Cal Poly Formula SAE Progress Management System

https://www.calpolyracing.org/

Preston Calderon, Kira Hudson, Tran Nguyen, Victor Lopatyuk

BUS 394 - Systems Analysis and Design

Professor Jim Burleson

December 7, 2023

Background Information

Cal Poly Racing is a student-run organization at California Polytechnic State University dedicated to the design, construction, and racing of cars. Bringing together students from various engineering disciplines, the organization focuses on creating vehicles for competitions like the Formula SAE (Society of Automotive Engineers) event, providing a platform for students to apply theoretical knowledge to practical engineering challenges. These competitions serve as arenas for honing teamwork and problem-solving skills.

In addition to its emphasis on technical skills, Cal Poly Racing places a strong emphasis on professional development. Students are afforded opportunities to refine their leadership, communication, and project management skills through hands-on experiences and collaboration with industry professionals.

With a membership exceeding 100 students, Cal Poly Racing operates through three distinct subteams: Formula SAE, Baja SAE, and Business Team, each responsible for different aspects of car development. The Formula SAE team focuses on designing and constructing formula-style race cars, while the Baja SAE team works on off-road race cars. The Business Team handles marketing and partner relations for the organization.

As a member of SAE International, a global association supporting engineering education and motorsports, Cal Poly Racing boasts numerous accolades, including several Formula SAE championships.

Despite its successes, the organization faces a challenge in the lack of a centralized system for progress tracking and management among team members. Currently, the use of information technology is individualized by team and, in some cases, specific team members. Given the complexity of ongoing engineering projects, varying software solutions are employed at different organizational levels, resulting in specific data and software for each team. Data is predominantly stored on local machines owned by team members, with limited data sharing facilitated through cloud solutions like Google Sheets. The absence of a unified approach hampers progress tracking, data sharing, and overall project cohesion, necessitating physical storage devices for data transfer between teams, subteams, and individual members.

Problem Identification

Cal Poly Racing faces significant challenges in the realm of information technology, primarily stemming from the absence of a centralized system for progress tracking and management. The organization, with over 100 members, operates through three sub teams, each utilizing individualized software solutions at different organizational levels. This decentralized approach results in specific data and software for each team, hindering overall project cohesion. Additionally, the reliance on local machines for data storage, coupled with limited data sharing through cloud solutions, such as Google Sheets, exacerbates the issue. The lack of a standardized system impedes efficient progress tracking and data transfer between teams and individual members, necessitating the use of physical storage devices like flash drives or portable hard disks. Consequently, Cal Poly Racing grapples with fragmented information management, posing challenges to effective communication, collaboration, and project oversight in the dynamic landscape of their ongoing engineering endeavors.

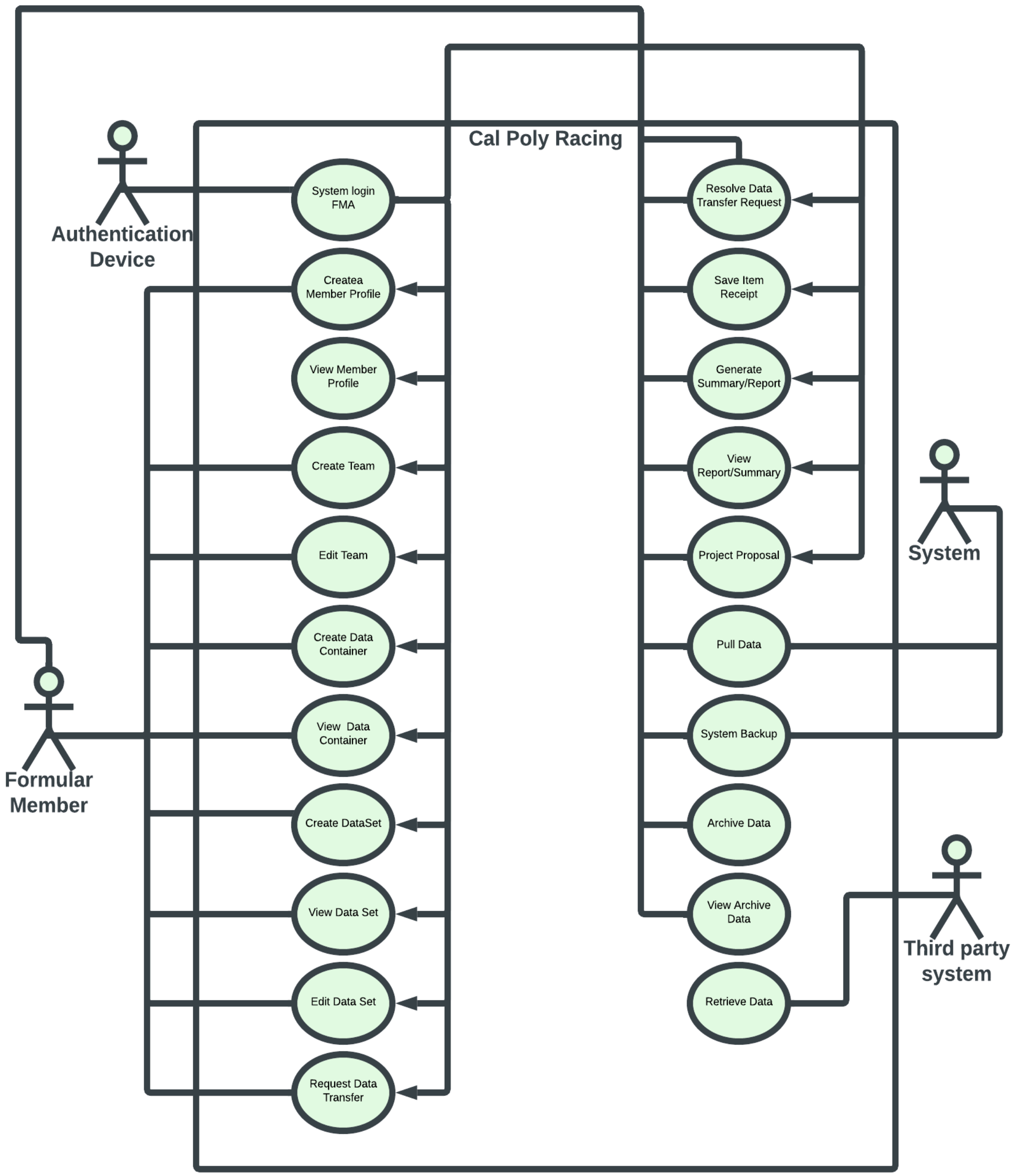
The proposed system for Cal Poly Racing aims to address the organization's information technology challenges by providing a centralized platform specifically designed for the Cal Poly Formula SAE President and associates. The system's primary purpose is to facilitate accurate documentation of work progress on the Formula SAE combustion car. It will enable the President and associates to document completed tasks, monitor ongoing work, communicate seamlessly with team members, and achieve overall data cohesion across the entire team. With a focus on managing the intricacies of 42 teams within the organization, the system ensures cohesive access to data based on individual credentials, allowing team members to access relevant information. Furthermore, the system incorporates data reservation mechanisms, ensuring that specific individuals can securely access data pertinent to their team, subteam, and other relevant criteria, thereby aligning with information security requirements. This comprehensive system seeks to streamline communication, enhance collaboration, and ensure efficient progress tracking for the complex and dynamic projects undertaken by Cal Poly Racing.

