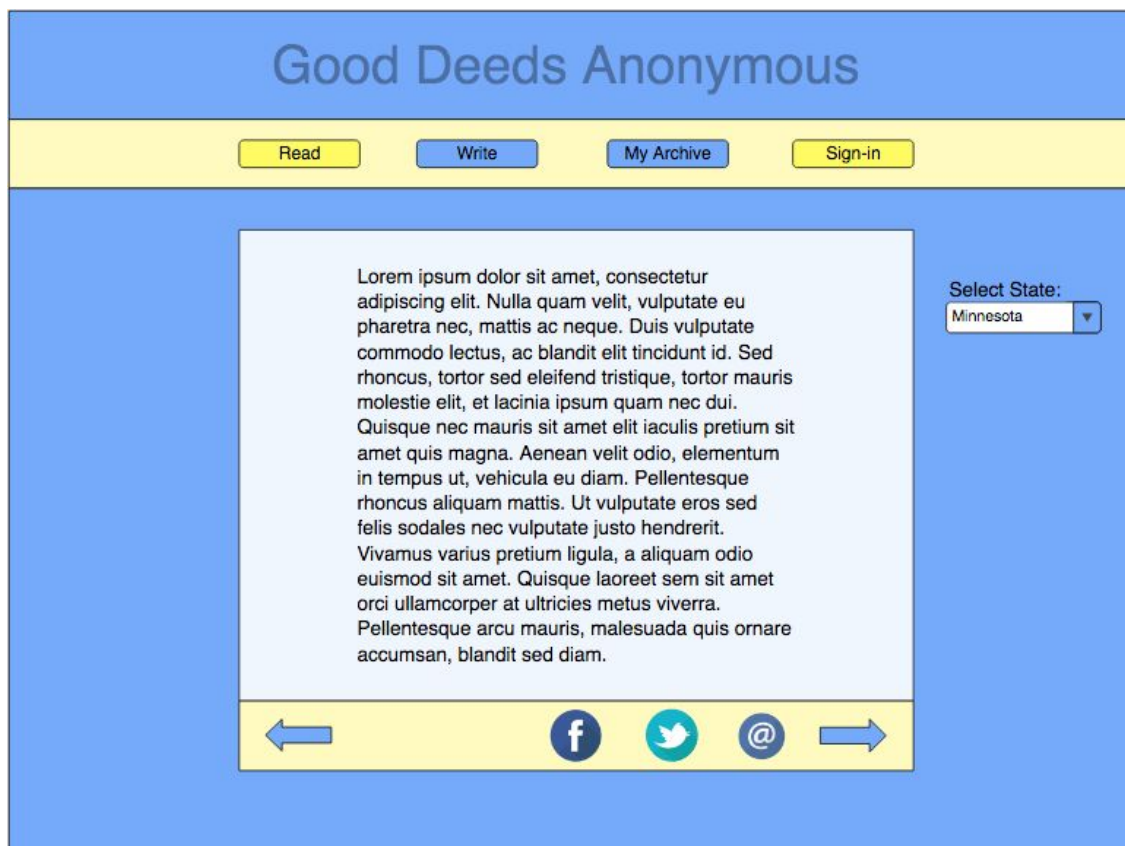
Good Deeds Anonymous (GDA) is a Single Page Application promoting the creation and reading of Stories describing a good deed done, sorted by State (USA).

Each Story is a piece of user-generated content from a Writer, a registered/logged-in user. All visitors to the site are Readers. Stories are sorted by each Writer's State of Residence. In a subsequent update, an Admin role will be added for the purpose of removing Stories flagged as inappropriate by registered users.

Following are descriptions of the views and functionality of Good Deeds Anonymous.

Read (not logged in)

The Read view is the Landing page/Home page of Good Deeds Anonymous. The default Story is a simple introduction to the site, its purpose, and its functionality. All visitors are Readers. Readers can go forward and backward through the randomized list of sorted stories. Using a drop-down, Readers can narrow the selection of Stories by State. Readers can share stories via Facebook, Twitter, and Email. All visitors are encouraged to become Writers via the Sign-in button in the navigation bar.



