CASE STUDY ·

Story Syntax, a modern framework for writing user stories

Intro

In 2016, I signed up to take General Assembly's 3 month full-stack development bootcamp to help solidify my understanding of development to become a more effective product manager. As one of many class projects, I had to develop a product on my own from conception to launch in 4 days, and I wanted it to be a unique tool for Product Managers like myself.

Challenges

Time Constraint	Scope	Unfamiliarity
Learning Fullstack development & ambitious even with a B.S. in Information Systems & Business	4 days to concept, design, architect, and develop this product as a class assignment	Making a unique tool that other product managers could benefit from and would want use

Planning

Requirements, deliverables, and success criteria

As a student project I had some non-negotiable technical requirements and deliverables. These helped set the project success criteria:

Project technical requirements included: at least 2 database models, sign up/login functionality with hashed passwords & an authentication flow, at least one API incorporated, complete RESTful routes for at least one resource, accessibility standards (class FTW), and CSS, and accessible to the public.

Project deliverables included: a working full stack application, readme.md file explaining technologies used, the approach taken, installation instructions, and known bugs or unsolved problems, and a gitt repository hosted on Github, with a link to your hosted project.

Items	Expectations
1	Functionality
2	Clear UX and aesthetics
3	Meets expectations, great job!
4	Exceeds expectations, you wanted it, you got it!

General Assembly's project success criteria

What to build?

As a product manager I found myself frustrated with the lack of useful, user-friendly tools available for the tasks I do on a regular basis. Writing user stories is essential to the product management and agile process used by many organizations today, yet the time wasted over story format such always sucks. This was a problem I was passionate about and capable of solving, which led to the conception of "Story Syntax, a framework to help product managers and other disciplines write modern, more effective user stories. With a unique product idea and a concise vision, I was ready to get started.

Research

What are user stories and why use them?

I started out by researching user story articles from other product management and agile experts to answer the following questions:

1. What are user stories?
2. Why do we use them?
3. What are some common formats for writing user stories?

User stories are (in theory) at feature or product requirements framed from the perspective of the user in agile software development environments. User stories are conversation starters and contact sections that help product team members, focus on defining the problem while empowering designers and developers to come up with creative solutions. Once written, these user stories can be used to create a prioritized product backlog.

The basic user story format uses the following structure:

```
"As a [user/persona] I want to [accomplish task] so that [value/goal]"
```

What's the difference between a good user story and a great one?

When I've taken over product management of existing products, most of the product backlog items seemed written in the basic user story format.

```
"As social media user I want to add a friend I searched for to my
friend network so that I can see their posts, they can see my posts
and I can direct message them."
```

While this user story format covers the basics, without acceptance criteria, user stories can be difficult to test. It's also more difficult for designers and developers to provide accurate and reliable time estimates for feature development.

What's acceptance criteria?

Acceptance criteria are the specifications that a software feature or product must accomplish before it can be accepted by or released to a user, stakeholder, or another software system. My research confirmed that scenario oriented acceptance criteria written in Gherkin Language is used by the software development experts at Pivotal Labs, popular testing platform Cucumber.io, and many other agile organizations.

While it may sound like a cool place to visit Gherk is not a nation state and Gherkin isn't a spoken language. It's a syntax for writing software acceptance tests.
An example of a scenario oriented acceptance test written using Gherkin:

```
Given: [Story context and preconditions]
When: [key action performed by the user]
Then: [Desired user output]

Optional Scenarios

And: [Additional condition]
But: [Common counter to user story premise]
```

Let's take a look at a **great** example of a user story with acceptance criteria that became the foundation of Story Syntax.

```
"As social media user I want to add a friend I searched for to my
friend network so that I can see their posts, they can see my posts
and I can direct message them.

Given I have searched for a friend's name

When I select "Add Friend" next to my friend's name

Then my friend's name should appear in my friend list on my
homepage."
```

Designing the solution

With the planning and research phases complete, it was time to organize this information into a technological solution. I used Pivotal Tracker to craft some great user, and used a combination of whiteboarding and draw.io to sketch out the entity relationship diagram (database schema) and wire frames.

