

Bloh Foo:: Ameer Alnasser, Wan Ying Li, Kevin Wang
SoftDev
P00 -- wiki design
2022-10-27
time spent: 1 hrs

DB organization:

Tables: (Kevin)

Authentication

- Username
- Password

Wiki

- Title
- Content
- Date created
- Date updated
- Recent contributor's username

Program components:

- Home page (Wan Ying)
 - Front end
 - Nav bar: links to
 - Login and sign up
 - create new wiki
 - Search
 - Table of content/alphabetically sorted directory
 - Back end
 - Return a random wiki to show on the home page
- Login Page (Ameer)
 - Front end
 - Form for user to submit username + password
 - Button to submit
 - Forgot password link to sign-up Page
 - Back end
 - Check with database if username and password are associated with account
 - True:
 - Redirect to home page, with username and password as apart of request.form

- Every link they press will be sent with username and password in the request.form
 - False:
 - Refresh the page?
- Sign up Page (Ameer)
 - Front End
 - Username
 - password
 - Backend
 - Makes sure username is unique
 - True
 - Make new account
 - Sign in with that account
 - Redirect to home page
 - False
 - Refresh the page
- Create new page (Wan Ying)
 - Front end
 - Form
 - Title
 - Content
 - Button to Submit
 - Back end
 - Check if logged in.
 - True: display page to user
 - False: direct user to login in page
 - When submit pressed:
 - Is title unique
 - True: push page into RDB
 - False: scrap everything they've written and refresh the page
- Search (nav bar) (Kevin)
 - Front-end
 - Wiki page titles that start with what was searched
 - Back-end
 - Query RDB for titles that start with the searched string
- Directory (Kevin)
 - Front-end
 - Alphabetically sorted list of wiki pages
 - Back-end
 - Get all of the titles and use python to sort them before displaying it
- Wiki page (Kevin)

-
- ```
graph TD
 subgraph WB [web browser]
 HP[home page]
 LIP[login in Page]
 CNP[create new Page]
 SUp[sign up Page]
 D[Directory]
 WP[Wiki Page]
 end
 HP --> LIP
 HP --> CNP
 HP --> SUp
 HP --> D
 LIP --> ILI{is logged in?}
 CNP --> ILI
 ILI -- Yes --> DB
 ILI -- No --> LIP
 SUp --> IUS{is unique username?}
 IUS -- Yes --> DB
 IUS -- No --> SUp
 DB --> Auth[Authentication]
 DB --> Wiki[Wiki]
 Auth --> ILI
 Wiki --> IUS
 Wiki --> WP
```
- The diagram illustrates the interactions for a Wiki system. The **web browser** (boundary) contains several use cases: **home page**, **login in Page**, **create new Page**, **sign up Page**, **Directory**, and **Wiki Page**. The **home page** use case initiates actions to **login in Page**, **create new Page**, **sign up Page**, and **Directory**. The **login in Page** and **create new Page** use cases lead to the **is logged in?** decision (control). If the user is logged in, the flow goes to the **Database**; otherwise, it loops back to **login in Page**. The **sign up Page** use case leads to the **is unique username?** decision (control). If the username is unique, the flow goes to the **Database**; otherwise, it loops back to **sign up Page**. The **Database** (entity) is connected to two data stores: **Authentication** and **Wiki**. The **Authentication** data store contains attributes: Username, Password, and is linked to the **is logged in?** decision. The **Wiki** data store contains attributes: Title, Content, Date created, Date updated, and Recent contributor username, and is linked to the **is unique username?** decision and the **Wiki Page** use case.

