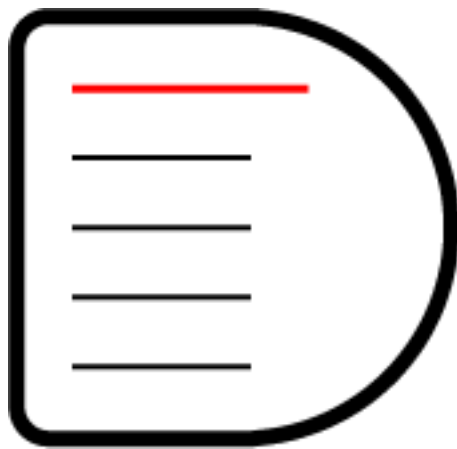


# Draftly



## Table of Contents

<b><i>Interactive Storytelling .....</i></b>	<b><i>2</i></b>
<b><i>How to run [NEW] .....</i></b>	<b><i>2</i></b>
<b><i>Old Version.....</i></b>	<b><i>3</i></b>
<b><i>Current Concepts .....</i></b>	<b><i>6</i></b>
<b><i>Current Wireframes .....</i></b>	<b><i>7</i></b>
<b><i>Current Sitemap .....</i></b>	<b><i>10</i></b>
<b><i>Whats on Each Page .....</i></b>	<b><i>10</i></b>
<b><i>Database Aspect that I ended up not using .....</i></b>	<b><i>13</i></b>
<b><i>Concept Sketches.....</i></b>	<b><i>14</i></b>
<b><i>How to get it working.....</i></b>	<b><i>16</i></b>
<b><i>Lists of Tables .....</i></b>	<b><i>17</i></b>
<b><i>Rundown on how it will be connected .....</i></b>	<b><i>19</i></b>

## Interactive Storytelling

My project currently called Draft is about providing a product or service which will be an Interactive Website that encourages gradual improvement for creative writing and freestyle. This in hope will inspire creativity and be an alternative interactive learning process that can be used. This pertains to writers or having an interest in writing.

## How to run [NEW]

Opted to build in Flask than Database method

In Terminal

Go to the Draft Interactive Directory

cd /DraftInteractive

Things you need to install include

### Install Homebrew

/bin/bash -c "\$(curl -fsSL

<https://raw.githubusercontent.com/Homebrew/install/master/install.sh>)"

### Install either Pipenv

\$ brew install pipenv

### or Virtualenv

pip install virtualenv

### Install Python

\$ brew install python

\$ python3

# Do I have a Python 3 installed?

\$ python --version

Python 3.7.1 # Success!

### Install Flask

\$ python3 -m venv venv

or

\$ python2 -m virtualenv venv

source venv/bin/activate

\$ pip install Flask

\$ pip install gunicorn

\$ pip install eventlet

\$ pip install requests

\$ pip install -U flask-cors

### Run the virtual environment

[Old]

```
export FLASK_APP=app
```

```
flask run
```

```
http://localhost:5000/
```

[New]

```
FLASK_DEBUG=1 python app.py
```

```
http://localhost:5000/
```

For Mobile view

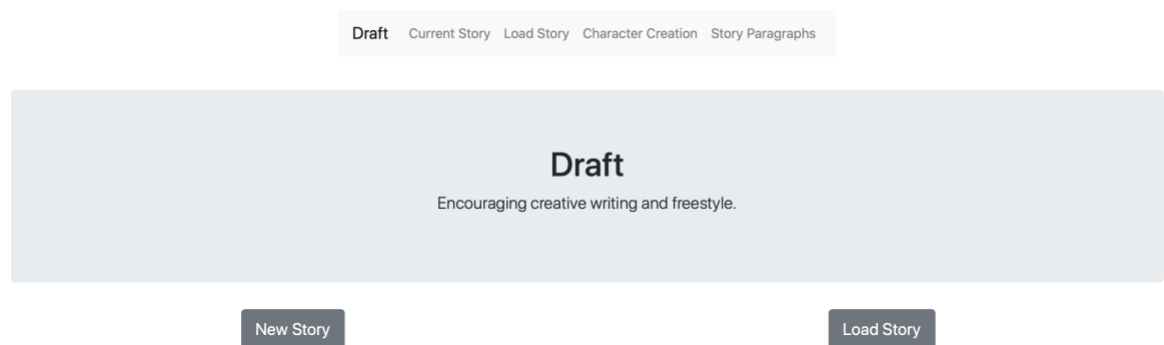
Your IP address:5000/

Or

<https://draftly.herokuapp.com>

## Old Version

- Home  
Page
- 



- Load Story Page

[Draft](#) [Current Story](#) [Load Story](#) [Character Creation](#) [Story Paragraphs](#)

## Draft

Encouraging creative writing and freestyle.