System Request

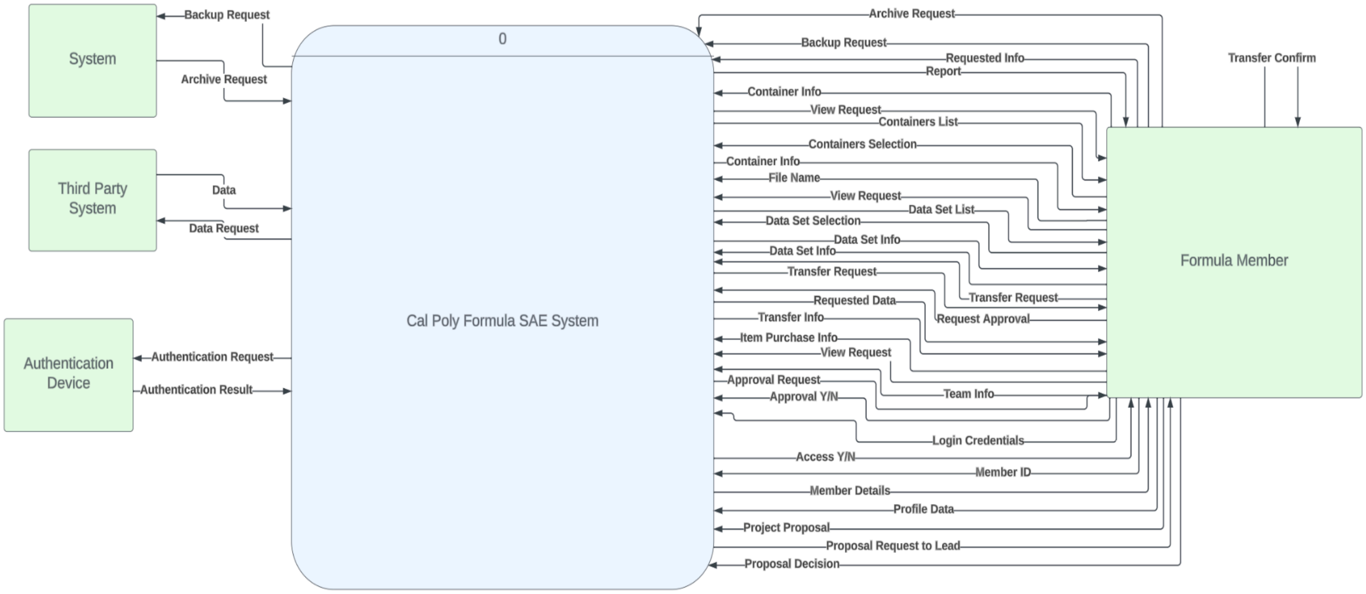
Project Name: Cal Poly Formula SAE Progress Management System *calpolyracing.org*	
Project Sponsor	SAE Club President: Eli Schulz
Project Overview	The purpose of this system request is to help the Cal Poly Formula SAE President and associates to accurately document work progress made on the Formula SAE combustion car. The system will help the President and their associates in documenting work completed, viewing work in progress, communication with team members, and overall data cohesion between all team members. The system will help manage the above with respect to 42 teams in total, allowing cohesive access to data as allowed by individuals credentials within the team. The system will also serve to reserve data to specific individuals based on their team, subteam, etc. in order to comply with information security requirements.
Business Requirements	The system will benefit the club by: ● Allowing individuals access to cohesive work progress data as allowed depending on the individuals credentials ● Creating a more efficient way to communicate information between teams ● Increased data security
Business Value	Reasonable expectations upon implementation ● 12% improvement in information accessing speed ● 10% improvement in communication efficiency ● 15% improvement in car production completion
Special Issues/Constraints	● Current system needs to keep data separated, each team can only have access to their specific team's data with their credentials ● Club members have concerns about switching over to different data management systems and learning how to utilize new systems ● Multifactor authorization system must be in place to ensure data security ● Cross validation system for approval of data transfer

