Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2021

STAR
SPONSORSHIP PROGRAM, INC.

Star.js
Star Sponsorship Web-App

Saugat Karki
Bishesh Pote
Ayush Bhandari
John Paul Jones

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	12.05.2021	AB, JJ, SK, TP, BP	document creation
0.2	12.08.2021	AB, JJ, SK, TP, BP	complete draft
0.3	04.19.2022	AB, JJ, SK, BP	update document
1.0	04.20.2022	AB, JJ, SK, BP	official release

Contents

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The web application will host a variety of features to enhance user experience. The core of the application will revolve around allowing sponsors to use the donation system to donate to the STAR program. Another core function for the application will be the implementation of a system to allow the administrator to edit the contents of the web application. The application will host a platform to detail the success stories of students associated with the STAR program. User authentication will be implemented to allow the availability of features for the administrator. Newsletter/blog upload functionality will be another feature for administrators.

2 System Overview

The entire system can be broken down into Graphical User Interface(GUI), Back-end server, Database and Data controller. Components within the Back-end server interact with other three layers to store/retrieve/make changes to any form of data stored within the online database. The back-end server retrieves input data from the user which is then stored in the database. The back-end server fetches the updated data from the database and displays it to the user through GUI. The back-end server also fetches from the Data controller and based on the type of data retrieved, the data is either displayed(example: Google Maps API) or used to perform operations(example: user authentication/donation).

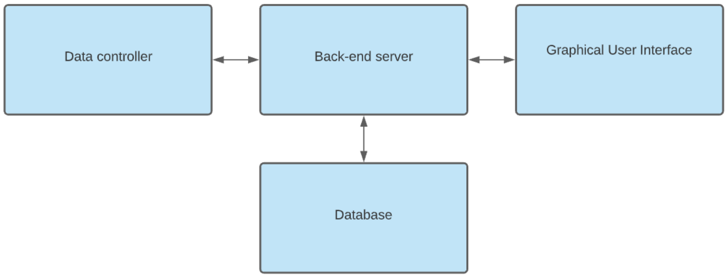

Figure 1: System architectural layer diagram

2.1 Graphical User Interface

The home page will include selected success stories of students associated with the organization. It will feature information about how the organization is changing the lives of those students. The website will let the users navigate through different pages like Blog, Success Stories, Newsletter, FAQs, etc. The Success Stories page will have an entire catalogue of success stories with description of each students. The web application will consist of an interface that navigates the users to the donation page. A login interface will be linked at the footer of the page for admin authentication. GUI features like updating success stories, newsletters, blogs, & managing donation will be available only for the administrator.

2.2 Database

Data generated from the following components will be stored in firebase realtime database and storage: Homepage, Success Stories, Newsletters/Blogs, Administrator Authentication, & About Star.

When logged in as an administrator, they will have permission to make changes to the data in the pages. This will update the database.

2.3 Data controller

The Data controller system manages the API's that will be used to authenticate the administrator, make donations, and integrate Google Maps services. The Data Controller communicates with the back-end server which initiates requests for retrieval of data in the database. After processing the request, the system sends the requested data to the Back-end server for processing.

2.4 Back-end server

The Back-end server will handle the communication between the database, data controller, and the GUI.

3 Subsystem Definitions & Data Flow

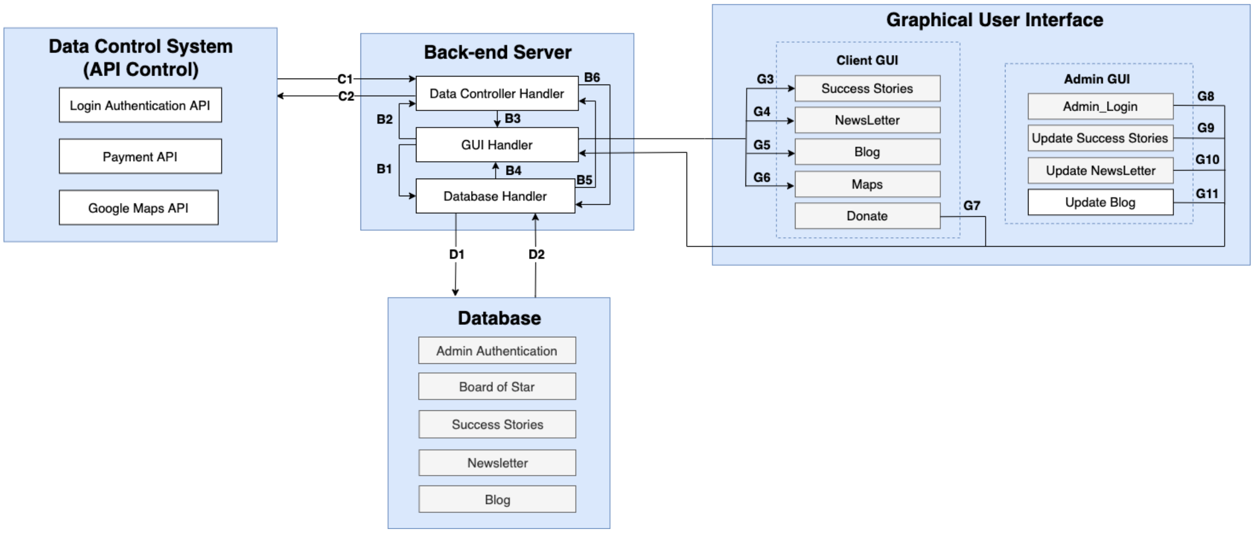

Figure 2: Overall data flow diagram

4 GUI Management System

This section describes the subsystems present in the GUI Layer of the web application. The GUI layer is divided into two parts: Client GUI and Admin GUI. The Client GUI includes subsystems that are applicable to a client, i.e. for any user visiting the web application. It consists of 5 subsystems: Success Stories, NewsLetter, Blog, Maps, and Donate. The Admin GUI includes 4 subsystems: Login, Update Success Stories, Update NewsLetter, Update Blog, and Manage Donation. For all the subsystems of GUI layer, the input comes from the back-end server (data that is to be displayed) and output goes to the back-end server (requests that are to be performed).