Sign-in

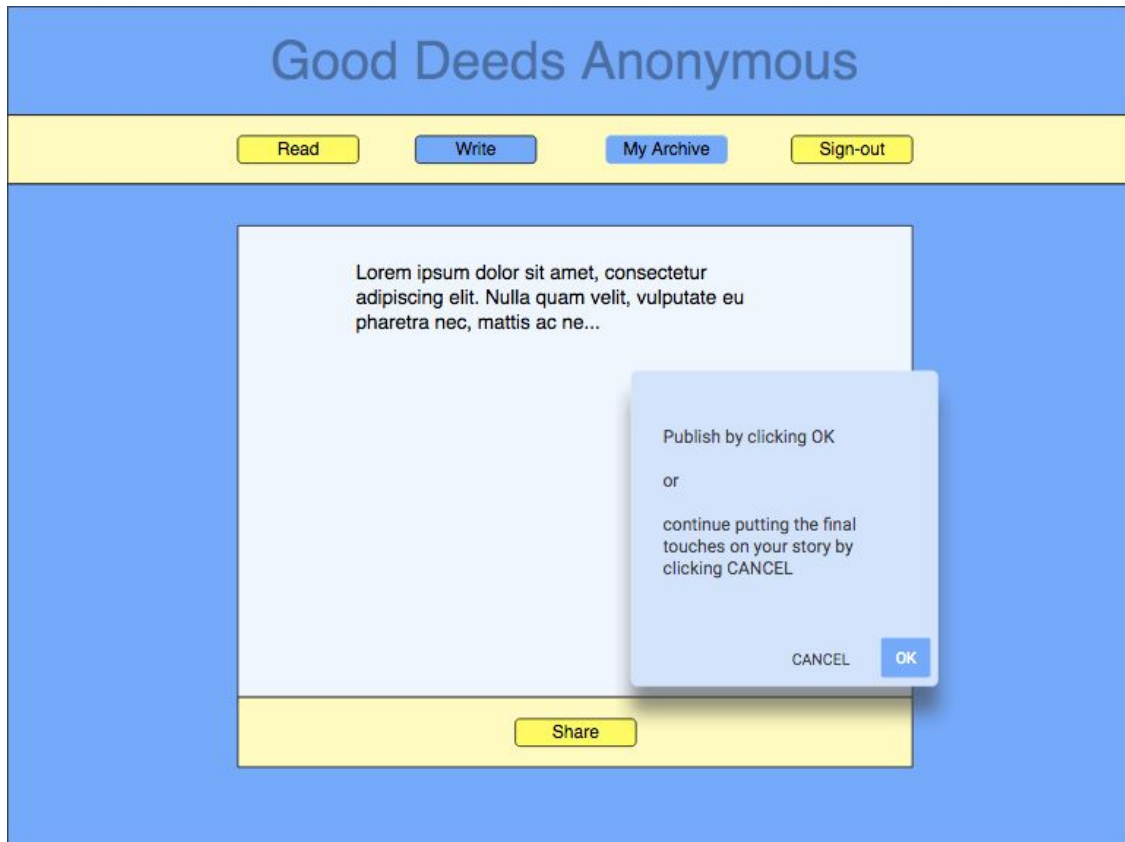
The Sign-in view allows a visitor to register as a Writer or, if already a Writer, they can login. The Read button remains active, so the user can go back if they decide not to register or login. If a registration or login is submitted and is successful, the Write view displays.

The screenshot shows a web interface for 'Good Deeds Anonymous'. At the top is a blue header with the site name. Below it is a yellow navigation bar with four buttons: 'Read' (yellow), 'Write' (blue), 'My Archive' (blue), and 'Sign-in' (blue). The main content area has a blue background and contains a white registration/sign-in box. This box is divided into two sections: 'Newcomer registration:' and 'Sign-in:'. The registration section includes text input fields for 'type a username...', 'type a password...', a dropdown menu for 'Select the State you live in...', and a yellow 'Submit' button. The sign-in section includes two text input fields for 'type your username...' and a yellow 'Submit' button.

NOTE: Sign in and sign up will be on two different pages.