## Load Story

- New Story Page

[Draft](#) [Current Story](#) [Load Story](#) [Character Creation](#) [Story Paragraphs](#)

## Draft

Encouraging creative writing and freestyle.

### Story Creation Page

Name your Story

Choose a Plotline

Overcoming the Monster

Create Character

Submit

Create Character

- Character Creation Page

Draft

Current Story

Load Story

Character Creation

Story Paragraphs

Draft

Encouraging creative writing and freestyle.

Character Creation Page

Name your Character

Name

Choose a Role

The Villain

Choose an Archetype

Innocent

Save

- Story Paragraph Page

Draft

Current Story

Load Story

Character Creation

Story Paragraphs

Draft

Encouraging creative writing and freestyle.

Story Paragraphs

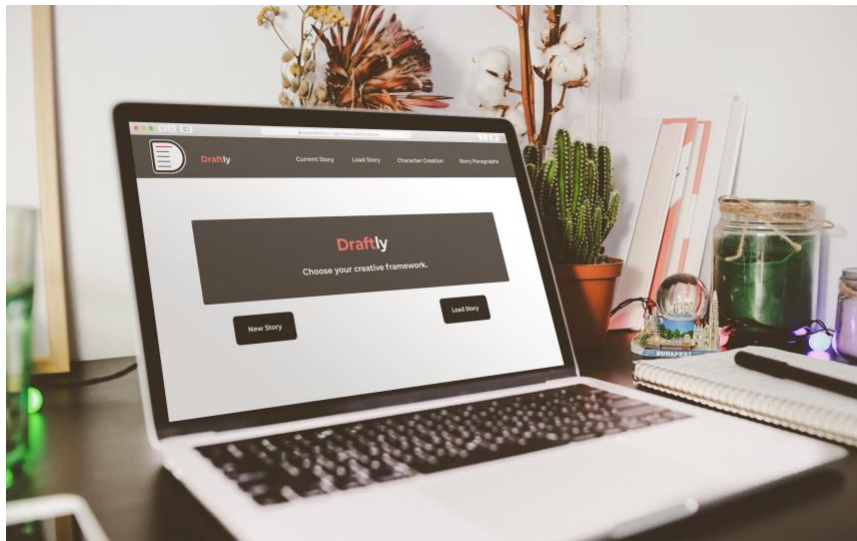
New Overcoming the Monster

Phase 1

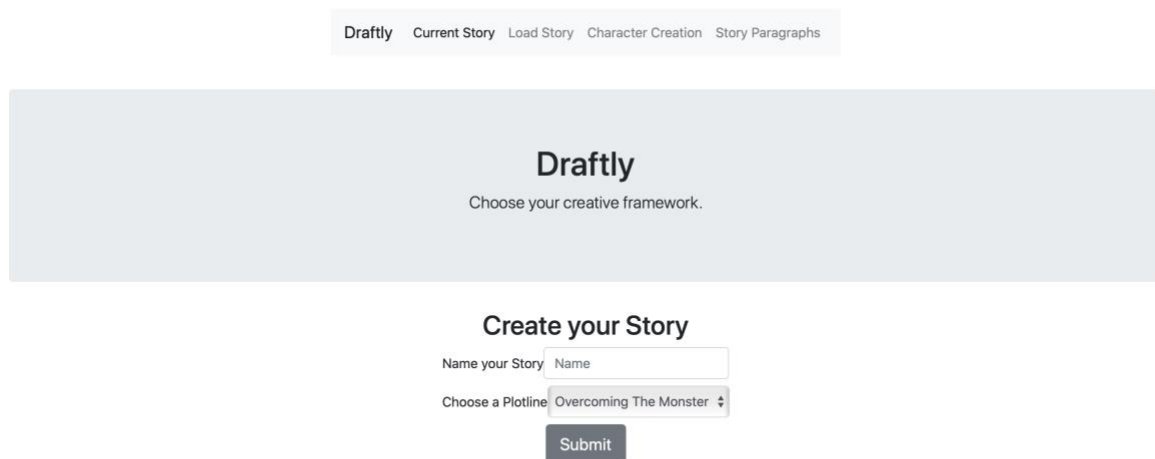
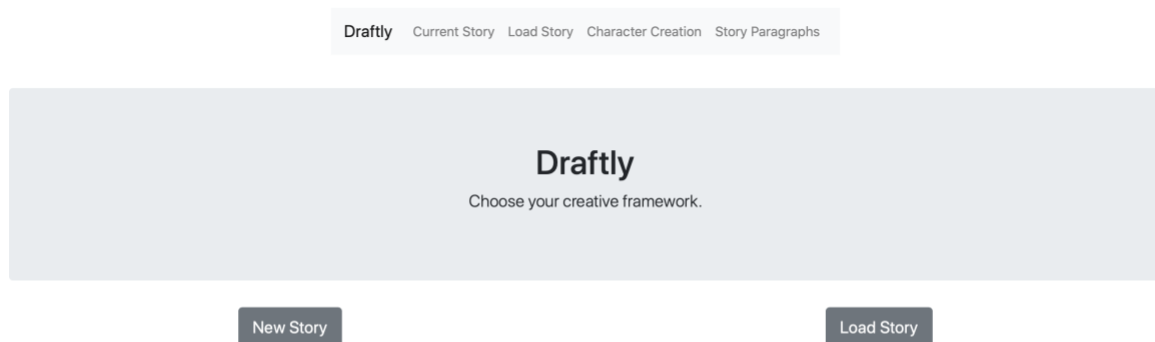
Phase 2