4.1 Success Stories

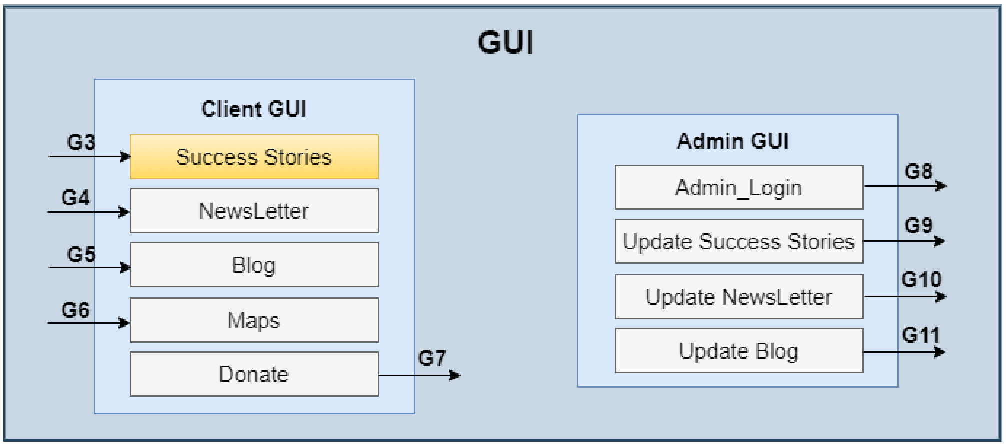

Figure 3: Success Stories subsystem description diagram

4.1.1 Responsibilities

This subsystem is responsible to display information, success stories, on the web-application that are received from the back-end server.

4.1.2 Subsystem Interfaces

Table 2: Success Stories Subsystem interfaces

ID	Description	Inputs	Outputs
G3	Display success stories on the web-application	Success stories information	N/A

4.2 NewsLetter

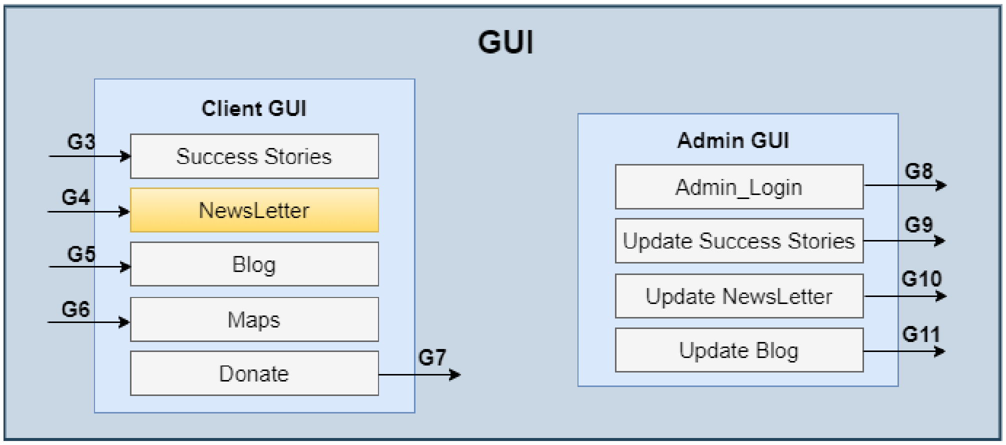

Figure 4: NewsLetter subsystem description diagram

4.2.1 Responsibilities

This subsystem is responsible to display newsletter information on the web-application that are received from the back-end server.

4.2.2 Subsystem Interfaces

Table 3: NewsLetter Subsystem interfaces

ID	Description	Inputs	Outputs
G4	Display newsletters on the web-application	Newsletter information	N/A

4.3 Blog

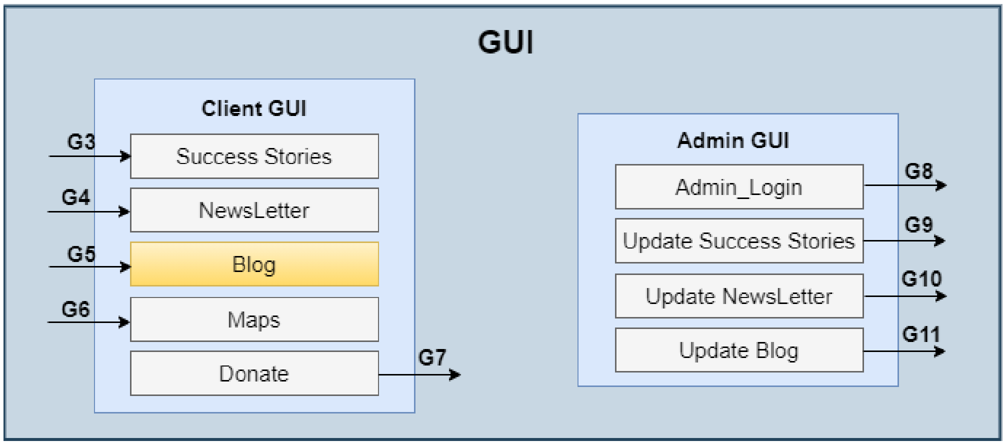

Figure 5: Blog subsystem description diagram

4.3.1 Responsibilities

This subsystem is responsible to display blogs on the web-application that are received from the back-end server.

