Team bendyStraws - Kevin Cai, Tammy Chen, Kazi Jamal & Lauren Pehlivanian
SoftDev1 pd9
P #00 — Task the First - The Plan
2019-10-21

Odyssey by Team bendyStraws

Project Objective:

The overall goal of the project is to create a collaborative storytelling game/website where registered & logged-in users can use the website to create new stories or add to existing ones.

List of Required Program Components:

Content:

- Home Page: The home page will only be accessible once the user is logged in. It will display a list of stories that the user has contributed to and can be fully read. There will also be two links/buttons: one to start a new story, and one to add to an existing story.

    *Data needed:*The complete listing of stories that exist(story-edits).
- Stories to Contribute to Page: The Stories to Contribute Page will contain a list of all the stories the user hasn't contributed to, and the story titles listed would be links to edit the story.

    *Data needed:*The complete listing of stories that exist(story-edits).

Actions:

- Read: The user can select and read the stories that they have already contributed to from the home page.

    Data needed: The entire text of the story (all contributions). This is done by iterating through the content field of the table for the story which the user is currently viewing.
- Add to Story: The user can read the latest update to the story, and then add some amount of text to the story.

    Data needed: The most recent contribution to the story, which is the content field of the last record in the table for the story which the user is currently adding to.

    Data created: The text this user contributes to the story will be stored in a new record in the story-edits table
- Create Story: The user will need to input some text for the initial content of the story and for the title.

    Data created: A new record will be added to the Story Masterlist with the name and a formatted name. Then a new table will be created for the story, and the content of the story and the author will be stored as the first record in this new table. Also, a new record will be added to each Individual User Table, and the Edited? field will be set to 1 for the user that created the story, and 0 for every other user.

User Accounts:

- All users must register to use the site.

Data needed: Usernames and passwords from users table.

- If the user is not logged in, they will be directed to the login page.
    - On the login page there is an option to sign up.

      Data created: A new record in the users table. In addition, a new Individual User Table will be created for this user with only a single column to contain the story IDs they edited.

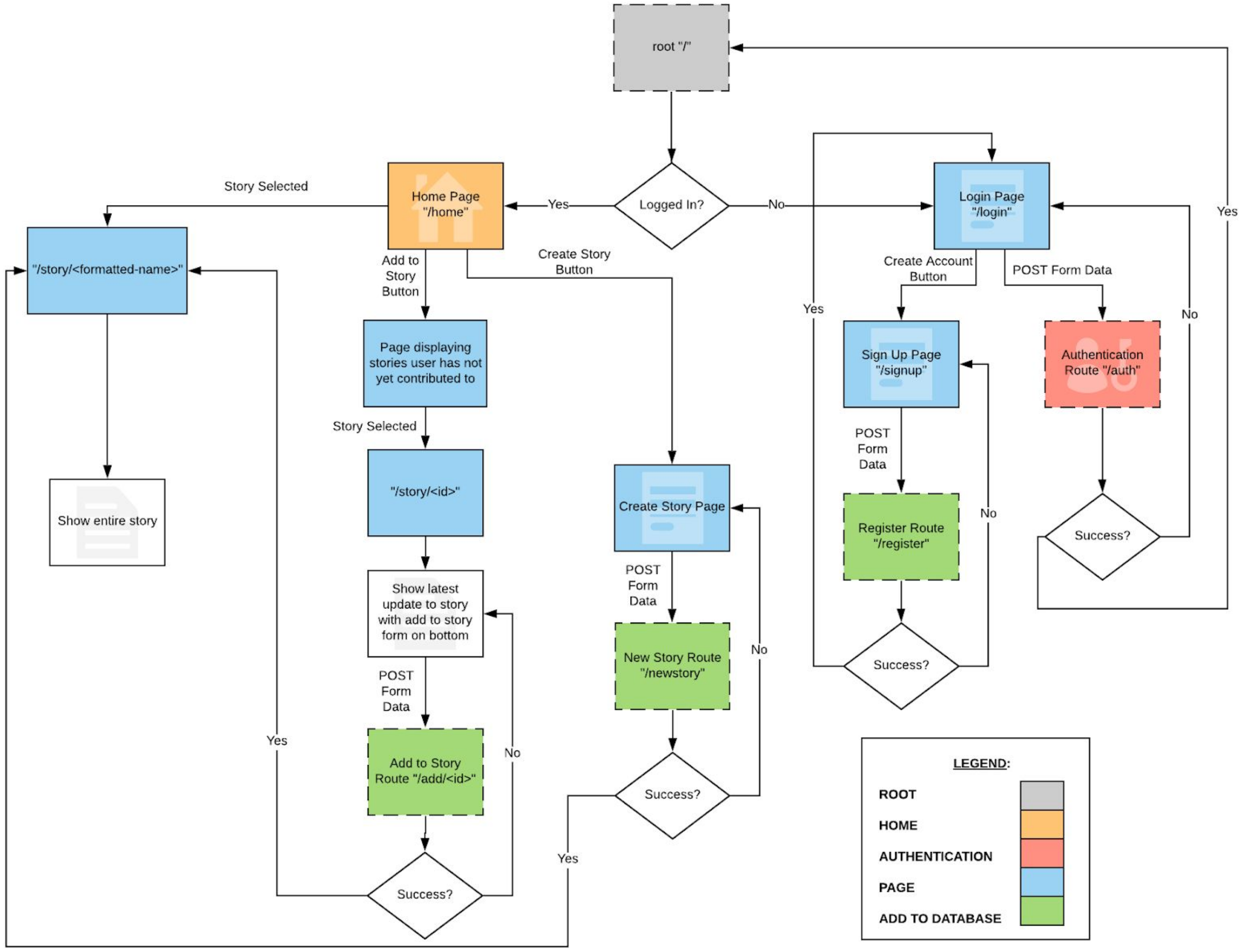
Optional Components:

- navigation bar (link back to home page, logout button)
- search bar
- upvotes and timestamps
- hashing passwords

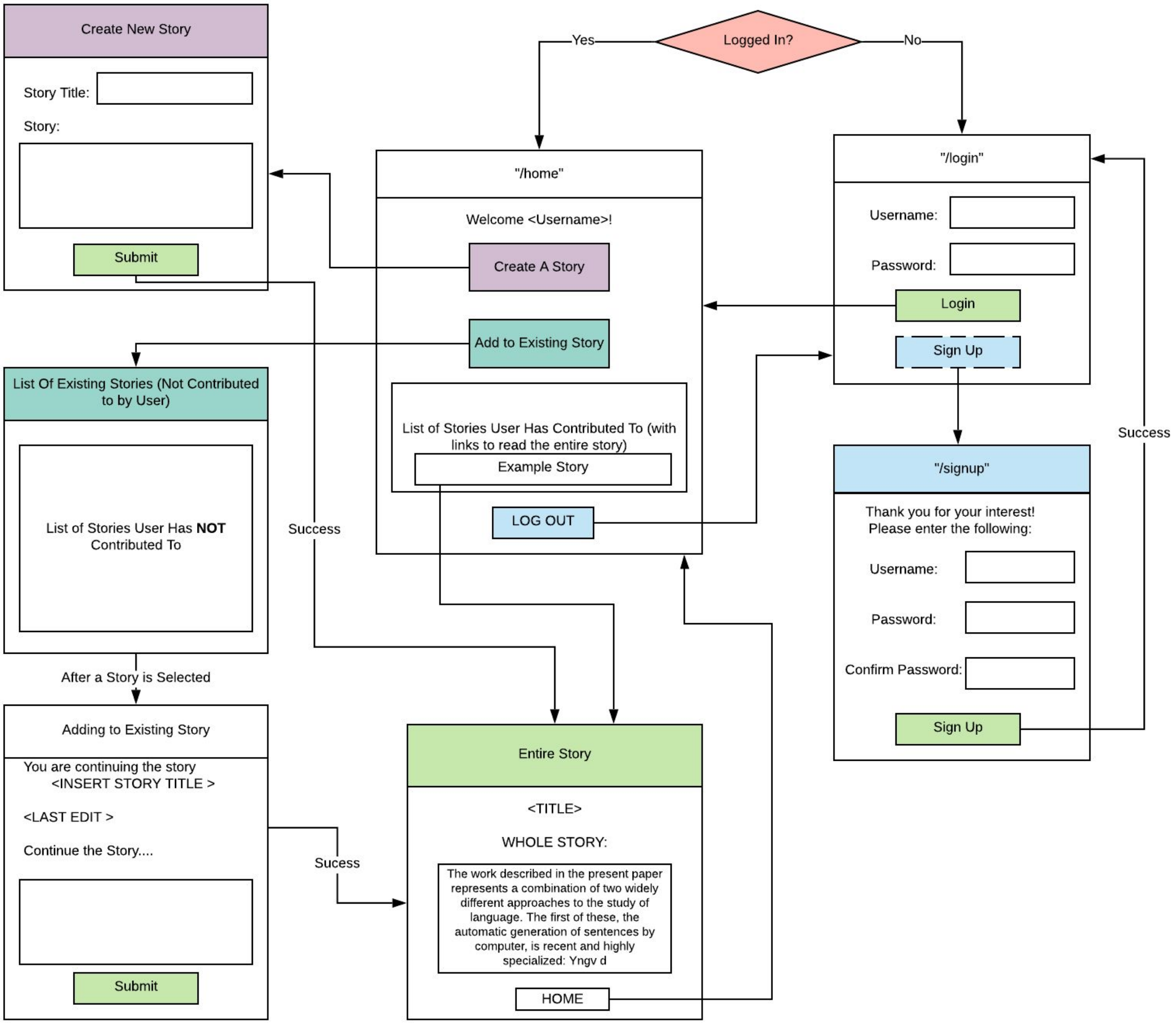
Database Tables:

1. Story Masterlist: The masterlist of all stories, each with a unique ID number.
2. Story-edits Table: Each edit on the site will have a unique ID based on timestamp, and contain the story ID of the story edited, and the User ID of the editor, as well as the author’s edit.
3. User Info: Stores users’ usernames and passwords.