Phase 3

## Current Concepts



## Current Wireframes



## Draftly

Choose your creative framework.

### Create your Characters

Name your Character

Choose a Role

Choose an Archetype

Create Character

### List of Characters

Submit

## Draftly

Choose your creative framework.

### Start writing your story

Story Name:

New

Chosen Plotline:

Overcoming The Monster

List of Characters

New The Villain Innocent

### Completed Phases

Current Phase:

Anticipation

Continue

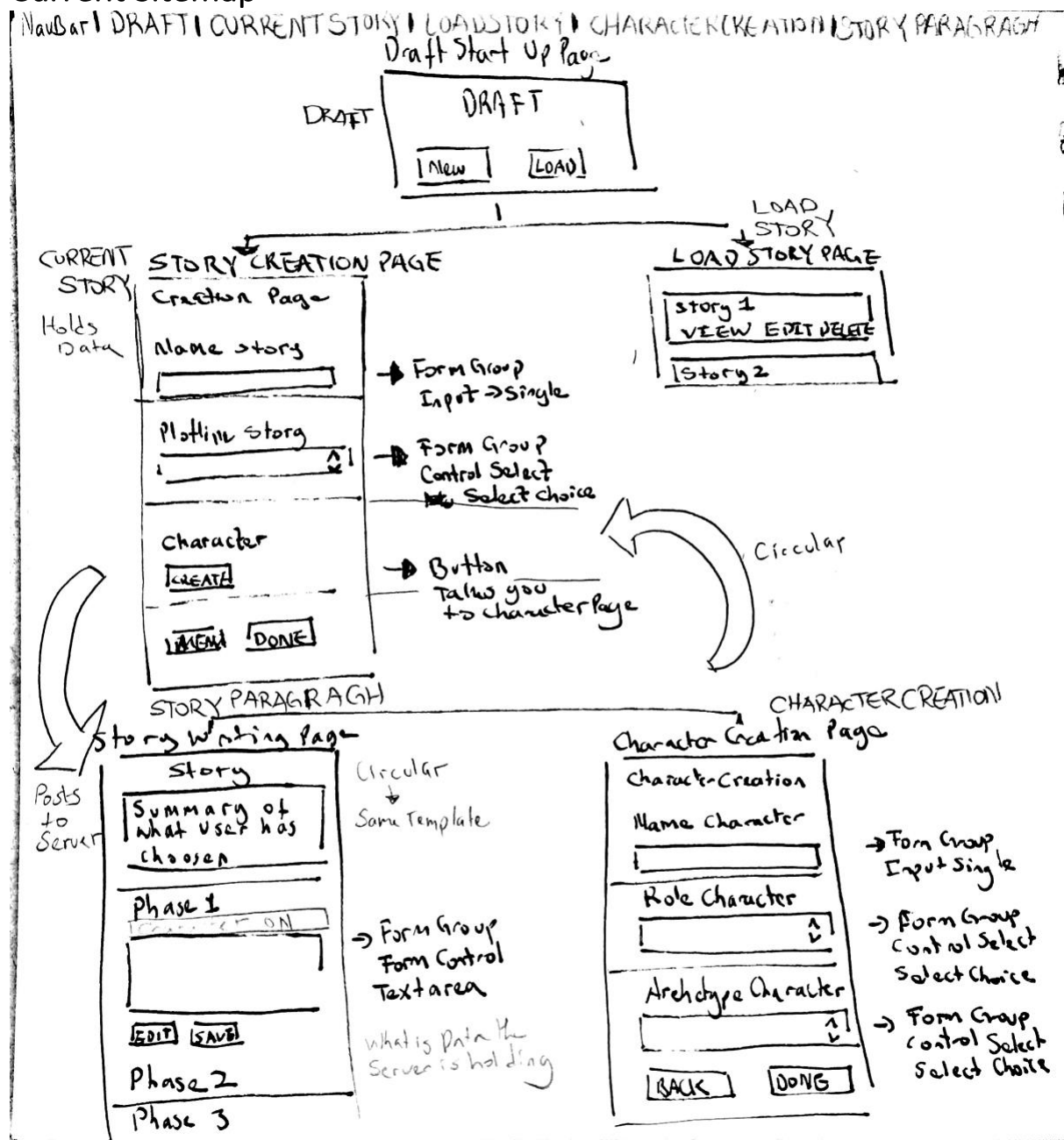


## Draftly

Choose your creative framework.

### Load Story

## Current Sitemap



## Whats on Each Page

startup Page : <http://localhost:5000/>

Template: startup.html

Method: GET

- Button -> newstory.html -> (POST)
- Button -> loadstory.html -> (POST)

NOTE:

- Contains Button to link to New Story Page
- Contains Button to link to Load Story Page

**newstory Page :** <http://localhost:5000/newstory>

Template: newstory.html

Method: POST

**INCOMING FIELDS**

- (none)

**PAGE FORM**

- Form Action -> charactercreation

1)StorynameFormControllInput (Story Name)

2)PlotlineControlSelect (Story Plotline)

- Button -> charactercreation -> (POST)

NOTE:

- Contains Story Name
- Conatins Story Plotlines
- Story Plotlines (Dict. storyPlotlineToPhase) have Phases (Append. phasesForStory)
- Contains Button to link to the Character creation Page

**charactercreation Page :** <http://localhost:5000/charactercreation>

Template : charactercreation.html

Method : POST

**INCOMING FIELDS (to the handler)**

- First time into the page/handler from newstory:
- Story name
- Story plotline
- Subsequent times into the page/handler from itself! (charactercreation)
- Character Name