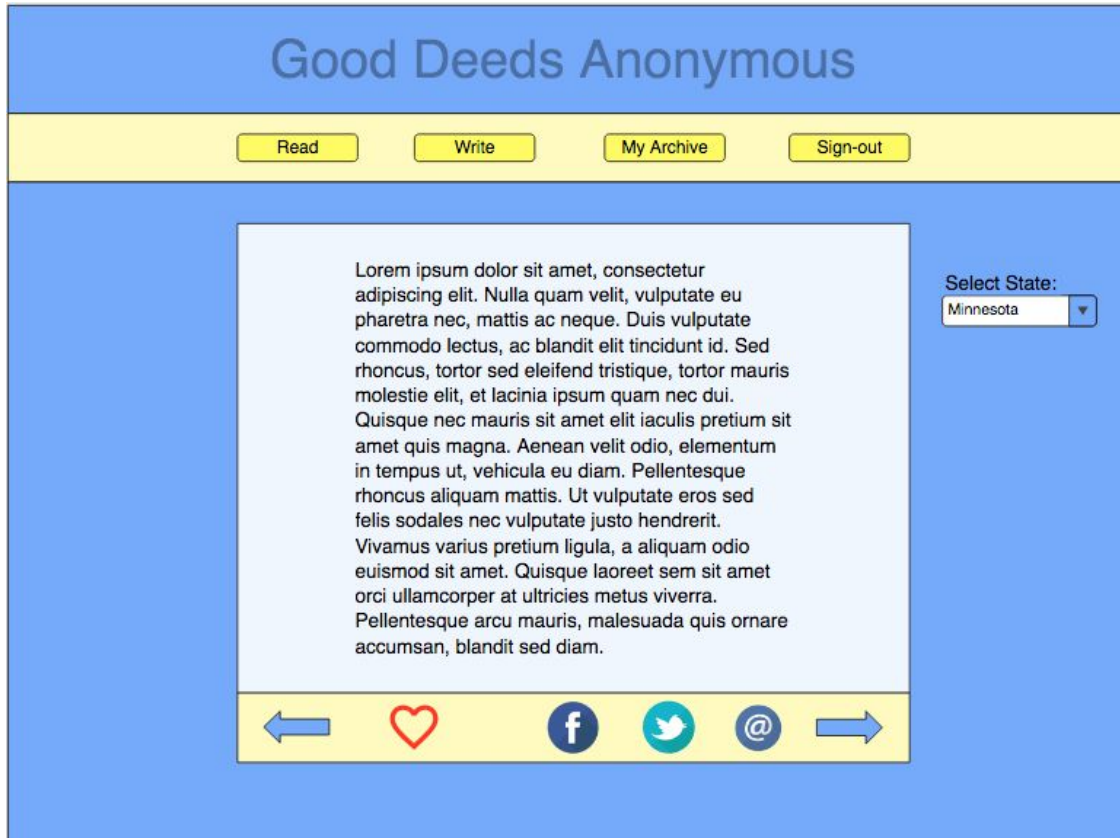
Write

The Write view allows a Writer to type and Share a Story. Upon clicking Share, a prompt is displayed asking if they are sure they are ready to share their story. They can select Publish or Edit. Selecting Edit keeps them in the Write view, allowing them to continue working on their Story. When Publish is selected, the Read page displays the Writer's newly published story. (The Writer can choose to Sign-out or navigate to the Read view, instead of sharing/publishing. In either case, the Writer is alerted that no draft of their work will be saved if they continue. They can then choose to stay or leave.)



Read (logged in)