Analysis

Use Case Diagram:



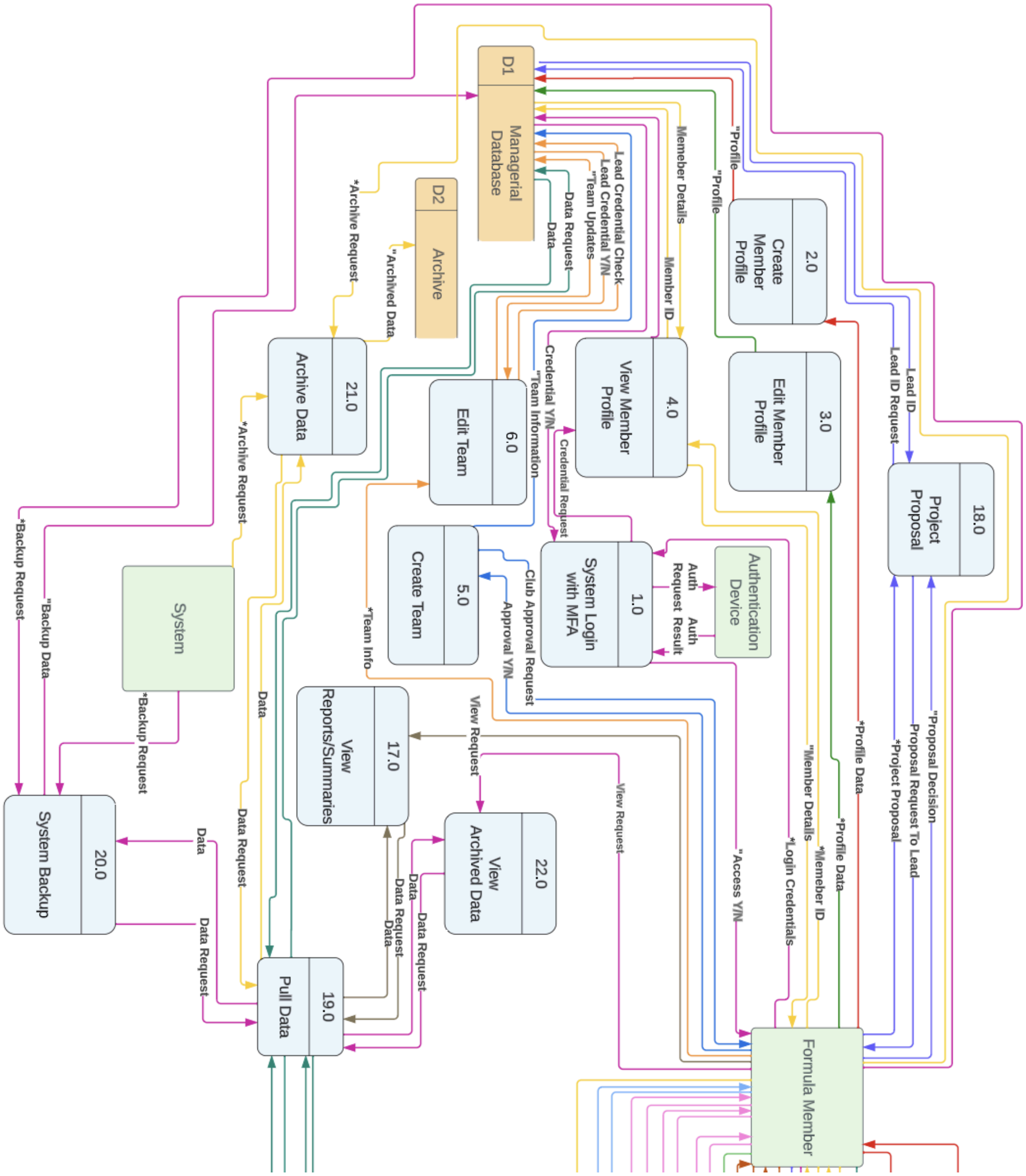
Context DFD:

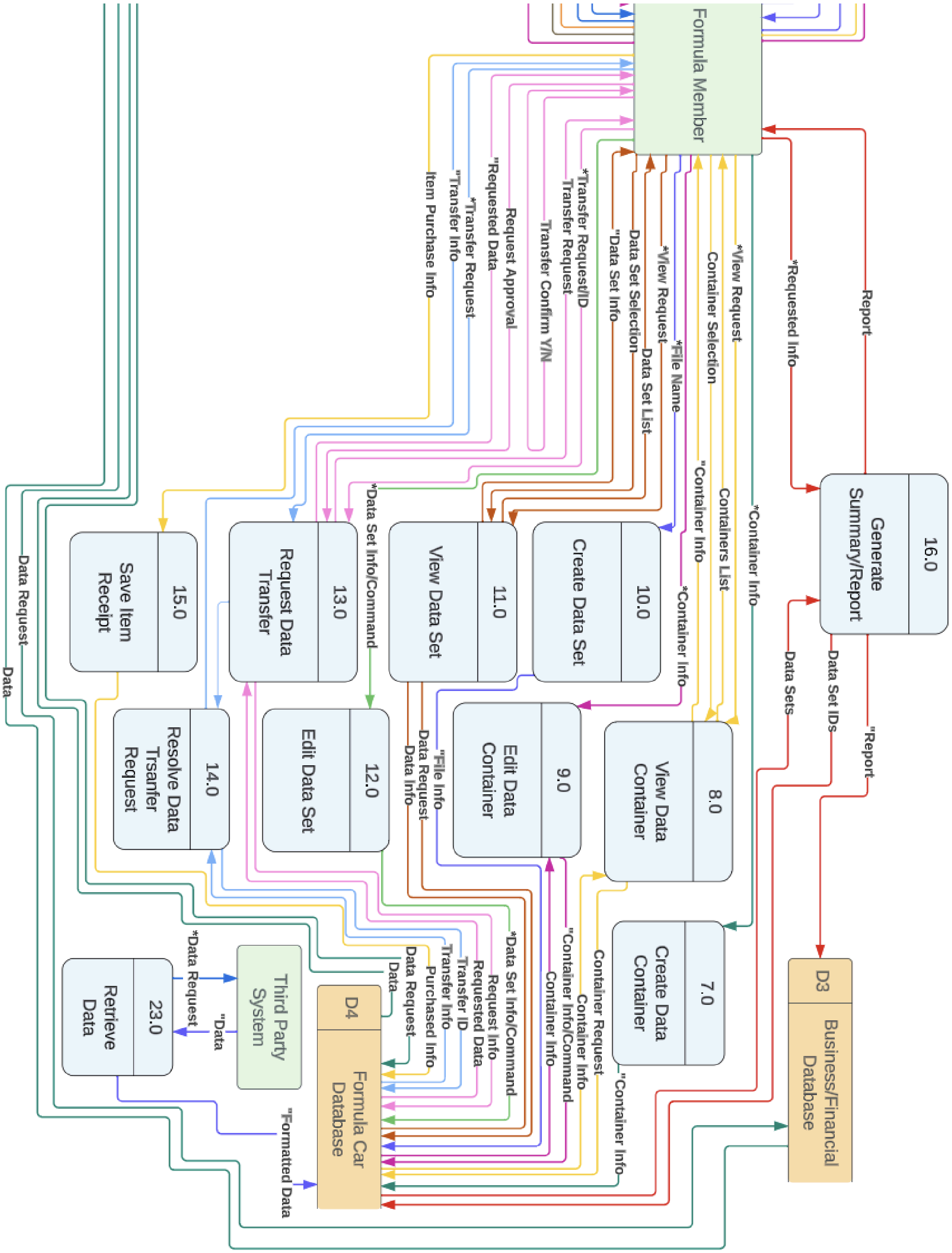

System
Backup Request
Archive Request
Third Party System
Data
Data Request
Authentication Device
Authentication Request
Authentication Result
0
Cal Poly Formula SAE System
Archive Request
Backup Request
Requested Info
Report
Container Info
View Request
Containers List
Containers Selection
Container Info
File Name
View Request
Data Set List
Data Set Selection
Data Set Info
Data Set Info
Transfer Request
Transfer Request
Requested Data
Request Approval
Transfer Info
Item Purchase Info
View Request
Approval Request
Team Info
Approval Y/N
Login Credentials
Access Y/N
Member ID
Member Details
Profile Data
Project Proposal
Proposal Request to Lead
Proposal Decision
Transfer Confirm
Formula Member

Level 0 DFD:

