Donation Blog

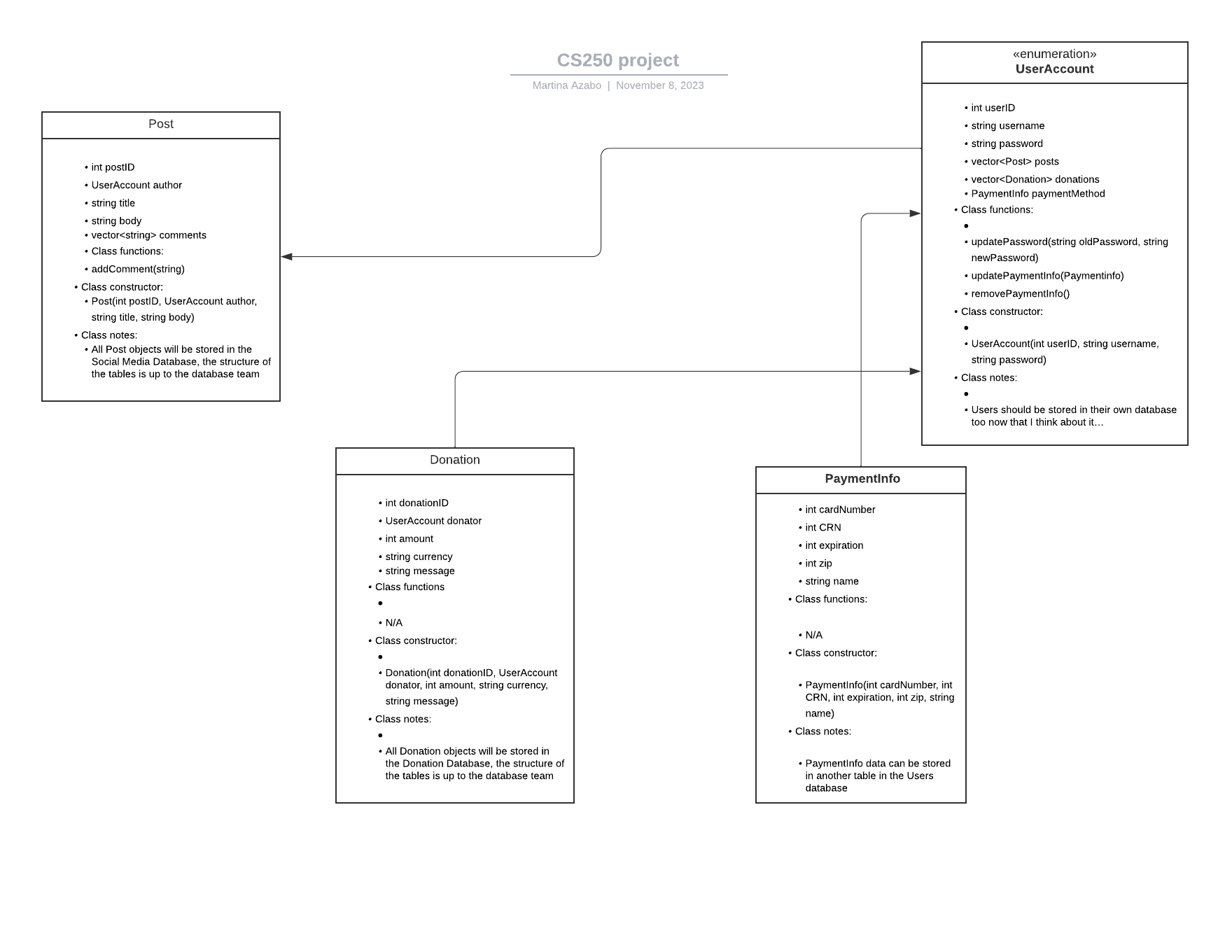
Software Design Specifications

Martina Azabo, Htet Hnin Su Wai, Nicholas Dibello-Hitta, Juan Diaz

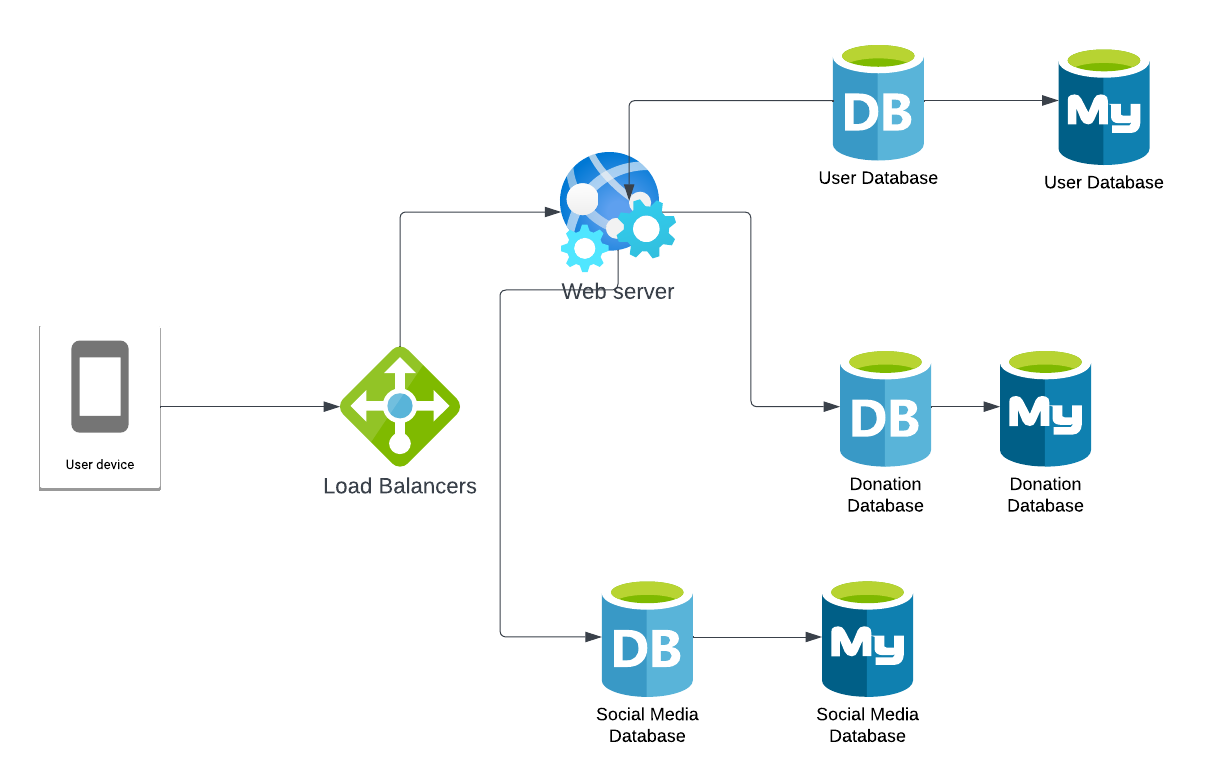
**System Description**

* The system we’re developing is a donation website. The main feature of this website is allowing users to browse through donation prompts posted by admin users and navigating to a payment tab to donate amount they desire. Users will land on a log in page on the landing page where they’ll be able to create an account where they’re able to keep tab of their donations and search for topics or locations that are relevant to them. Users will also be able to share these donation prompts via sharing a link or QR Code or by sharing it on social media platforms. Users will also be able to log out, delete accounts and customize their search results to find the donation best suitable for them. Admin users will be able to log in and out, post prompts, keep track of donations received, delete, or edit prompts and post updates regarding the donation project and how it’s progressing.

**Software Architecture Overview**



**Software Architecture Descriptions**



*Class Diagram*

* + Class name: Post
    - Class variables:
      * int postID
      * UserAccount author
      * string title
      * string body
      * vector<string> comments
    - Class functions:
      * addComment(string)
    - Class constructor:
      * Post(int postID, UserAccount author, string title, string body)
    - Class notes:
      * All Post objects will be stored in the Social Media Database the structure of the tables is up to the database team
  + Class name: Donation
    - Class variables:
      * int donationID
      * UserAccount donator
      * int amount
      * string currency
      * string message
    - Class functions
      * N/A
    - Class constructor:
      * Donation(int donationID, UserAccount donator, int amount, string currency, string message)
    - Class notes:
      * All Donation objects will be stored in the Donation Database
  + Class name: UserAccount
    - Class variables:
      * int userID
      * string username
      * string password
      * vector<Post> posts
      * vector<Donation> donations
      * PaymentInfo paymentMethod
    - Class functions:
      * updatePassword(string oldPassword, string newPassword)
      * updatePaymentInfo(Paymentinfo)
      * removePaymentInfo()
    - Class constructor:
      * UserAccount(int userID, string username, string password)
    - Class notes:
      * All User objects will be stored in the Users Database
  + Class name: PaymentInfo
    - Class variables:
      * int cardNumber
      * int CRN
      * int expiration
      * int zip
      * string name
    - Class functions:
      * N/A
    - Class constructor:
      * PaymentInfo(int cardNumber, int CRN, int expiration, int zip, string name)
    - Class notes:
      * PaymentInfo data can be stored in another table in the Users database

