

Bioethics Project

Building Interactive Cases

Nov. 22, 2016

David Wauchope • 902 830 7251 • david.wauchope@dal.ca

Devon Rimmington • 902 989 1285 • dv728483@dal.ca

Faye Teeuwen • 902 221 2520 • faye@dal.ca

Matt Smith • 902 449 2868 • matthew.smith@dal.ca

Table of Contents

Description	2
Project Summary	2
Scope	3
Planned User Stories	3
Completed User Stories	3
Future User Stories	3
Implementation Scope	4
Outstanding Issues	5
Integrating Twine stable version with the application	5
Popup dialogue	5
Deliverables	5
Modified Twine Application	5
Database Diagram for Integrating Twine	6
Bootstrap Template for Displaying Cases	6
Reasons for Another Group to Choose this Project	7
Closure Agreement	7

Description

Our client, the Bioethics department of Dalhousie University, is looking for a web-based resource that simulates ethical consultation for the students in its faculty. They want a prototype to be made so that they can present it to their textbook manufacturer for them to fill with certain ethical problems that learning health professional may face. This prototype should contain story-like situations where the user can traverse back and forth through the story, as well as link to other problems should they need further information.

Project Summary

This Bioethics project goal was to create a non-linear interactive learning tool. This tool is being created for use by studying medical students and practicing health professionals. When fully completed, it will include interactive stories aimed at enhancing their understanding of ethical situations and how to navigate those ethical decisions.

During this phase of development, as the second group to work on this project our first goal was to determine the possibility of a solution that allows administrators to create non-linear stories. Assuming that a solution was found, the rest of our goal consisted of developing an API, code structure, and a database schema to support the client's needs.

Scope

Planned User Stories

1. As a reader, I want to read cases so that I gain knowledge.
2. As an editor, I want to easily create cases so that users can interact with them to find a solution.
3. As an editor, I want to write choices so that choices can lead to other choices.
4. As an editor, I want to be able to edit stories as bioethical solutions shift externally to the website.
5. As a reader, I want to be able to navigate backwards and forwards through a case so that I am not punished for my choices.
6. As an editor, I want to have clear documentation so as to easily learn how to use the website and Twine.
7. As a reader, I want feedback on the quality of my choices.
8. As an editor, I want to be able to bundle stories in collections to keep stories organized to the reader and I.
9. As a reader, I want to be able to view similar cases after I finish a case.
10. As a reader, I want to be able to contact the editor should I need further information.
11. As a reader, I would like to be able to flag questions as confusing or missing context/knowledge.

Completed User Stories

1. As a reader, I want to read cases so that I gain knowledge.
2. As an editor, I want to easily create cases so that users can interact with them to find a solution.
3. As an editor, I want to write choices so that choices can lead to other choices.
4. As an editor, I want to be able to edit stories as bioethical solutions shift externally to the website.
5. As a reader, I want to be able to navigate backwards and forwards through a case so that I am not punished for my choices.
6. As an editor, I want to have clear documentation so as to easily learn how to use the website and Twine.

Future User Stories

7. As a reader, I want feedback on the quality of my choices.
8. As an editor, I want to be able to bundle stories in collections to keep stories organized to the reader and I.
9. As a reader, I want to be able to view similar cases after I finish a case.

10. As a reader, I want to be able to contact the editor should I need further information.
11. As a reader, I would like to be able to flag questions as confusing or missing context/knowledge

Implementation Scope

Planned Feature	Actual Feature	Why we changed it
The original plan included a Django framework	We are now recommending the project to continue development in PHP	We integrated Twine; PHP will allow development to be much smoother when integrating other applications that require unique object-relational management
Simple front-end for the user to interact with built in Django	We have built a template file in Django for the front end to use, but has not been implemented yet	The template will have to be modified from Django-specific syntax to PHP-specific syntax in a later iteration of this project
Interactive case backend	We implemented Twine to handle cases and case data	Twine is much simpler to use, includes a well built UI, and fits the project requirements better than any backend we would have been able to build in the allotted timeline

