

ANNALES MEMUM

By Samuel Pearce and Julian Werner

CONTENTS

1 - Project Concept.....	2
2 - Requirements.....	2
2.1 - Functional Requirements.....	2
2.2 – Non-Functional Requirements	2
3 – Development.....	3
3.1 -	Fehler! Textmarke nicht definiert.

1 - PROJECT CONCEPT

The idea behind the Annales Memum is to be a chronicle of “memes”. These are images, videos, stories, etc. That are shared throughout the internet virally. We believe that, due to them being often misunderstood, it is important to keep a record online of the history of these strange, but entertaining ideas. We can use the site to preserve history for future generations to understand internet culture, to keep statistics records, or just to reflect on an idea’s impact to society.

The website will be a custom wiki that allows users to create/edit pages themselves with no planned moderation. Should the project become successful, a long-term moderation plan should be created mimicking that of Wikipedia. Each page should document three main aspects: a meme’s origin/history, a meme’s popularity i.e. what media it was popular on and how many people were exposed to it, and a meme’s cultural impact. The impact should describe whether the meme had a lasting effect, e.g. “Pepe” had an extremely potent effect and lasted far past its due expiration date, but “Sirenhead” had a pretty standard lifespan.

Another aspect of meme-culture to be explored and documented on the site are the theories and sciences pertaining to the study of memes. This would encompass things like the theory of meme lifespan which seek to allow the philosophical discussion about memes.

2 - REQUIREMENTS

2.1 - FUNCTIONAL REQUIREMENTS

- Users can create accounts.
- A user with an account can create new pages.
- A user with an account can edit pages.
- Each page should document the history and public opinion about a meme.
- Meta-pages may exist.
- The users may use markdown to style their pages.

2.2 – NON-FUNCTIONAL REQUIREMENTS

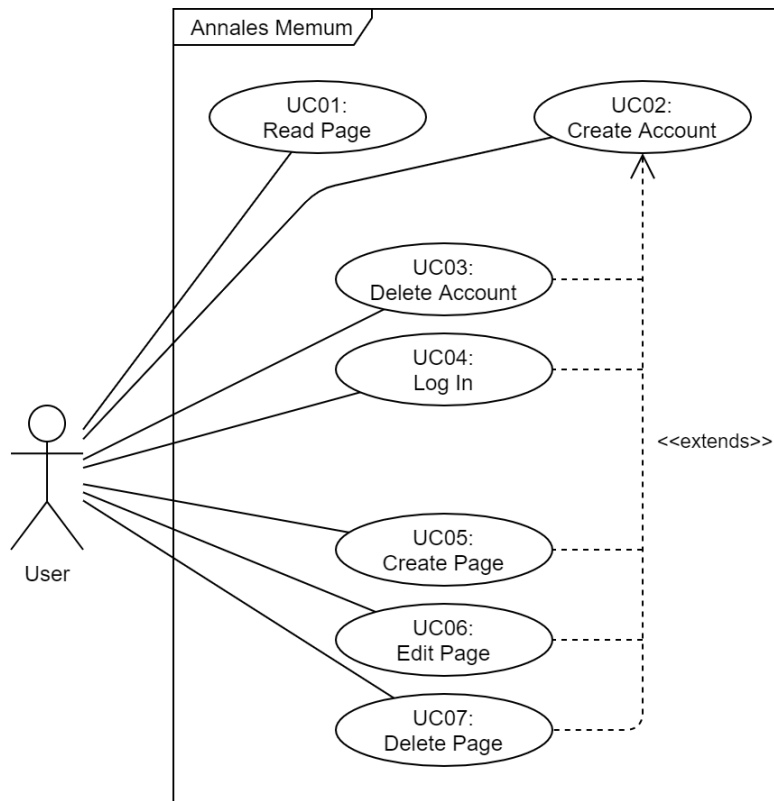
- The project will use a Spring backend.
- The project may be deployed using Docker containers.
- The project will use MySQL and PostgreSQL for the data layer.
- The front-end will use HTML, CSS, and JS.

2.3 – NICE-TO-HAVES

- Users can translate a page into multiple languages.
- Users can write a short description about themselves on a personal page.
- Users can earn the privilege to higher moderation rights and only higher-level users may edit protected pages.
- Pages can be protected by higher-level users to prevent grieving via deletion or modification.
- Admins can restore deleted pages.