4.3.2 Subsystem Interfaces

Table 4: Blog Subsystem interfaces

ID	Description	Inputs	Outputs
G5	Display blogs on the web-application	Blog information	N/A

4.4 Maps

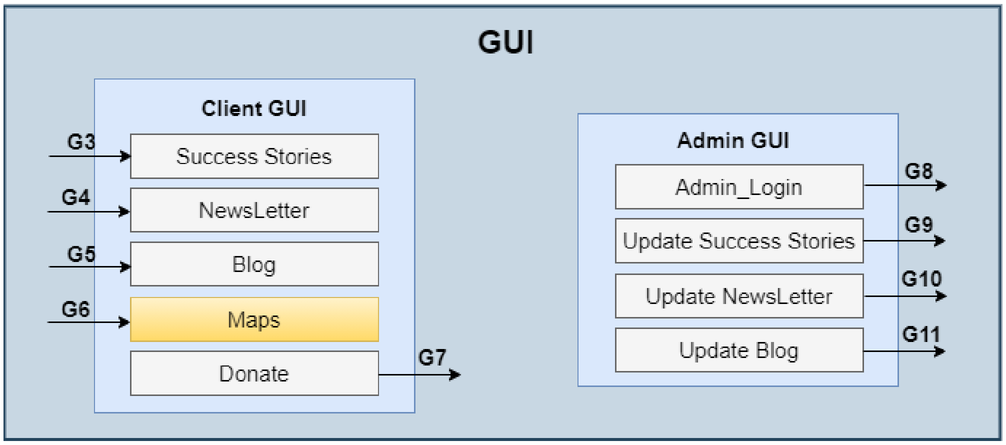

Figure 6: Maps subsystem description diagram

4.4.1 Responsibilities

This subsystem is responsible to display map on the web-application that is received from the back-end server.

4.4.2 Subsystem Interfaces

Table 5: Maps Subsystem interfaces

ID	Description	Inputs	Outputs
G6	Display map on the web-application	Google Map information	N/A

4.5 Donate

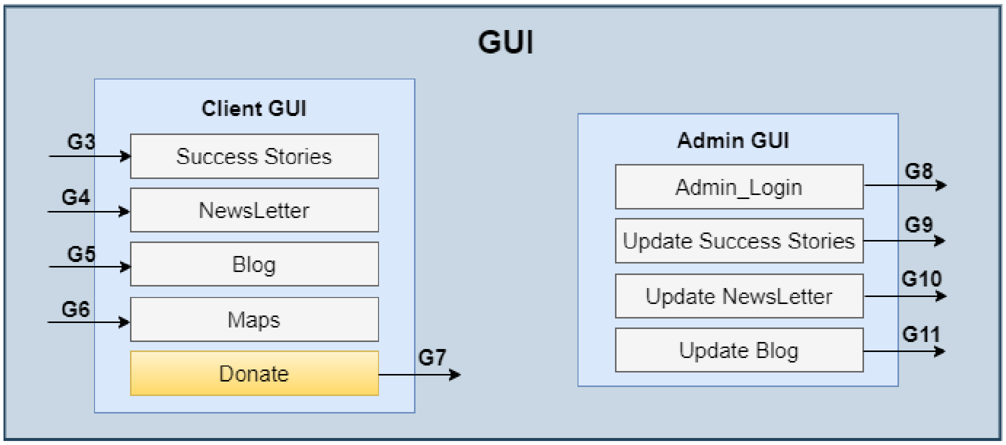

Figure 7: Donate subsystem description diagram

4.5.1 Assumptions

- All login inputs are type safe.

4.5.2 Responsibilities

Donate subsystem captures the user input, for donation, and sends it to the back-end server for encryption, verification, and completion of payment process.

4.5.3 Subsystem Interfaces

Table 6: Donate Subsystem interfaces

ID	Description	Inputs	Outputs
G7	Send donation information	User Input	Payment information

4.6 Admin Login

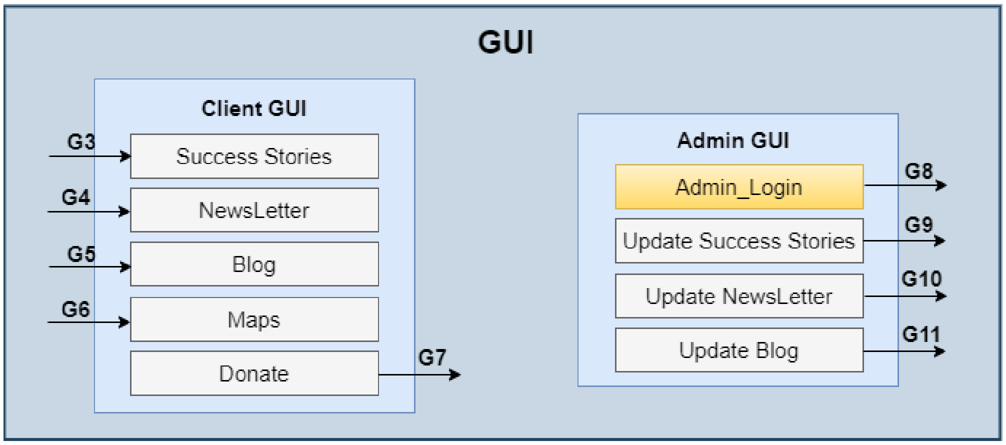

Figure 8: Admin Login subsystem description diagram

4.6.1 Assumptions

- All login inputs are type safe.