Outstanding Issues

Integrating Twine stable version with the application

- The current version of the project uses the development version of Twine
- The development version was used for its Javascript code base, as opposed to the old stable release in Python
- When Twine releases a stable version of Twine with a Javascript code base, we recommend that this be implemented to replace the currently used Twine beta

Popup dialogue

- When publishing a user case to the server, a popup dialogue appears
- This dialogue displays some placeholder text, in the next iteration of this project, that text should be replaced to give users more accurate information
- Save popup dialog text can be found and modified at the following file path:
twine/src/file/save.js

Deliverables

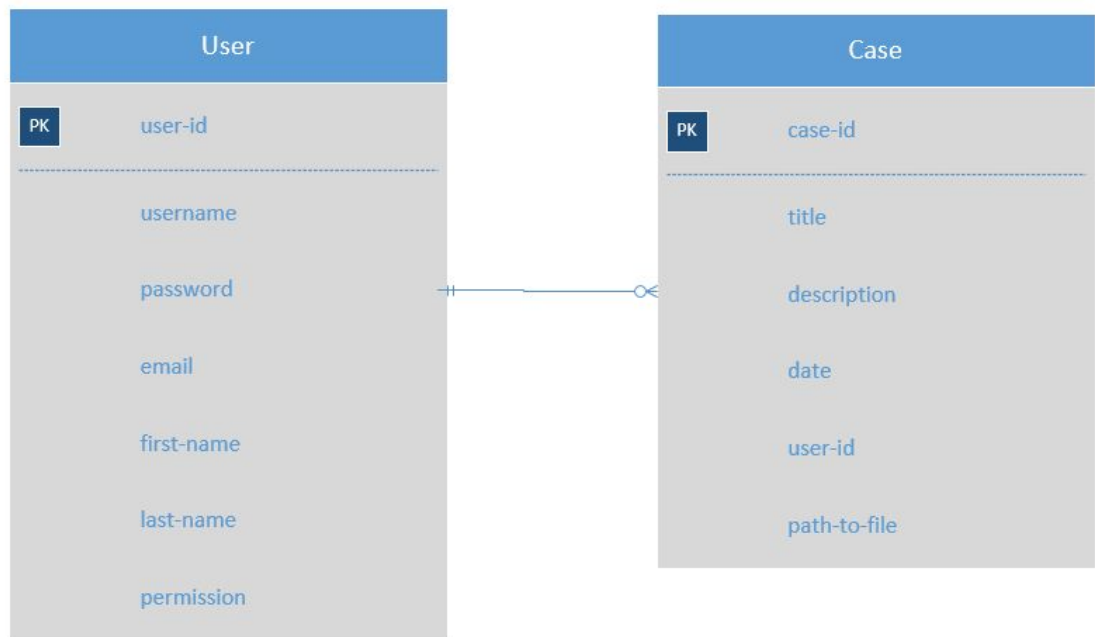
Modified Twine Application

- Modified version of the open source interactive story generator “Twine”
 - Addition of publish to server button
 - Addition of saving to file on remote server functionality

Database Diagram for Integrating Twine

- Crow's Foot database diagram that can be used for the next iteration.

Twine DB Diagram



Bootstrap Template for Displaying Cases

- A template for displaying content on a webpage.
- Displays cases as blocks within a subsection, replacing Django tags with content from the database.
- Designed for Django, but can easily be modified to work with with PHP by replacing Django syntax with PHP syntax.

Reasons for Another Group to Choose this Project

This project offers the opportunity to create a website using PHP. There are four main components which the next group will have an opportunity to work on. Most of these components are applying course-learned skills in a real-world situation.

1. Creating a database using the provided schema.
2. Creating a front-end informational website using PHP.
3. Creating an authorization system
4. Integrating Twine into the website and database.

Closure Agreement

The goals completed were based on the specified scope found in the initial Project Plan document. The original goal of creating a working backend for creating interactive cases was completed. No further goals were accomplished. The client and group members hereby confirm the completion of the project and hand all documentation and code over to the client, Marika Warren. The group members each agree to pass on their work to be edited and used in future courses. The client agrees that the work completed is satisfactory.

Project Client

Marika Warren

Student Development Group Leader

David Wauchope