The symbol * denotes the first step in a use case to help understand the diagram

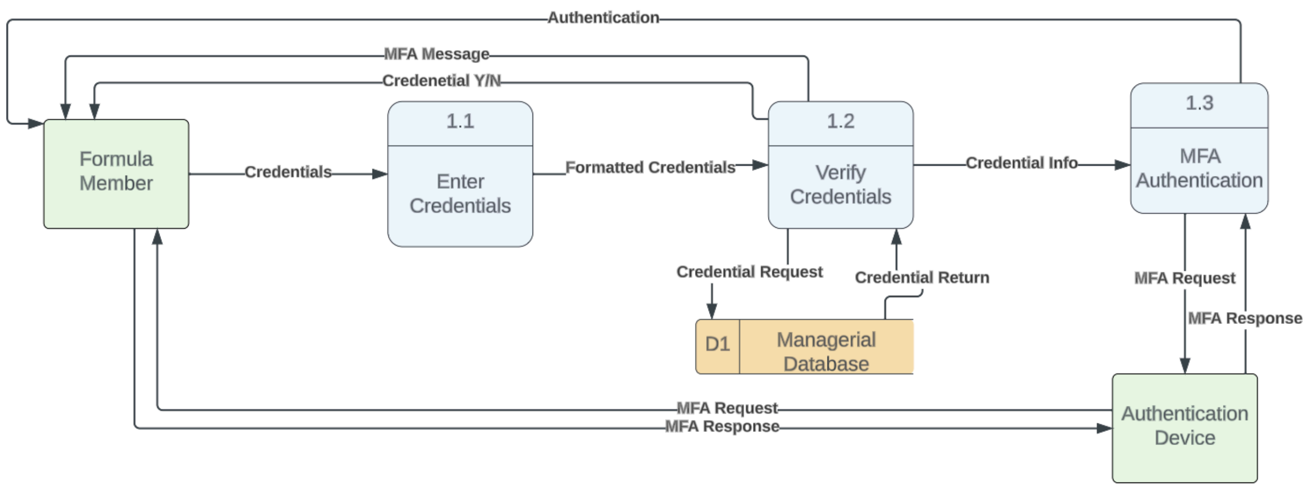
The symbol " denotes the last step in a use case to help understand the diagram



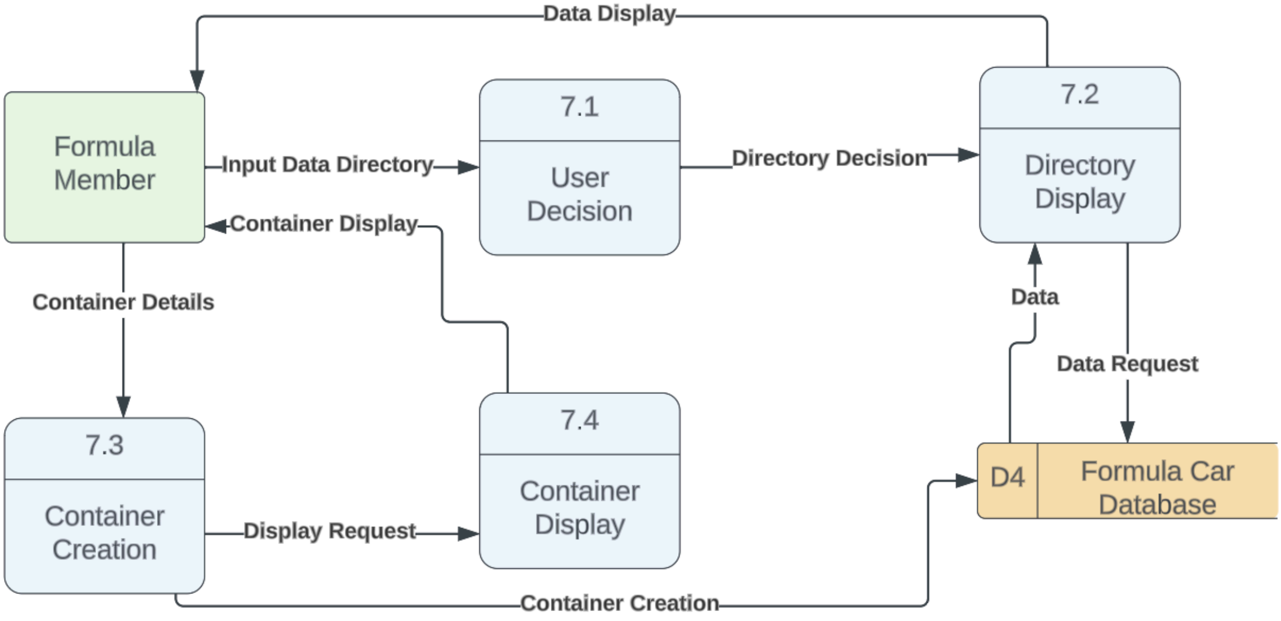

Formula Member
16.0
Generate Summary/Report
D3
Business/Financial Database
10.0
Create Data Set
11.0
View Data Set
13.0
Request Data Transfer
15.0
Save Item Receipt
8.0
View Data Container
9.0
Edit Data Container
12.0
Edit Data Set
14.0
Resolve Data Trsanfer Request
7.0
Create Data Container
D4
Formula Car Database
Third Party System
23.0
Retrieve Data
Report
*Requested Info
*View Request
Container Selection
*View Request
*File Name
Data Set List
Data Set Selection
*Data Set Info
*Transfer Request/ID
Transfer Request
Transfer Confirm Y/N
Request Approval
*Requested Data
*Transfer Request
*Transfer Info
Item Purchase Info
*Container Info
Containers List
*Container Info
*Container Info
*Data Set Info/Command
Data Sets
Data Set IDs
*Report
*File Info
Data Request
Data Info
*Data Set Info/Command
Request Info
Requested Data
Transfer ID
Transfer Info
Purchased Info
Data Request
Data
*Container Info/Command
Container Info
Container Request
Container Info
*Container Info
Data Request
Data
*Data Request
*Data
*Formatted Data

Level 1 DFDs:

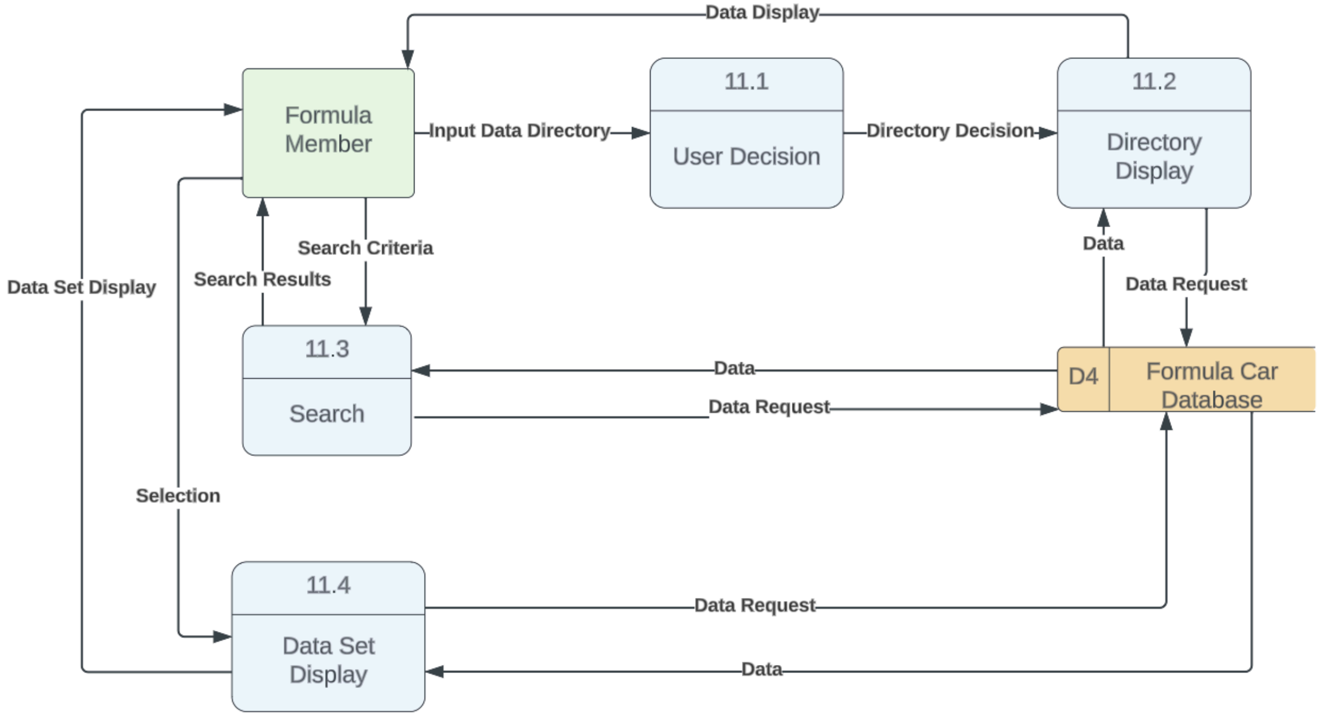
Login With MFA



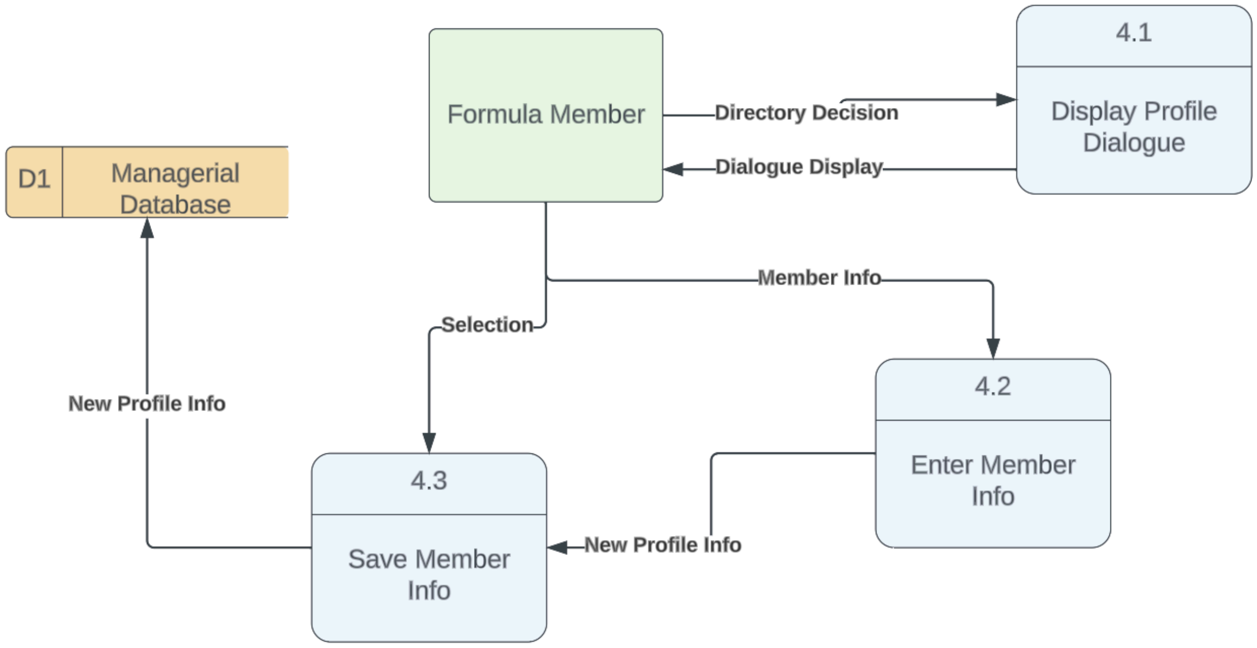
Create Data Container



View Data Set



Create Member Profile



Use Case Narratives:

Title:	Login with MFA
Description:	This use case describes the process of logging into the system with Multi Factor Authentication.
Primary Actor:	Formula Member
Trigger:	User would like to log in to the system.
Preconditions:	n/a
Normal Course:	1. User selects “Login” option 2. System prompts username and password entry 3. User enters username and password, then selects “Log In” 4. System checks the validity of the user’s username and password. 5. System sends an authentication request to the Authentication Device 6. User utilizes Authentication Device to accept or deny the request 7. Authentication Device conveys the authentication result to the system 8. System grants access to user
Postconditions:	1. User is logged into the system
Alternative Courses:	

Exceptions:	The user does not have a valid username or password (Occurs at step 3) 1. System displays error message: “Invalid Credentials”. Continue at step 2. The user is not granted access by the MFA to login (Occurs at step 8) 1. System displays error message: “MFA Access Denied” Continue at step 2.
Priority:	High

Title:	Create Member Profile
Description:	This use case describes the process of creating a new Member profile.
Primary Actor:	Formula Member
Trigger:	A user wishes to create a new profile for a Member to the system.
Preconditions:	1. User is logged into the system.
Normal Course:	1. User selects the “Member” option on the menu 2. System displays the member landing page 3. User selects the “Create” option 4. System prompt user for member information with the Create Profile dialogue 5. User enters team Member information ● Member name ● Major ● Phone Number ● Cal Poly Email ● Member ID ● Username ● Password 6. User selects the “Save” option 7. System validates information 8. System adds Member to database
Postconditions:	1. Member Profile is added to the Managerial Database
Alternative Courses:	

Exceptions:	If user’s email is not a Cal Poly email (occurs at step 3) 1. System displays error message - “Please enter a valid Cal Poly email address” Continue at step 2.
Priority:	High

Title:	Edit Member Profile
Description:	This use case describes the process of updating an existing Member’s profile.
Primary Actor:	Formula Member
Trigger:	The user wishes to make updates to a current Member’s profile.
Preconditions:	1. User must be logged in to the system. 2. There is at least one member profile existing within the database.
Normal Course:	1. User selects the “Member” option on the menu 2. System displays the member landing page 3. User selects the “Edit Member Profile” option. 4. System displays a list of Members, with an option to filter by: ● Name ● Major ● Contact information 5. User selects a Member to edit. 6. System displays the current Member information 7. User selects “Edit” 8. User makes updates the Member’s information. 9. User selects the “Save” option. 10. System saves updated Member information to the database.
Postconditions:	1. Member’s information is updated to the Managerial Database
Alternative Courses:	

Exceptions:	
Priority:	Medium

Title:	View Member Profile
Description:	This use case describes the process of viewing an existing member profile.
Primary Actor:	Formula Member
Trigger:	User wishes to view a member’s profile.
Preconditions:	1. User must be logged into the system 2. At least one member profile must exist within the system
Normal Course:	1. User selects the “Member” option on the menu 2. System displays the member landing page 3. User selects the “View” option 4. System displays a dialogue with search criteria a. Member Name b. Member ID 5. User enters desired search criteria and selects “Search” 6. System displays all search results corresponding to the search criteria 7. User selects a member profile 8. System displays member profile
Postconditions:	n/a
Alternative Courses:	
Exceptions:	User enters search criteria that does not match any existing profiles (Occurs at step 5) 1. System displays error message: “No Results Found”

	Continue at step 4
Priority:	Medium

Title:	Create Team
Description:	This use case describes the process of creating a team.
Primary Actor:	Formula member
Trigger:	User wishes to create a new team.
Preconditions:	1. User must be logged into the system 2. At least one member profile exists within the system
Normal Course:	1. User selects the “Team” option in the menu 2. System displays the Team landing page 3. User selects the “Create Team” option 4. System displays the team creation dialogue 5. User enters the team information a. Team name b. Member IDs 6. User selects “Create” option 7. System sends an approval request to user 8. User accepts approval request 9. System saves the new team into database
Postconditions:	1. Team information is saved in the Managerial Database
Alternative Courses:	
Exceptions:	The user rejects the team creation request (Occurs at step 8) 1. System displays an error message: “Team Creation Request Denied”

	Continue at step 4
Priority:	High

Title:	Edit Team
Description:	This use case describes the process of editing a team's information.
Primary Actor:	Formula Member
Trigger:	The user wants to make changes to an existing team.
Preconditions:	1. User is logged into the system 2. There is at least one team in the database 3. At least one member profile exists within the system
Normal Course:	1. User selects the "Team" option in the menu 2. System displays the Team landing page 3. User selects the "Edit Team" option 4. System displays a Credential Check dialogue 5. User enters Member ID and selects "Continue" 6. System displays a list of all existing teams in alphabetical order 7. User selects the team they want to edit 8. User selects "Edit" 9. User enters/edits team information 10. User selects the "Save" option. 11. System saves the updated team information into the database
Postconditions:	1. Team information is updated in the Managerial Database.
Alternative Courses:	