4.6.2 Responsibilities

Admin Login subsystem captures the user input, for admin login, and sends it to the back-end server for encryption, verification, and authentication.

4.6.3 Subsystem Interfaces

Table 7: Admin Login Subsystem interfaces

ID	Description	Inputs	Outputs
G8	Send admin login information	Email, Password	Email, Password

4.7 Update Success Stories

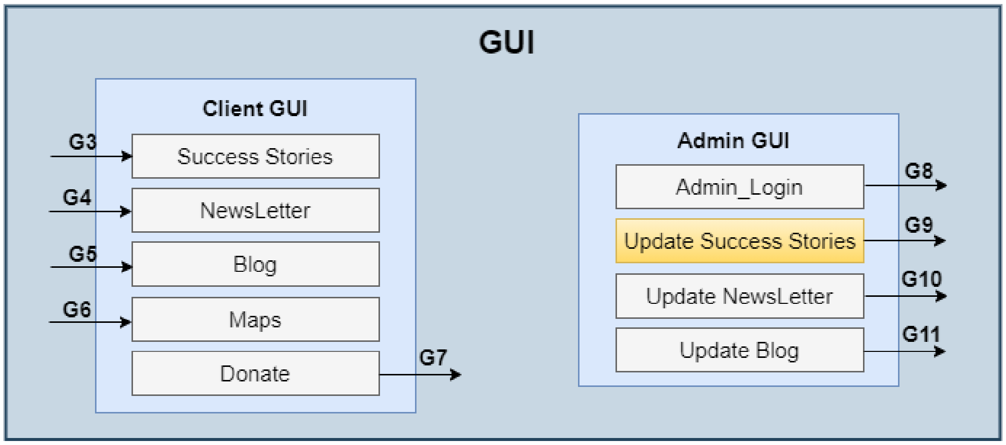

Figure 9: Update Success Stories subsystem description diagram

4.7.1 Responsibilities

This subsystem captures the changes selected/made, in success stories, by the admin. The change requests are sent to the back-end-server for validation and modification.

4.7.2 Subsystem Interfaces

Table 8: Update Success Subsystem interfaces

ID	Description	Inputs	Outputs
G9	Send success stories change request	User Input/Changes	Change request

4.8 Update NewsLetter

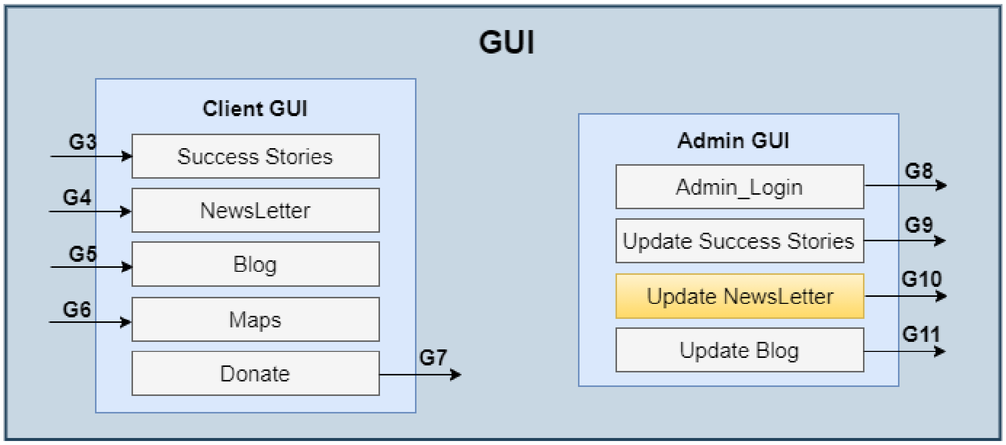

Figure 10: Update NewsLetter subsystem description diagram

4.8.1 Responsibilities

This subsystem captures the changes selected/made, in newsletters, by the admin. The change requests are sent to the back-end-server for validation and modification.

4.8.2 Subsystem Interfaces

Table 9: Update NewsLetter Subsystem interfaces

ID	Description	Inputs	Outputs
G10	Send newsletters change request	User Input/Changes	Change request

4.9 Update Blog

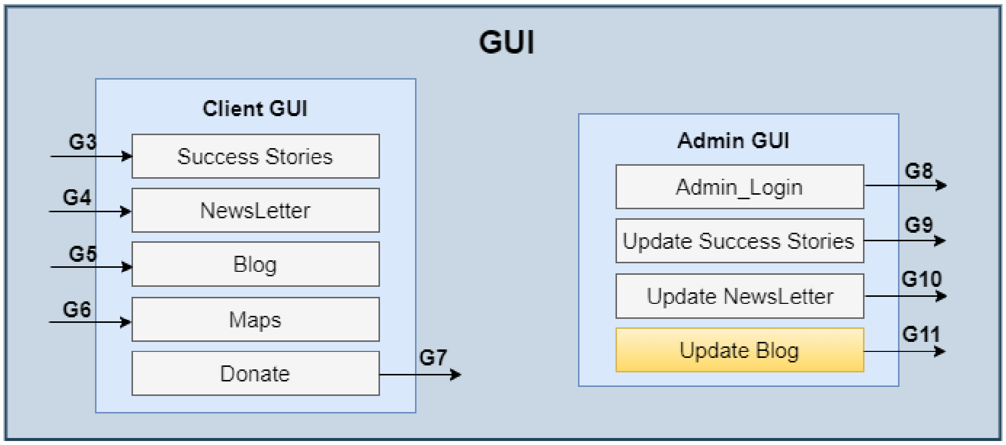

Figure 11: Update Blog subsystem description diagram

4.9.1 Responsibilities

This subsystem captures the changes selected/made, in blogs, by the admin. The change requests are sent to the back-end-server for validation and modification.

