

# **North Dakota State University**

---

## **Graveyard Management System (GMS)**

---

**Students:**

Joshua Heeren  
Noah Huesman  
Ashton Hoeft  
Gavin Kestner

**Instructor:**

Pratap Kotala

12 April 2024

<b>Project Overview:</b>	<b>2</b>
I. Our Team:	2
II. Project Description:	2
III. Project Purpose:	2
IV. Scope Changes:	3
<b>Project Management:</b>	<b>3</b>
I. Timeline of Trello Board:	3
II. Sprint Tasks:	4
III. Sprint Tasks:	4
IV. Communication Methods:	4
<b>Project Design:</b>	<b>5</b>
I. Balsamiq Mockups:	5
Framework Specification:	7
I. Framework Description:	7
II. Django Apps:	7
A. API:	7

# Graveyard Management System (GMS)

By: JANG Tech.

## Project Overview:

### I. Our Team:

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

### II. Project Description:

The Graveyard Management System (GMS) project aims to develop and create a refined web application for Rosemound Cemetery, located in Barnesville, Minnesota, and overseen by St. James United Church of Christ. The application emphasizes a user-friendly interface, an interactive plot map, administrative tools, robust search capabilities, and privacy measures. The project will consist of two main web pages: one for public access, displaying all of the information such as a map of the plots with details of buried individuals, and another for administrator/Sexton access that allows inputs to manage existing and new plot data of deceased members of the church.

### III. Project Purpose:

The purpose of the GMS project is to help solve the problem of manual cemetery information upkeep by providing a streamlined and efficient digital solution for managing and displaying Rosemound Cemetery's vast amount of information. Users of this web application will be able to gather detailed information about the plots in Rosemound Cemetery. One notable limitation involves the reliance on manual data input and maintenance by authorized users, introducing the potential for human error and requiring consistent effort in data upkeep. However, the GMS project also aims to mitigate this limitation by implementing features that streamline the data entry processes.

## IV. Scope Changes:

One scope change that we are looking into making is how we are going to implement the interactive graveyard map. Our initial plan was to use Google Maps' free API for displaying our graveyard and overlaying our plot map over it. We may use a different approach and instead, just use a satellite screenshot of the graveyard to create our interactive map. This will make the process of developing an interactive map simpler. Other than that, we have no other scope changes at this time.

# Project Management:

## I. Timeline of Trello Board:

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

## II. Sprint Tasks:

Our team used weekly meetings to plan and divide tasks accordingly. Our first meeting was used to discuss and create our product backlog. To create our product backlog, our team carefully went through the provided project requirements guidelines document and identified key tasks of this phase. During our second meeting, the division of tasks was laid out. This was done by first asking if any team members had a preference for their assigned tasks and then distributing the remaining tasks based on current workloads. The following meetings consisted of going through our project backlog and sprint planning/revision.

## III. Sprint Tasks:

### A. Sprint #1 Backlog:

1. Develop Mockups (**Start**).
2. Wireframe Home Page (**Start & Finish**).
3. Wireframe Sexton (Admin) Page (**Start & Finish**).
4. Identify scope changes (**Start & Finish**).
5. Identify a list of Django apps (**Start & Finish**).
6. Make Models (**Start & Finish**).
7. Manage project (**Start**).
8. List user stories (**Start**).

### B. Sprint #2 Backlog:

1. Manage project (**Finish**).
2. API for backend data (**Start & Finish**).
3. Identify a list of Django apps (**Start & Finish**).
4. Make Models (**Start & Finish**).
5. Format document (**Start & Finish**).
6. Revise and submit (**Start & Finish**).

## IV. Communication Methods:

A Discord server and weekly lunch meetings facilitated team communication during the project's requirements phase. Discord provided flexible communication, while in-person meetings ensured thorough discussions. This combination allowed for constant collaboration, leading to a timely project conclusion.

