

North Dakota State University

Graveyard Management System (GMS)

Students:

Joshua Heeren
Noah Huesman
Ashton Hoeft
Gavin Kestner

Instructor:

Pratap Kotala

12 April 2024

Table of Contents:

Project Overview:	3
I. Github Repository:.....	3
II. Our Team:.....	3
III. Reflection:.....	3
Project Management:	4
I. Timeline of Trello Board:.....	4
II. Project Changes:.....	5
III. Sprint Tasks:.....	5
a. Sprint #1 Backlog:.....	5
b. Sprint #2 Backlog:.....	5

Graveyard Management System (GMS)

By: JANG Tech.

Project Overview:

I. Github Repository:

<https://github.com/nhuesman1043/GMS>

II. Our Team:

- A. Ashton Hoeft (Implementation Phase Team Leader)
- B. Noah Huesman
- C. Joshua Heeren
- D. Gavin Kestner

III. Reflection:

Our team had an overall good experience learning and using the Angular software framework. Developing the Graveyard Management System (GMS) application with an Angular frontend and a Django backend led us to many challenges that we were able to overcome. The team learned a lot of new skills that will be useful in our future software development careers.

The main differences we found from working with other platforms were combining a backend framework with a frontend framework, working with components instead of object oriented programming, and using asynchronous operations. We also found that the learning curve for a framework is much higher than learning a library. This is seen in Angular vs React, as some of our team members have prior experience with the React.js libraries. Working with Angular has introduced our team to modern web-development practices that emphasize modularity and maintainability. It has taught us how to structure a medium scale application and how to manage dependencies efficiently.

In conclusion, while our team faced many challenges and faced a steep learning curve, working with Angular and Django has equipped us with valuable skills in modern web development, component based architecture, and full stack development. These experiences have prepared us for building robust and scalable applications in the future.

Project Management:

I. Timeline of Trello Board:

Starting Line	End of Sprint #1
End of Sprint #2	Finish Line

II. Project Changes:

- a. We are not going to use a plot component and instead loading the plots will be done in the map component.
- b. In our models, we added a plot_identifier field to our Plot model.
- c. Since we did Angular and Django, we used components instead of templates.

Below is the list of our components.

1. Header
 2. Weather-widget
 3. Login
 4. Map
 5. Sidebar
 6. Sexton-sidebar-content
 7. User-sidebar-content
- d. We decided not to implement the ability to create new plot statuses.
 - e. We decided to use the google api for the map instead of a screenshot
 - f. There is no list view of the plots

III. Sprint Tasks:

a. Sprint #1 Backlog:

1. Set up a basic application template (**Start & Finish**).
2. Set up Django project structure (**Start & Finish**).
3. Define database models (**Start & Finish**).
4. Implement Django REST Framework API (**Start & Finish**).
5. Integrate Angular frontend (**Start & Finish**).
6. Setup basic map component (**Start & Finish**).
7. Implement authentication and authorization (**Start**).
8. Format user content for viewer sidebar (**Start**).
9. Set up a plot component for the frontend map (**Start**).

b. Sprint #2 Backlog:

1. Implement authentication and authorization (**Finish**).
2. Format user content for viewer sidebar (**Finish**).
3. Set up a plot component for the frontend map (**Finish**).

4. Create Header/Info/Weather components (**Start & Finish**).
5. Create a generic user sidebar view (**Start & Finish**).
6. Create Sexton sidebar CRUD view (**Start & Finish**).
7. Create login and logout functionality (**Start & Finish**).
8. Finish the custom plot markers and map (**Start & Finish**).