4.9.2 Subsystem Interfaces

Table 10: Update Blog Subsystem interfaces

ID	Description	Inputs	Outputs
G11	Send newsletters change request	User Input/Changes	Change request

5 Database Management System

This section describes the database layer in the architecture design of the web application. The database management system stores and manages all the data required for the web application to operate. It has 5 subsystems each handling different category of data. Each data in the database can be retrieved, updated, or deleted by Database Controller. The description of each sub system of database management system is as follows:

5.1 Admin Authentication

This entity in the database management system stores information to authenticate admin of the web application to manage various resources in the website.

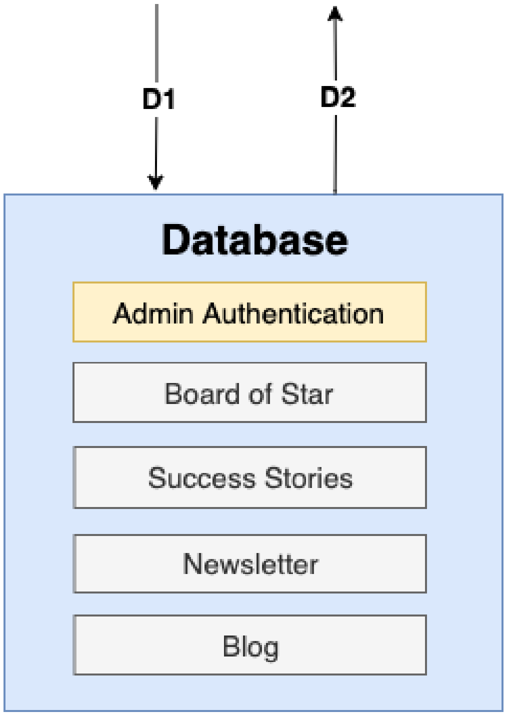

Figure 12: DBMS Admin Authentication subsystem description diagram

5.1.1 Responsibilities

This entity is responsible for storing and handling administrator account information.

5.1.2 Subsystem Interfaces

Table 11: DBMS Admin Authentication Subsystem interfaces

ID	Description	Inputs	Outputs
D1	Modify Credentials	Email, Password	N/A
D2	Retrieve Credentials	N/A	Email, Password

5.2 Board of Star

This entity in the database management system stores information about board members of the organization.

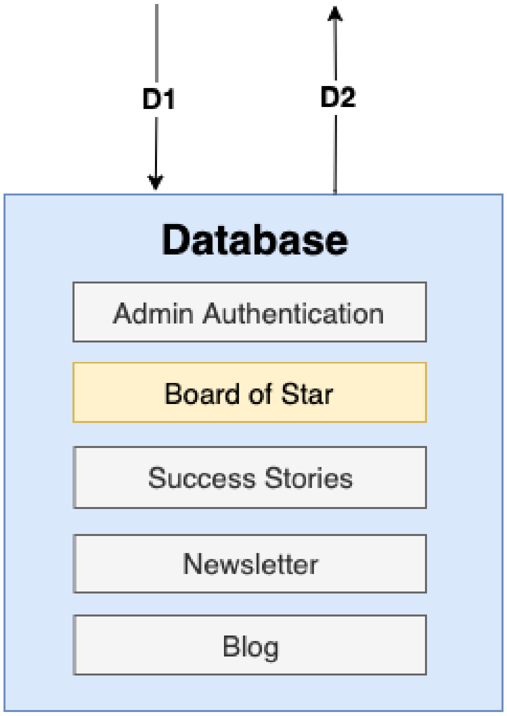

Figure 13: DBMS Board of Star subsystem description diagram

5.2.1 Responsibilities

This entity is responsible for storing and handling information about board members such as their name, position, and contact information.

5.2.2 Subsystem Interfaces

Table 12: DBMS Board of Star Subsystem interfaces

ID	Description	Inputs	Outputs
D1	Add/Delete a board member	Board member info	N/A
D2	Retrieve a board member	Board member id	Board member info

5.3 Success Stories

This entity in the database management system stores information about success stories of selected recipients.

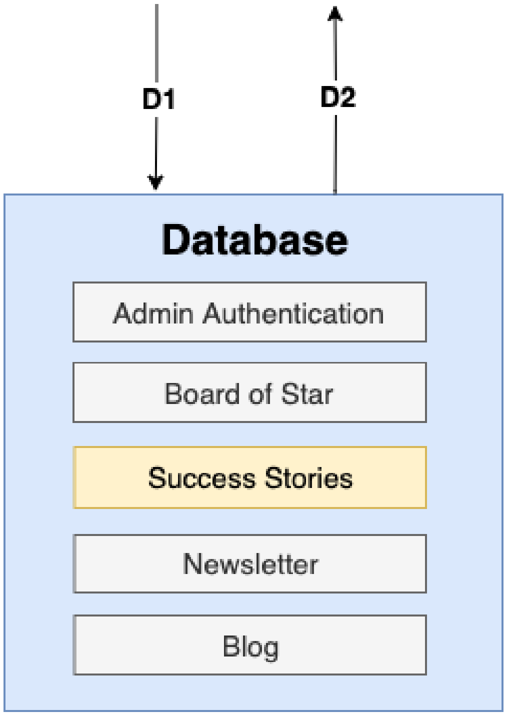

Figure 14: DBMS Success Stories description diagram

5.3.1 Responsibilities

This entity is responsible for storing and handling text, picture, and/or video relating to success stories of recipients of the donation.