- Character Role
- Character Archetypes

**PAGE FORM**

- Form Action -> storyparagraphs

1)CharacterFormControl (Character Name)

2)RoleControlSelect (Character Role)

3)ArchetypeControlSelect (Character Archetype)

- Button -> Create Character -> (VALUE)
- Button -> storyparagraphs -> (POST)
- List of Characters (Append. theCharacters)

1)Name

2)Role

3)Arche

NOTE:

- Contains Character Name
- Contains Character Role
- Contains Character Archetype
- Contains List of Character table with Name, Role and Archetype of all characters created
- Contains Button to create the character
- Contains Button to link to the Story Paragraph Page

**storyparagraphs Page :** <http://localhost:5000/storyparagraphs>

Template : storyparagraphs.html

Method : POST

- Badge -> StorynameFormControllInput (Story Name)
- Badge -> PlotlineControlSelect (Story Plotline)
- Badge -> List of Characters (Append. theCharacters)
- Phases (Append. phasesForStory) from (Dict. storyPlotlineToPhase)

1)Phase Label

2)paragraphsFormControlTextarea (Story Paragraph Text Area)

- Button -> Saves Fixed Story Paragraph, Moves to Next Story Phase -> (VALUE)

NOTE:

- Contains Story Name Label
- Contains Story Plotline Label
- Contains List of Character Table Label
- Conatins Phases and Paragraphs
- Contains Button for making the paragraphs fixed and moving to the next Story Phase

**loadstory Page :** <http://localhost:5000/loadstory>

Template : loadstory.html

Method : POST

NOTE:

- Future Development or Update for Selecting Multiple Story Works

# Database Aspect that I ended up not using

## Bare Bones of How it used to look like [OLD]

file:///Users/tatidisu/Documents/GitHub/DraftInteractive/app/templates/index.html