*Architecture Diagram*

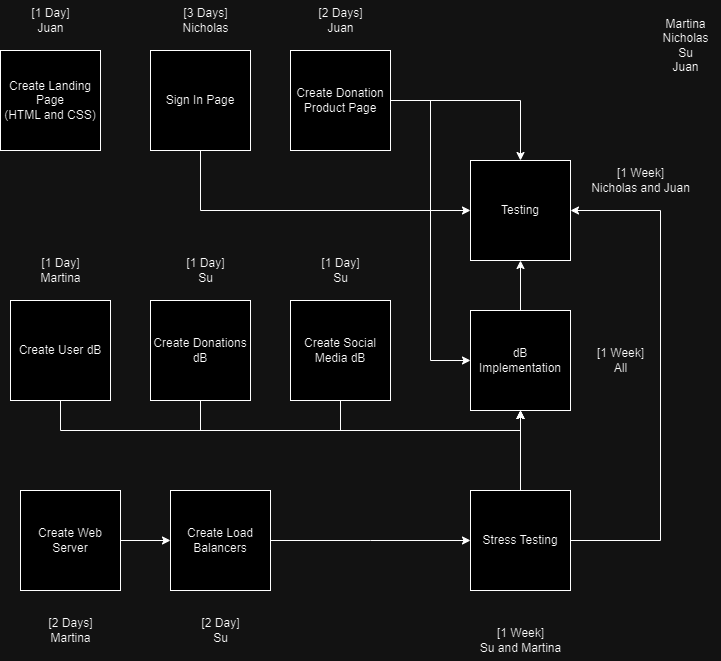
* User Device: any device that allows either consumers or administrators to access the website
* Load Balancers: protects servers from being overloaded and DDoS attacks.
* Web Server: main web hosting server for the page.
* Social Media Database: hosts information on the sites’ API data and usage of other sites
* Donation Database: contains information on fundraisers currently present on site. Each entry is removed if the admin user who made it requests it to be deleted.

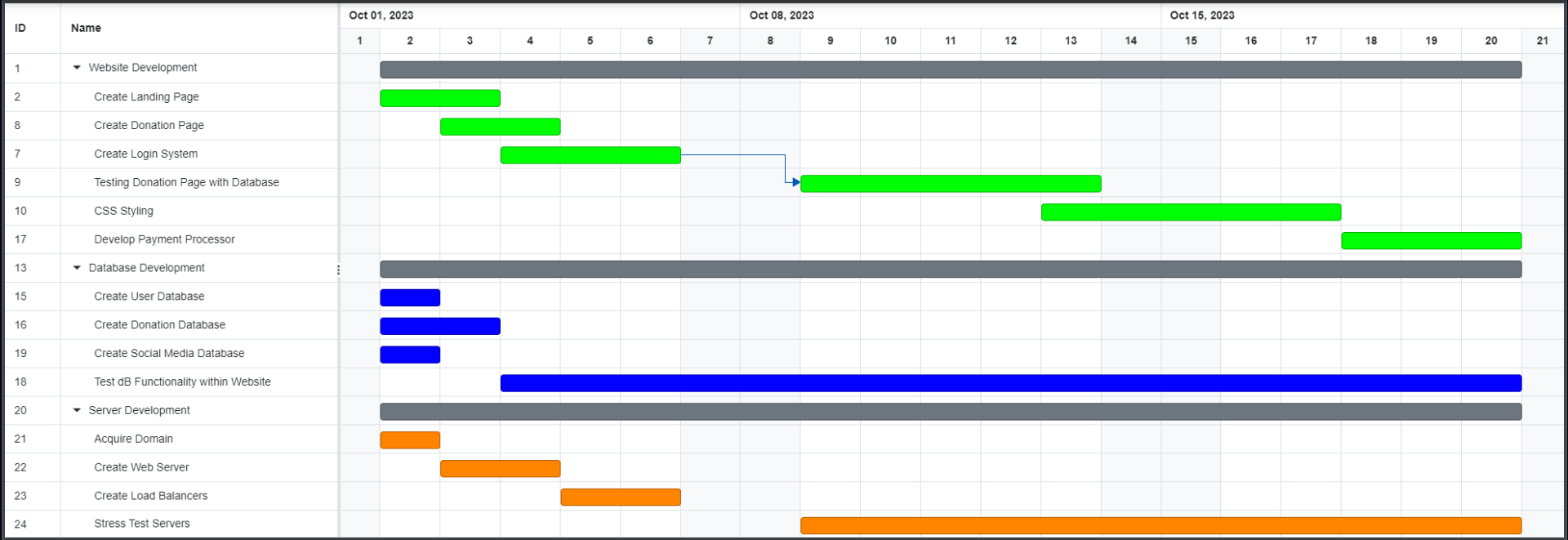
*Attributes*

* Responsive website that can be easily viewed on any device
* Built so that the system can be converted into a mobile phone app at a later date
* Fast response time when accessing different pages and updates to the database
* Uses HTML5, CSS, and mySQL
* Two types of users
  + Consumers: those who contribute to fundraisers and view publicly available information
  + Administrators: higher access users who create fundraisers and manage them using the tools provided by the website
    - Must be approved by managers of the website.
* Payment process is done by a trusted third party.
* User information, including but not limited to:
  + Passwords
  + Payment Information

will be encrypted to increase security

**Development Plan and timeline**

****

****

**Data Management Strategy**

We will be using SQL Data Management for this website. Since our data is structured and the relationships between each data such as User information, Payment information and sharing post are consistent, it is suitable for us to use SQL databases to manage our data. Information such as payment transactions are important to be stored securely in a database. Due to the simplicity of our website, I think creating one database with several tables that correspond to each class will help us manage it successfully.