5.3.2 Subsystem Interfaces

Table 13: DBMS Success Stories Subsystem interfaces

ID	Description	Inputs	Outputs
D1	Create success story	Story info	N/A
D1	Update/Delete success story	Story id, Story info	N/A
D2	Retrieve success story	Story id	Story info

5.4 Newsletter

This entity in the database management system stores information newsletters to be presented in the web application.

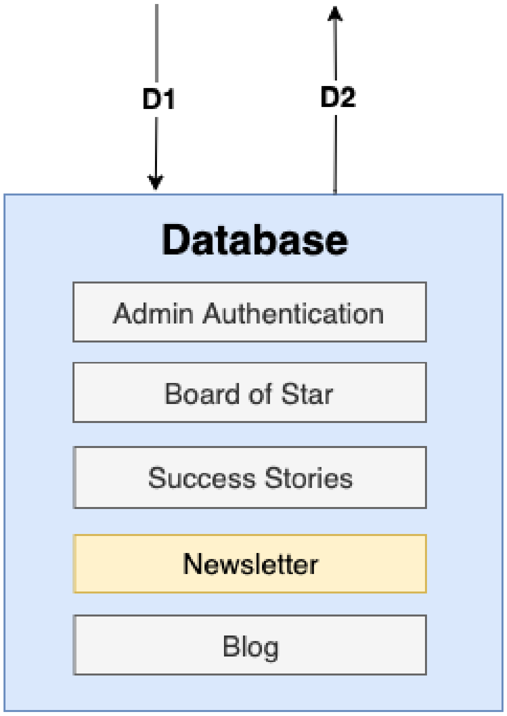

Figure 15: DBMS Newsletter subsystem description diagram

5.4.1 Responsibilities

This entity is responsible for storing and handling data associated with newsletters of the organization and their publication date.

5.4.2 Subsystem Interfaces

Table 14: DBMS Newsletter Subsystem interfaces

ID	Description	Inputs	Outputs
D1	Upload newsletter	Newsletter info	N/A
D1	Delete newsletter	Newsletter id	N/A
D2	Retrieve newsletter	Newsletter id	Newsletter info

5.5 Blog

This entity in the database management system stores blogs of the organization.

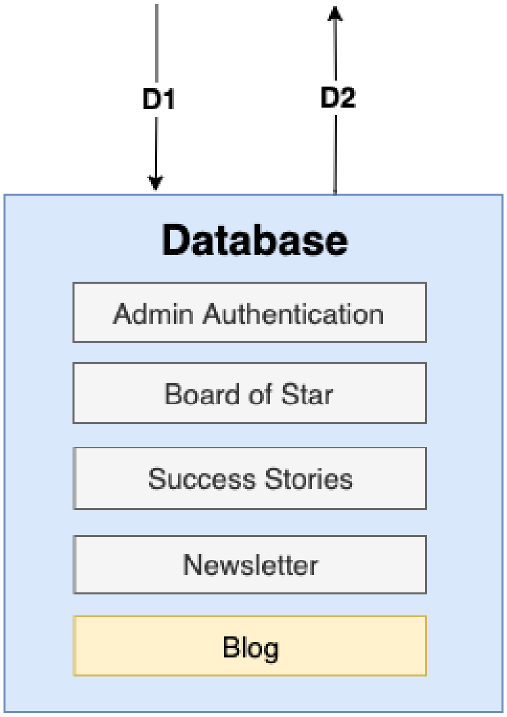

Figure 16: DBMS Blog subsystem description diagram

5.5.1 Responsibilities

This entity is responsible for storing and handling blogs and their publication date and time.

5.5.2 Subsystem Interfaces

Table 15: DBMS Blog Subsystem interfaces

ID	Description	Inputs	Outputs
D1	Upload blog	Blog info	N/A
D1	Delete blog	Blog id	N/A
D2	Retrieve blog	Blog id	Blog info

6 Data Controller System

This section describes the Data Controller System in the architecture design of the web application. The control system manages the API's that will be used to authenticate login sessions, make donations, and integrate Google Maps services. This system will directly communicate with the back-end server to initiate requests for retrieval of data and information located in the database.

6.1 Login Authentication API

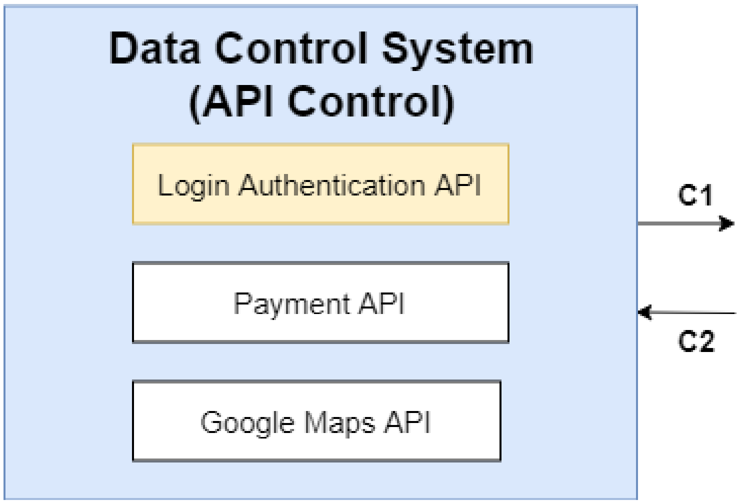

Figure 17: Login Authentication API subsystem description diagram

6.1.1 Responsibilities

Login API communicates with the back-end server to validate administrator credentials.