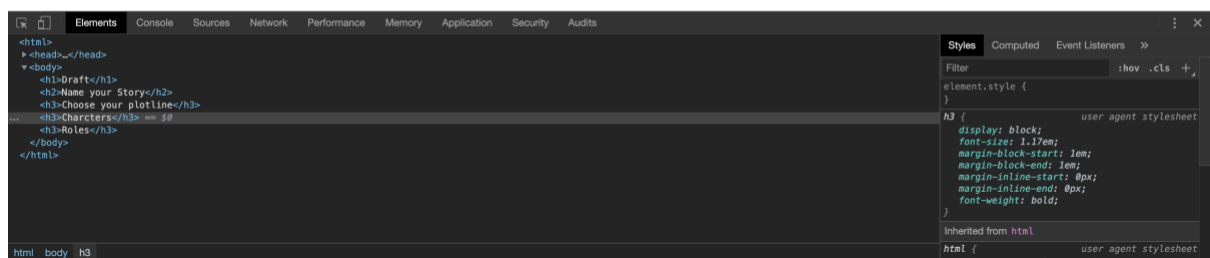
### Draft

#### Name your Story

Choose your plotline

#### Characters

#### Roles



# Concept Sketches

## Homepage

DRAFT

[ ] [ ] [ ] [ ]

FIRST  
CREATION  
PAGE

↓  
Name your game / story

[ ]

choose your plotline

[ ] [v]

~~Name~~ Name your character

[ ]

choose your role

[ ] [v]

[START]

## Load Screens

[ ]

[ ]

[ ]

Name  
choose  
character  
[+] character role

The more you  
add the more it  
increases

[ ] [ ] [ ]

Sitemap  
now through  
Pages

• HTML

Py  
templates & Py  
chatrooms. nglah

Name your character

[ ]

[+]

maybe you can click on  
some parts to see your  
characters

DRAFT

[ ]

STAGE 1

[ ]

[ ]

Form posting  
multi pages

NOTE

Navigation  
Different story subplots  
with the diff plots

# Database Sketch

Interaction  
(line Data Input)

Fixed  
Build from Code

STORY
ID Name

PLOTLINE	FIXED
ID Name	

FIXED		
PLOT	PHASE	ROLE

CHARACTER
ID NAME

ROLE	FIXED
ID Name	

PROSE
ID TEXT

PHASE	FIXED
ID Name	

Sequential (in Order)

Sequential (in Order)  
Note: might need to have  
an add selection  
character

CORM  
Objectrelational  
mapping

## How to get it working

Using Pewee

<https://github.com/coleifer/peewee>

<http://docs.peewee-orm.com/en/latest/peewee/installation.html>

- Database with all the Databases

source venv/bin/activate

python app/database/plotline.py

sqlite\_web my-database.db

### my-database.db — plotlines

archetypes

characters

linkages

paragraphs

phases

**plotlines**

roles

storys

Structure

Content

Query

Drop

Import

new\_table\_name

Create

#### SQL

```
CREATE TABLE plotlines (  
  id INTEGER NOT NULL,  
  name VARCHAR,  
  PRIMARY KEY (id)  
)
```

#### Columns

+ Add column

Column	Data type	Allow null	Primary key	Actions
id	INTEGER			Rename   Drop
name	VARCHAR			Rename   Drop

#### Indexes

+ Add index

Name	Columns	Unique	SQL	Drop?
------	---------	--------	-----	-------

Web-based SQLite database browser, powered by Flask and Peewee. © 2020 Charles Leifer

- Database with the foreign linkage

source venv/bin/activate

python app/database/tester.py

sqlite\_web database.db



table name...
plotline
story

Toggle helper tables

Structure

Content

Query

DropImport

SQL

```

CREATE TABLE "plotline" (
  "id" INTEGER NOT NULL PRIMARY KEY,
  "child_id" INTEGER NOT NULL,
  FOREIGN KEY ("child_id") REFERENCES "story" ("id")
)

```

Columns

+ Add column

Column	Data type	Allow null	Primary key	Actions
id	INTEGER			Rename   Drop
child_id	INTEGER			Rename   Drop

Foreign Keys

Column	Destination
child_id	story.id

Indexes

+ Add index

Name	Columns	Unique	SQL	Drop?
plotline_child_id	child_id		SQL	Drop

## Lists of Tables

There are 8 Tables, in which 5 are fixed entire applications and 3 are dynamic created by the user.