3 – USE-CASES

3.1 – USE CASE DIAGRAM



3.2 – USE CASES

ID	UC01
Title	Read Page
Preconditions	- The user is on the website's main page.
Main Flow	1. The user enters a search term into the search box. 2. The user selects one of the found pages. 3. The page content is displayed to the user.
Possible Exceptions	- Page not found: 1. The user is informed that no results were found. 2. Proceed as above.
Postconditions	- The user is on the page they wanted.

ID	UC02
Title	Create Account
Preconditions	- The user is on the website.
Main Flow	1. The user clicks on the "Create Account" button. 2. The user enters the required details. 3. The user is redirected to the main page and a success message is displayed.
Possible Exceptions	- Invalid input Data: 1. The user is informed their details are invalid. 2. Proceed as above.
Postconditions	- The user's account has been created.

ID	UC03
Title	Delete Account
Preconditions	<ul style="list-style-type: none"> - The user is on the website. - The user is logged in.
Main Flow	<ol style="list-style-type: none"> 1. The user selects the dropdown next to their name. 2. The user selects the "Delete Account" option. 3. The user enters their password and confirms they wish to permanently remove their account. 4. The user is redirected to the main page and a success message is displayed.
Possible Exceptions	<ul style="list-style-type: none"> - Invalid Password: <ol style="list-style-type: none"> 1. The user is informed their credentials were invalid. 2. Proceed as above.
Postconditions	<ul style="list-style-type: none"> - The user's account has been permanently deleted.

ID	UC04
Title	Log In
Preconditions	<ul style="list-style-type: none"> - The user is on the website. - The user has an account. - The user is logged out.
Main Flow	<ol style="list-style-type: none"> 1. The user clicks the "Log In" Button and enters their account credentials. 2. The user is redirected to the main page and a success message is displayed-
Possible Exceptions	<ul style="list-style-type: none"> - Invalid Password: <ol style="list-style-type: none"> 1. The user is informed their credentials were invalid. 2. Proceed as above.
Postconditions	<ul style="list-style-type: none"> - The user is logged in

ID	UC05
Title	Create Page
Preconditions	<ul style="list-style-type: none"> - The user is on the website. - The user is logged in.
Main Flow	<ol style="list-style-type: none"> 1. The user selects the dropdown next to their name. 2. The user selects the "Create Page" option. 3. The user enters the title and content for the new page. 4. The page is created, and the user is redirected to the new page.
Possible Exceptions	<ul style="list-style-type: none"> - Invalid Page Data: <ol style="list-style-type: none"> 1. The user is informed that their chosen data is invalid and asked to try again. 2. Proceed as above.
Postconditions	<ul style="list-style-type: none"> - The new page is created and publicly visible.

ID	UC06
Title	Edit Page
Preconditions	<ul style="list-style-type: none"> - The user is on the website. - The user is logged in. - The user is on any editable page.
Main Flow	<ol style="list-style-type: none"> 1. The user clicks on the "Edit" Button. 2. The user is shown an editor with the current page contents where they make their changes. 3. The user clicks the "Save Changes" Button. 4. The user is redirected to the page and a success message is displayed.
Possible Exceptions	<ul style="list-style-type: none"> - Invalid Page Data: <ol style="list-style-type: none"> 1. The user is informed that their chosen data is invalid and asked to try again. 2. Proceed as above.
Postconditions	<ul style="list-style-type: none"> - The page has been updated with the new content.

ID	UC07
Title	Delete Page
Preconditions	<ul style="list-style-type: none"> - The user is on the website. - The user is logged in. - The user is on any editable page.
Main Flow	<ol style="list-style-type: none"> 1. The user clicks on the "Delete" Button. 2. The user is prompted for their credentials and asked to confirm they want to delete the page. 3. The user is redirected to the main page and a success message informs them the page was successfully removed.
Possible Exceptions	<ul style="list-style-type: none"> - Invalid Credentials: <ol style="list-style-type: none"> 1. The user is informed that their credentials invalid. 2. Proceed as above.
Postconditions	<ul style="list-style-type: none"> - The page has been marked as deleted. (NOT Deleted from the database though.)

4 – MOCKUPS