## Project Design:

### I. Balsamiq Mockups:

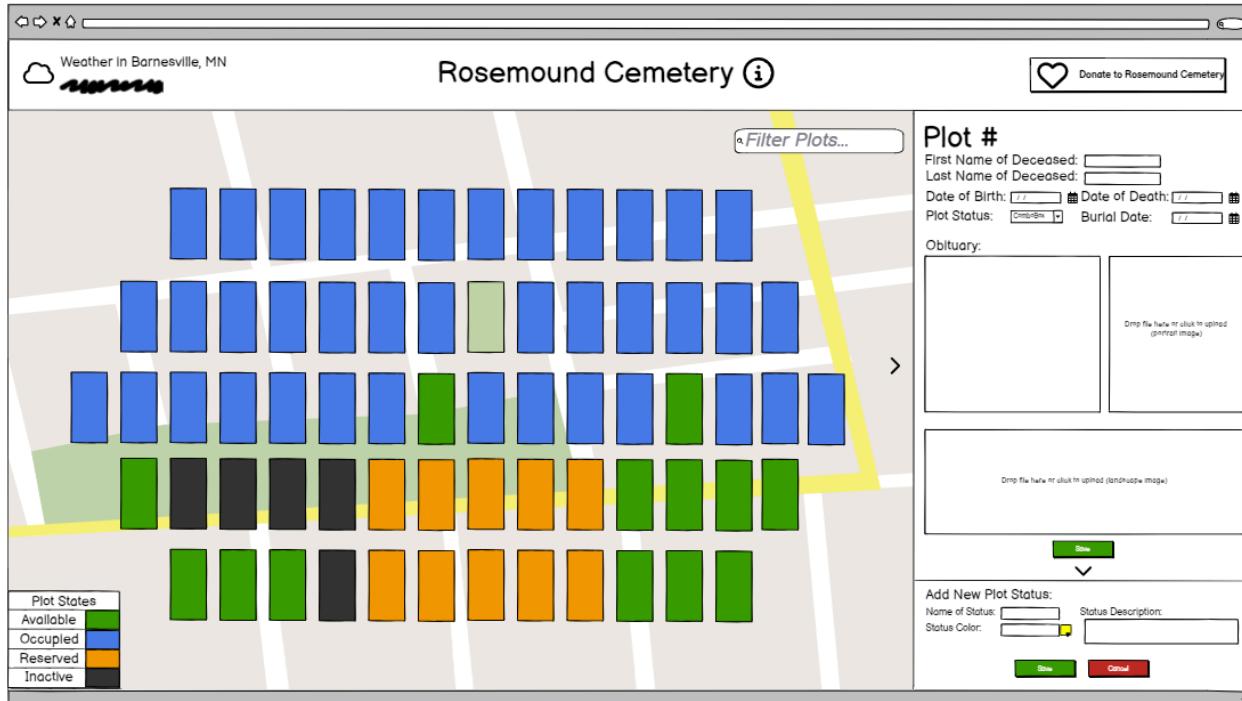
**Generic User Page**

The mockup illustrates a web-based interface for managing cemetery plots. At the top, there's a header bar with a weather icon ('Weather in Barnesville, MN'), the cemetery name ('Rosemound Cemetery'), and a 'Filter Plots...' button. A yellow arrow points from this button to a 'Plot #' section on the right. The central area contains a grid of plots, some highlighted in green (selected) or gray (available). A yellow arrow also points from the bottom of the plot grid to the same 'Plot #' section. The sidebar on the right provides details for a selected plot, including the deceased's name, important dates, and two small thumbnail images.

**Notes:**

- ❖ Current layout of plots is not accurate to - Currently waiting on the Sexton to give us an up-to-date map.
- ❖ Map in the background is a placeholder for either a satellite image or Google Map interface, this depends on our final project scope.
- ❖ Right side "folds" in and out depending on whether or not a plot is selected (green).
- ❖ When filtering plots, plots that do not match the filter criteria will be grayed out.
- ❖ The donate button will take users to St. James Church's donation page.
- ❖ The "i" button will open a small popup showing more information about the cemetery.
- ❖ Up-to-date weather info will be displayed in the top left.
- ❖ Plots that are not available to be selected are grayed out (Darker than plots not matching search).
- ❖ The images are not required, the layout of information will be dynamic based on Sexton input.

## Sexton Page



### Notes:

- ❖ Identical to Generic User Page, but the right side acts as a data entry section.
- ❖ Bottom of the right side allows for the Sexton to add additional states, which will be added to the legend.
- ❖ In the plot status dropdown, the option to add a new status will expand the Add Status section
- ❖ Plots are also different colors based on their status - The legend is in the bottom left.
- ❖ Selected plot is slightly opaque.

## Info Page

**Rules & Regulations of Rosemount Cemetery**

(Large block of text, mostly illegible)

---

**Contact Us**  
St. James Email: churchoffice.stjames@gmail.com  
St. James Phone: 218-354-7198

**Find Us**  
201 3rd Street SE  
PO Box 369  
Barnesville, Mn 56514

© 2024 Jong Tech USA, LLC All rights reserved.

## Framework Specification:

### I. Framework Description:

For the framework, we will be harnessing the Django REST Framework for all of the server-side operations, while also integrating the Angular platform for the client-side interface. Together the combination will provide a seamless user experience in both functionality and user experience.

### II. Django Apps

#### A. API

##### 1. Models:

###### a) Person

- (1) **person\_id**: serial (primary key)
- (2) **first\_name**: varchar(50)
- (3) **last\_name**: varchar(50)
- (4) **date\_of\_birth**: datetime
- (5) **date\_of\_death**: datetime
- (6) **date\_of\_burial**: datetime
- (7) **obituary**: ntext
- (8) **portrait\_image**: ImageField
- (9) **landscape\_image**: ImageField

###### b) Plot

- (1) **plot\_id**: serial (primary key)
- (2) **plot\_location\_x**: decimal
- (3) **plot\_location\_y**: decimal
- (4) **plot\_state**: int
- (5) **person\_id**: int (foreign key: Person.person\_id)

###### c) Plot\_Status

- (1) **status\_id**: serial (primary key)
- (2) **status\_name**: varchar(50)
- (3) **color\_hex**: varchar(6)
- (4) **description**: varchar(255)

2. Views:

- a) CRUD for Plots
- b) CRUD for People
- c) CRUD for Plot Status
- d) index view

3. Templates:

- a) base\_generic.html
- b) index.html
- c) sexton\_index.html
- d) info\_popup.html
- e) sexton\_login.html
- f) confirm\_delete.html
- g) side\_bar.html