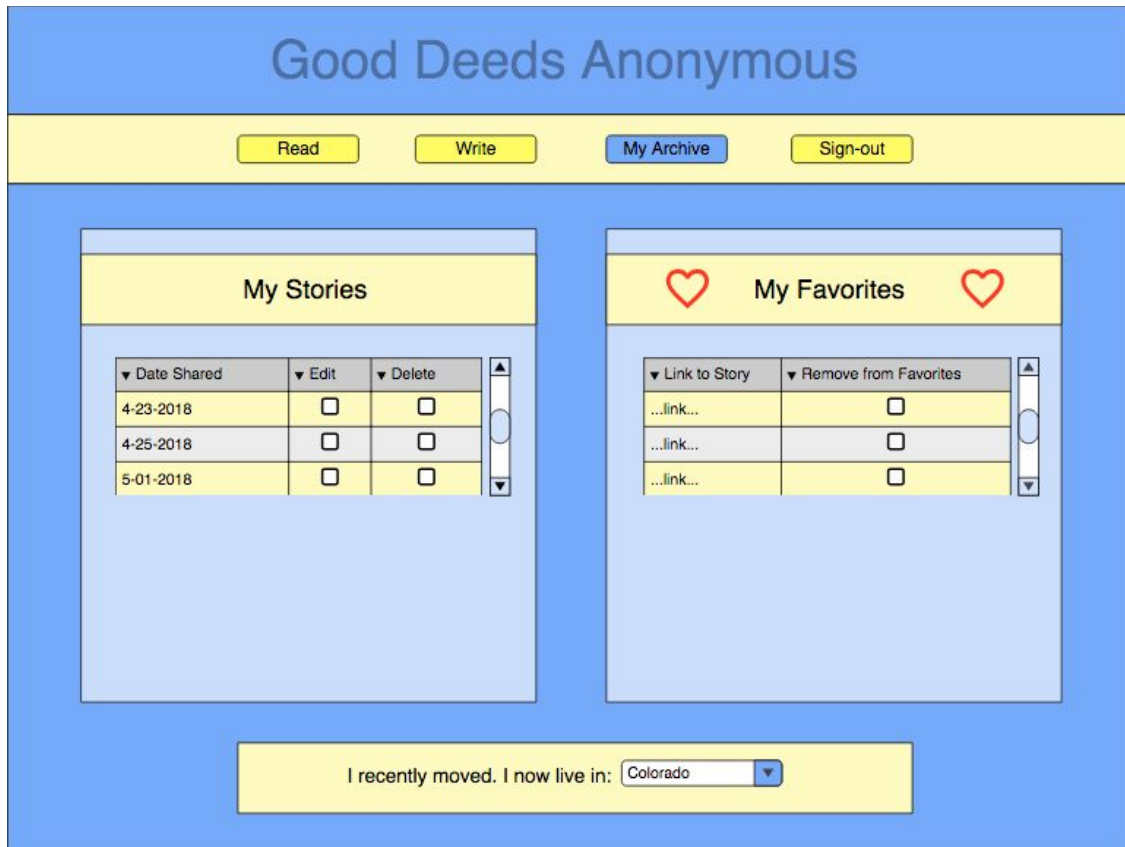
The Read view is always accessible for a logged-in Writer with all functionality active. Unlike a Reader, a Writer can mark Stories as Favorites and they can access their Favorites list and a list of their published Stories by clicking the My Archive button in the navigation bar, where the Sign-in button label now reads Sign-out. In a subsequent update, Writers will be able to flag Stories as inappropriate, alerting an Admin to decide whether or not to delete the offending Story. An Admin view will be added at that time.



NOTE: added a button to mark a story inappropriate; to be reviewed by Admin

My Archive

The My Archive view displays two lists: a list of the Writer's published stories, which allows them to Edit or Delete Stories, and a list of the Writer's Favorites, which allows them to unfavorite an item, which removes it from their Favorites list. In case a Writer moves, they can update their State of Residence.



Mockups created using: <https://app.moqups.com/>

Project Milestones and Schedule

Milestone (feature)	Hours Estimate	Estimated Due Date
Create database with dummy data	.5	30-April
Set up project	.5	30-April
Create base file structure	1	30-April
Create App, sketch basic page (header and nav)	2	30-April
Sketch out Landing/Read view	1	30-April
Sketch out Registration view	1	30-April
Sketch out Login view	1	30-April
Sketch out Write view	1	30-April