1

2

Developing & launching the solution

Database, server, & restful route setup

As a developer in training, I used the technologies I was learning in the General Assembly Web Development Immersive program: postgresql as the core language and Express, Node, and SQL for the Backend, and Bootstrap and EJS for the frontend.

express · Node.js · SQL · Bootstrap · EJS

Using the ERD diagram, I setup my structured query language (SQL) database and connected it to my Node server using object relational mapping (ORM). To satisfy the project requirements, I setup Create, Read, Update, and Delete (CRUD) functionality for the authors and stories models,/database tables. With an API that approach, I began setting up RESTful routes for the homepage, login, and dashboard pages using Express.

With basic CRUD setup, I wanted users to be able to login to their user stories with secure user sessions, so I setup basic authentication encrypting the database and hashing user passwords to ensure cyber security best practices and added facebook auth. Once I had both these options working, I added error states to the form fields and validations upon submit, to ensure users would be successful using the Story Syntax forms.

After users log in, I wanted to route them to a dashboard page where they could view all the user stories they created. This required injecting user specific database content into the HTML via EJS for programmatic reading and updating of information before and after it's been added to a database. With the core functionality of the app developed, I decided to call it a day.

Front-end, testing, and launch

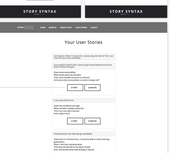
The next day, even though I woke up with a fever, I went to class to design and develop the front end HTML and CSS pages. I went with a visual style that was clean and professional, ensured my code was semantically clean, added comment, and implemented an icon set.

With the front-end and backend complete, I conducted user testing with classmates, friends, and instructors. The feedback was overwhelmingly positive. I noticed users could create a user story without creating an account (resulting in their user story being lost), so I figured out how to store user stories as cookie storage for posting to the database once a user logs in. Using cookie storage, I could temporarily store multiple user stories that would persist even if the user left the site and came back a week later.

Happy with Story Syntax's functionality, I finished up front-end cleanup and responsive bug fixes, and prepared for launch. I selected Heroku for cloud deployment since they offer free cloud hosting and data storage for web apps under a certain threshold of usage.

The website

Story Syntax web app opens with a hero image with product messaging and an invitation to create your first story. Users can navigate to the "Why Story Syntax" section to learn about the development process, the primary services of Story Syntax and the user story creation form, w/ added tooltips to help users understand each field and Cherbin Syntax placeholder text to provide an example.

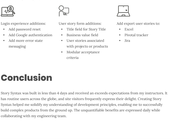
1


2

3

4

After creating a story, you can save and view your story by creating an account or by using Facebook signup[1]. Once logged in, you can view, edit or delete[2] your user stories or create new ones.

5


6

Measuring results

Gathering metrics and observing users

I successfully conceived, designed, developed and launched Story Syntax in less than 4 days, receiving "exceeds expectations" and ranking in the highest percentile in my cohort. Afterwards, I used Google Analytics and Hotjar to measure the results of my stretch success criteria. To my surprise and delight, people all over the world are using Story Syntax, most of which were finding Story Syntax organically through Google or Reddit.

1

2

From the data, I detailed 2 distinct user personas. The first persona I called "The Reader" and the second "The Writer". Readers comprised 60% of Story Syntax site visitors, and their primary behaviors included reading site content and browsing form functionality before exiting the site. The Writers, however, used the product and created user stories and accounted for ~60% of all site visits.

Next steps

While Story Syntax is providing value for users across the globe, there are numerous improvements and new features that would make Story Syntax provide more value for users. I've created a backlog of feature updates using data gathered from the first release:

Login experiences additions	User story forms additions	Add export user stories to:
• Add password reset • Add Google authentication • Add more error state messaging	• Title field for Story Title • Business value field • User stories associated with projects or products • Modular acceptance criteria	• Excel • Pivotal tracker • Jira

Conclusion

Story Syntax was built in less than 4 days and received an exceeds expectations from my instructors. It has receive users across the globe, and site visitors frequently express their delight. Creating Story Syntax helped me solidify my understanding of development principles, enabling me to successfully build complex products from the ground up. The unquantifiable benefits are expressed daily while collaborating with my engineering team.

Ready to write your own user stories?

VISIT STORYSYNTAX.COM →