Site Map:


root "/"
Logged In?
Yes
No
Home Page "/home"
Story Selected
"/story/<formatted-name>"
Add to Story Button
Create Story Button
Page displaying stories user has not yet contributed to
Story Selected
"/story/<id>"
Show latest update to story with add to story form on bottom
POST Form Data
Add to Story Route "/add/<id>"
Success?
Yes
No
Show entire story
Create Story Page
POST Form Data
New Story Route "/newstory"
Success?
No
Yes
Login Page "/login"
Create Account Button
POST Form Data
Sign Up Page "/signup"
POST Form Data
Register Route "/register"
Success?
Yes
No
Authentication Route "/auth"
Success?
No
Yes
LEGEND:
ROOT
HOME
AUTHENTICATION
PAGE
ADD TO DATABASE

Site Components:



The work described in the present paper represents a combination of two widely different approaches to the study of language. The first of these, the automatic generation of sentences by computer, is recent and highly specialized: Yngv d

Database:

stories:(This is the main database)

storyID PRIMARY_KEY	Title
0	“Hello world”
1	“A Cool Story”
2	“Why Softdev is cool”
3	“How to learn Java”

story-edits:

storyID	userID	Content	Timestamp
0	0	“This is the beginning ..."	2019-01-08 8:20:05.123
1	1	“Once upon a time ...”	2019-02-13 9:21:05.124
2	0	“The beginning ...”	2019-03-21 10:22:05.523
1	3	“A chicken crossed the road”	2019-04-23 11:23:06.123
2	2	“To the end”	2019-05-25 12:34:05.123

users:

userID PRIMARY_KEY	Username	Password
0	“TheLastStraw”	1234
1	“bendyStraws”	pleasehackthisaccount
2	“StrawBerry”	password
3	“StrawsAreBendy”	password1234

Database Operations

Seeing stories the user did not edit:

```
SELECT stories.storyID, userID
FROM stories
LEFT JOIN story-edits WHERE userID = <desired-user-id> USING(storyID)
WHERE userID IS NULL
```

Seeing stories the user edited:

```
SELECT storyID FROM story-edits WHERE userID = <desired-user-id>
```

Seeing all contributions to a story in order:

```
SELECT * FROM story-edits WHERE storyID = <desired-story-id>
ORDER BY datetime(timestamp) DESC
```

Viewing most recent edit of story:

```
SELECT * FROM story-edits WHERE storyID = <desired-story-id>
ORDER BY datetime(timestamp) DESC LIMIT 1
```

Creating a new story:

```
INSERT INTO stories VALUES (NULL, <title>)
currID = SELECT storyID from stories where title = <title>
INSERT INTO story-edits VALUES (currID, <userID>, <content>, datetime(‘now’))
```

Adding to a story:

```
INSERT INTO story-edits VALUES (<storyID>, <userID>, <content>, datetime(‘now’))
```

Registering a new user:

```
INSERT INTO users VALUES (NULL, <username>, <password>)
```

BREAKDOWN OF TASKS

> Kevin Cai (KC)
> Tammy Chen (TC)
> Kazi Jamal (KJ)
> Lauren Pehlivanian (LP)

I. Minimum Viable Product

A. (TC) Authentication: Manage creating new users, and authenticating users when they want to log in.

B. (LP) Home Page and Reading Stories: Create home page, manage displaying stories user has contributed to and the interface for reading those stories.

C. (KJ) Creating Stories: Create page for creating new stories with a form, route for handling creating a new story, and manage adding new stories to the database.

D. (KC) Adding to Stories: Create page for displaying stories which the user has not contributed to with links to add to each one, route for handling adding those stories, and manage the process of adding new records for edits to the database.

II. Extra Features (not delegated yet, we will see what time allocates us to do)

A. Navigation Bar: Create a navigation bar with link back to home page and button to log out.

B. Search Bar: Create a search bar to search through stories with key words.

C. Upvotes and Timestamps: Add upvotes to stories on the home page, and timestamps to contributions.

D. Hashing Passwords: Hash passwords before storing them in the database for increased security

ROLES

> Kevin Cai (KC): Adding to Stories System Manager
> Tammy Chen (TC): Project Manager, Authentication System Manager
> Kazi Jamal (KJ): Creating Stories System Manager
> Lauren Pehlivanian (LP): Home Page and Reading Stories System Manager