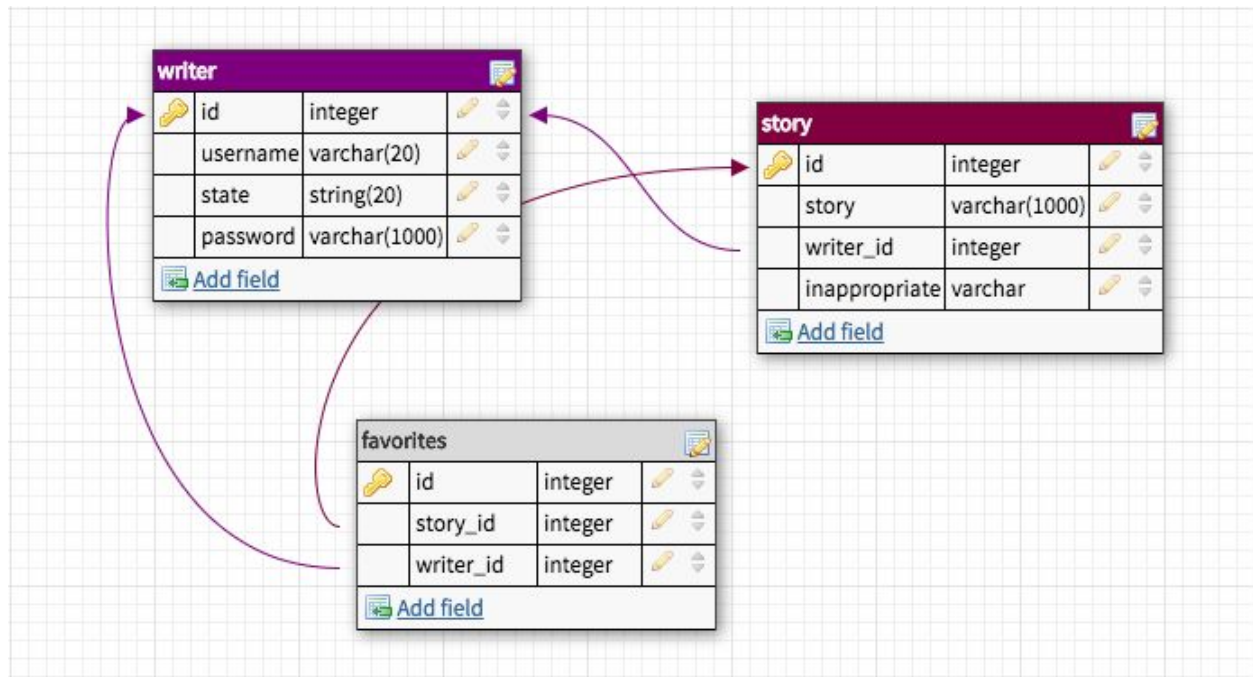
Sketch out Landing/Read view	1	1-May
GET for Landing/Read default view using dummy data	2	1-May
GET all for Read view sorted by most recent	2	1-May
Sketch out Landing/Read view mini-toolbar	1	1-May
Implement forward and backward buttons for Landing/Read view	3	1-May
Implement combo box for selection of State; default is to bring in all stories sorted by most recent.	2	2-May
GET by location for Read view using dummy data	2	2-May
Registration/Login/Logout	2	2-May
POST for Write view	2	2-May
GET for logged-in Read and Archive views by writer.id sorted by most recent; default is all. Select by state to narrow stories to single state.	2	3-May
DELETE for deleting a Story	2	3-May
UPDATE for displaying and editing a published Story in Write view (GET specific story by id)	3	3-May
UPDATE for adding Favorites	2	3-May
GET my favorite stories	2	4-May
UPDATE for removing Favorites	2	4-May
UPDATE for Writer's change of residence (State)	2	4-May
Add social media sharing functionality to Read view mini-toolbar	2	7-May
Implement Material UI	24	9-May
User Testing	2	10-May
UPDATE for marking stories inappropriate	2	10-May
GET for Admin and display in Archive of stories marked inappropriate by logged-in users	2	10-May

Add button for marking a story inappropriate to Read mini-toolbar

2

10-May

Database Documentation



ERD created using: <http://www.dbdesigner.net/>

NOTE: all id's are SERIAL; state created as varchar; favorites table name made singular, favorite.

Routes

Registration/Login

GET all for Read view sorted by most recent

GET by location for Read view

POST for new Story from Write view

UPDATE for adding Favorites

UPDATE for removing Favorites

GET my stories (logged in user)

GET my favorite stories

GET specific story by id

UPDATE for displaying and editing a published Story in Write view

DELETE for deleting a Story

Stretch

UPDATE to flag a story

Browsers

Application will fully support browsers listed below. All browsers or versions not listed below are considered out of scope.

Browser Name	Mobile or Desktop?	Version
Chrome	Desktop	66.0.3359.117

Assumptions

While completing this estimate the following assumptions were made:

Technologies

- React
- Redux
- Material UI
- Node
- Express
- PostgreSQL
- Heroku