6.1.2 Subsystem Interfaces

Table 16: Data Control System - Login API Subsystem interfaces

ID	Description	Inputs	Outputs
C2	Receive authentication information	Email, Password	N/A
C1	Send authentication result	N/A	Pass/Fail

6.2 Payment API

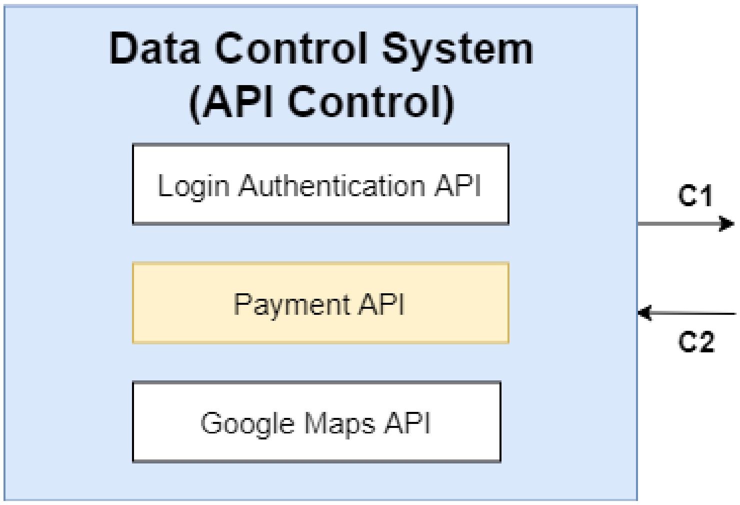

Figure 18: Payment API subsystem description diagram

6.2.1 Responsibilities

Payment API communicates with back-end server to process payments.

6.2.2 Subsystem Interfaces

Table 17: Data Control System - Payment API Subsystem interfaces

ID	Description	Inputs	Outputs
C2	Get donation information	Donation Information	N/A
C1	Validate Payment	N/A	Pass/Fail

6.3 Google Maps API

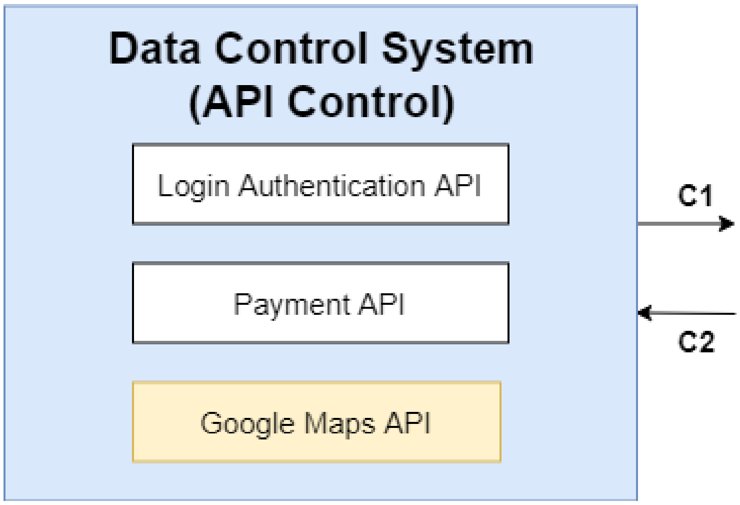

Figure 19: Google Maps API subsystem description diagram

6.3.1 Responsibilities

Google Maps API returns directions and map information of the Star Sponsorship Program to the back-end server.

6.3.2 Subsystem Interfaces

Table 18: Data Control System - Google Maps API Subsystem interfaces

ID	Description	Inputs	Outputs
D2	Google Maps API Request	Google Map address	N/A
D1	Send map results	N/A	Map information

7 Back-End Server

This section describes the Back-end Server System in the architecture design of the web application. The Back-end server will process, organize, and facilitate the retrieval and sending of data and information located in the database. It will accomplish this by having a handler for Data Controller, GUI, and Database.

7.1 Data Controller Handler

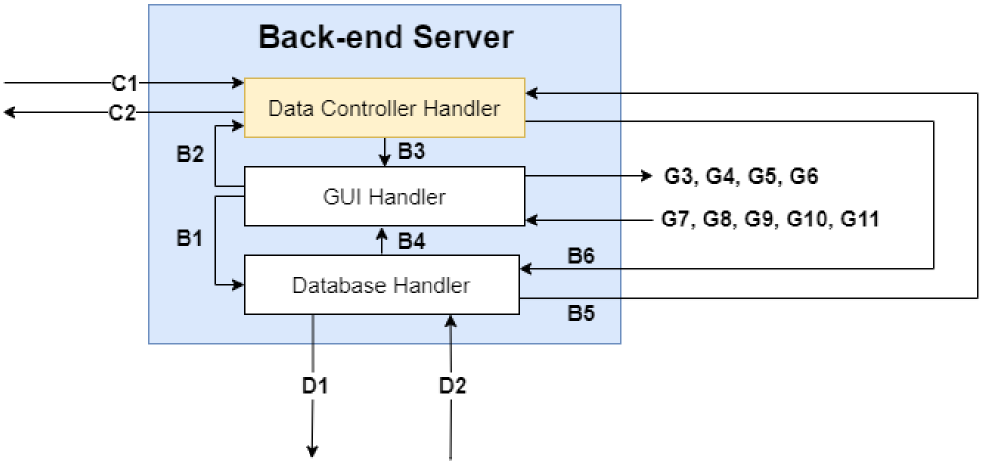

Figure 20: Data Controller Handler subsystem description diagram

7.1.1 Responsibilities

The Data Controller Handler will communicate with the Data Control System layer to process API requests. If it is the Login Authentication API, then it will send back a validation for the login information. If it is the Payment API, then it will begin the payment process based off the input from the API. If it is the google maps API, then it will send back information for directions and map information for display. It also communicates with the database handler and GUI handler.