- Table of Story (Dynamic)
- Table of Character (Dynamic)
- Table of Paragraph (Dynamic)
- Table of Plotline (Fixed)
- Table of Roles (Fixed)
- Table of Archetype (Fixed)
- Table of Phase (Fixed)
- Table of Linkage (Fixed)

**Story (Dynamic)**

This is populated by the user. Name of Story New Story Created. In a created Story, it contains the Paragraphs created by the user through forms, the Characters created by the user through forms and the plot lines chosen by the user which is fixed.

**Character (Dynamic)**

Contains Name generated by User Can choose Role (FIXED) Paragraph contains characters In the phases or mini subplots assigned to different character archetypes Saved under one Story

**Paragraph (Dynamic)**

Text Based Contains Paragraphs the user creates

**Plotlines(Fixed)**

Has 7 Basic Story Archetypes by Christopher Booker It is fixed so select based

**Role (Fixed)**

Has 7 Abstract Character Functions by Vladimir Propp Roles are fixed List of all the character roles Select Based What role needs to be displayed in plot line and phase

**Archetype (FIXED)**

Has 12 Jungian Archetypes by Carl Jung Archetypes are fixed List of all the archetype roles select based Under character role

**Phase (Fixed)**

Has 7 Basic Story Archetypes by Christopher Booker and Story Empire Blog by Staci Troilo Contains 34 subplots Depending on the plot line chosen associated subplots will be used Characters created (roles) interact with subplots (1 character for 1 subplot) Dif Perspective Interacts with the Prose the user writes through their paragraphs

**Linkage (Fixed)**

3 Tables Plot, phase and role

## Rundown on how it will be connected

### Schema Design for Interactive Storytelling

Simple SQL schema Tables with Primary keys and foreign keys No complicated linkage part from 1 Table that has 3 keys(Phase, Role, Plotline)

Schema divides into data that is completely Fixed (entire application) and Dynamic (created by the user). There are 4 completely fixed and 3 dynamic.

Table of Plotlines (7) Potential plots and templates Completely fixed user is required to pick a plot line

Character of a story There is a role for the character the roles are completely fixed

As the story is written it goes through a set of phases and these phases are completely fixed

Dynamic in the schema You have stories Create story with a name The stories have to pick

from a plot line Many to one relationships between story and plotline

Plotline has an ID the story has a foreign key linkage into the Plotline As stories are built up there will be one plot line per story, potentially several stories could be on the same plot line(made by same user or made by someone different)

Stories have characters Stories can have more than one character Every character has a preset role Foreign key character that goes to primary key in role (Potentially more than one character with a single role) More than one hero and more than one villain, and might have no character for a particular role

Linkage from character to story which is many to one relationship Story can have 1 or more characters Many to one linkage from character to role because every character has a role and the roles are preset

Having set that up the user then gets to start writing paragraphs of text, got a table called prose where each entry in the table is a block of text Every text paragraph is written from the point of view of a character in a certain role

And the narrative voice switches between characters or between roles as the story progress

Linkage needed If we link from each block/ entry of prose into the character whose voice is speaking it that is many to one because the character may actually drop into the story more than once and have more than one block of text from character we know we can get to the role of the character and we can get to the story and we can get to the plotline

These block of prose each associated with a phase of the story and again the phases are completely fixed and you work through phases as the stories get written The phases vary according to plot line, different plot line have to have different phases and each phase is associated with a role so the first phase is a certain role like hero, second could be villain, third could be something else. So when I say a Line of text is written from the point of view of a character it has to be a character of the correct role for the story according to the phase of the story

Potentially when you are writing from a certain block of text you know what role it has to be Written in, which voice, because we know which phase of the story we are in because we are progressing through the phase if there is one character who has that role then you are telling the story from that characters point of view, if there is more than one character it could be a choice given to the user, right you are now in phase three you want to write from the role of the villain and you have two villains so you need to pick which one you want to write from

Then the remaining bit of the Schema which we haven't talked about is How we link plot lines roles and phases Given a plot line, that tells us what phases we go through and then

for every phase we have a role for the character whose voice we are going to use. We have a table here which is a table of 3 , foreign keys for plot, phase and role.

Given the plot line you can filter that table down, so that gives you phases and roles, as you work through the phases that gives you the role you need to pick out for that phases and plot line in order to allow selection of the character and to have that block of text entered.