7.1.2 Subsystem Interfaces

Table 19: Data Controller Handler interfaces

ID	Description	Inputs	Outputs
B2	Retrieve GUI requests	GUI Request	N/A
B3	Send GUI requests results	N/A	GUI Request Results
B6	Send database requests	N/A	DB requests
B5	Retrieve database results	DB results	N/A
C2	Retrieve API feedback	API results	N/A
C1	Send API requests	N/A	API requests

7.2 GUI Handler

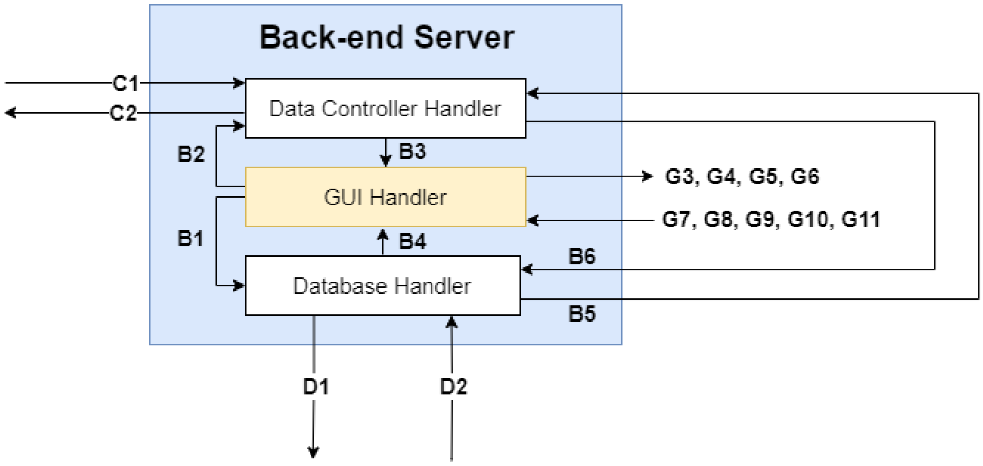

Figure 21: GUI Handler subsystem description diagram

7.2.1 Responsibilities

The GUI Handler will communicate with the GUI in order to relay data and information that is to be put on display on the GUI. The GUI Handler will request data and information from the Data Controller and Database to send to the GUI to display.

7.2.2 Subsystem Interfaces

Table 20: GUI Handler interfaces

ID	Description	Inputs	Outputs
B1	Send requests to Database Handler	N/A	GUI Handler requests
B4	Retrieve query results from handler	Query results	N/A
B2	Send requests to Data Controller Handler	N/A	Data Controller Handler requests
B3	Retrieve API results from handler	API results	N/A
G7, G8, G9, G10, G11	Retrieve requests from GUI	GUI requests	N/A
G3, G4, G5, G6	Send results from Database and API handler	N/A	API and database results

7.3 Database Handler

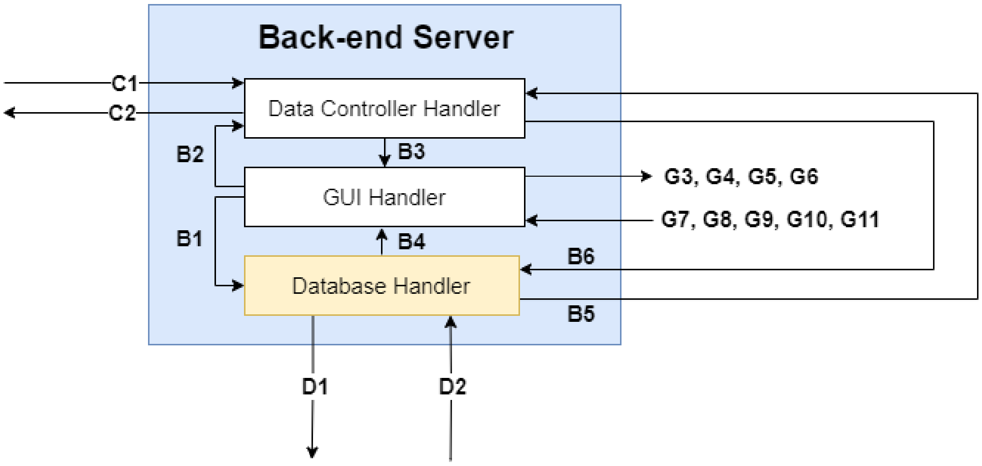

Figure 22: Database Handler subsystem description diagram

7.3.1 Responsibilities

Whenever any request is made that requires data or information stored in the database, the Back-end server will get a request from the Data Controller or GUI and communicate with the Database Handler in order to retrieve the necessary data. The Back-end server will check with the Database and see if the information will be used for validation, given to the Data Controller, or to be displayed on the GUI.

7.3.2 Subsystem Interfaces

Table 21: Database Handler interfaces

ID	Description	Inputs	Outputs
B1	Retrieve GUI Handler requests	GUI Handler requests	N/A
B4	Send GUI Handler results	N/A	Query results
B6	Retrieve Data Controller Handler requests	Data Controller Input Requests	N/A
B5	Send Data Controller Handler results	N/A	Login credentials
D2	Retrieve Database results	Query results	N/A
D1	Query Execution	N/A	Query Requests