Exceptions:	The user’s credentials do not permit them to make edits (Occurs at step 6) 1. System displays an error message: “Invalid Credentials” Continue at step 2
Priority:	High

Title:	Create Data Container
Description:	This use case describes the process of creating a new data container.
Primary Actor:	Formula Member
Trigger:	User wishes to create a new data container.
Preconditions:	1. User must be logged into the system
Normal Course:	1. User selects the “Data” option in the menu 2. System displays the Data landing page 3. User selects the “Create New Data Container” option 4. System prompts user to create a title for the container 5. User enters a corresponding title and selects “Create” 6. System opens a new blank Data Container 7. System saves the data container into the database.
Postconditions:	1. The data container is saved into the Formula Car Database
Alternative Courses:	
Exceptions:	If data container title already exists (occurs at step 4) 1. System displays an error message: “Data container title already exists. Please enter a different title.” Continue at step 4. If data container title is not entered (occurs at step 4)

	2. System displays an error message: “Please enter container title.” Continue at step 4.
Priority:	High

Title:	View Data Container
Description:	This use case describes the process of viewing a data container.
Primary Actor:	Formula Member
Trigger:	User wishes to view an existing data container
Preconditions:	1. User must be logged into the system 2. At least one data container must exist within the system
Normal Course:	1. User selects the “Data” option on the menu 2. System displays the data landing page 3. User selects the ”View Data Container” option 4. System displays a list of all data containers in alphabetical order with a search option for refined criteria 5. User selects a data container and selects “View” 6. System displays the selected data container
Postconditions:	
Alternative Courses:	
Exceptions:	
Priority:	High

Title:	Edit Data Container
Description:	This use case describes the process of editing an existing data container.
Primary Actor:	Formula Member
Trigger:	User wishes to edit a data container.
Preconditions:	1. User must be logged into the system 2. At least one data container must exist within the system
Normal Course:	1. User selects the “Data” option in the menu 2. System displays the data landing page 3. User selects the “Edit Data Container” option 4. System displays a list of all data containers in alphabetical order with a search bar for refined criteria 5. User selects a data container 6. User selects “Edit” 7. System displays the selected data container 8. User makes changes to the data container and selects “Save” 9. All changes are saved into the database
Postconditions:	1. System saves the data container edits into the Formula Car Database
Alternative Courses:	If the user wishes to delete the data container (Occurs at step 9) 9a. User selects the “Delete” option 9b. System displays a warning message: “Are you sure you want to delete this Data Container?” 9c. User selects the “Yes” option 9d. System deletes the Data Container and displays message: “Data Container has been deleted”

	Continue at step 2.
Exceptions:	
Priority:	Medium

Title:	Create Data Set
Description:	This use case describes the process of creating a new data set.
Primary Actor:	Formula Member
Trigger:	User wishes to create a new data set.
Preconditions:	1. User must be logged into the system
Normal Course:	1. User selects the “Data” option on the menu 2. System displays the data landing page 3. User selects the “Create Data Set” option 4. System prompts user to create a title for the data set 5. User enters a title and selects “Continue” 6. System prompts user to select the data container in which this file will be stored a. Listed alphabetically with search bar option for precise criteria 7. User selects a data container 8. User selects “Create” 9. System opens a new blank data set within the chosen data container 10. System saves data set within the data container into the database .
Postconditions:	1. Data set is saved in the Formula Car Database
Alternative Courses:	

Exceptions:	The user enters data set title already exists (occurs at step 5) 1. System displays an error message: “Data set title already exists. Please enter a different title.” Continue at step 4.
Priority:	High

Title:	View Data Set
Description:	This use case describes the process of viewing a data set.
Primary Actor:	Formula Member
Trigger:	User wishes to view an existing data set.
Preconditions:	1. User must be logged into the system 2. At least one data set must exist within the system
Normal Course:	1. User selects the “Data” option on the menu 2. System displays the data landing page 3. User selects the ”View” option 4. System displays a list of all data sets in alphabetical order with a search option for refined criteria 5. User selects a data set 6. User selects “View” 7. System displays the selected data set
Postconditions:	
Alternative Courses:	
Exceptions:	
Priority:	High

Title:	Edit Data Set
Description:	This use case describes the process of editing an existing data set.
Primary Actor:	Formula Member
Trigger:	User wishes to make changes to a data set.
Preconditions:	1. User must be logged into the system 2. At least one data set must exist within the system
Normal Course:	1. User selects the “Data” option in the menu 2. System displays the data landing page 3. User selects the “Edit Data Set” option 4. System displays a list of all data sets in alphabetical order with a search bar for refined criteria 5. User selects a data set 6. User selects “Edit” 7. System displays the selected data set 8. User makes changes to the data set and selects “Save” 9. All changes are saved into the database
Postconditions:	1. Edits made to the data set is updated into the Formula Car Database

Alternative Courses:	If the user wishes to delete the data container (Occurs at step 9) 9a. User selects the “Delete Data Set” option 9b. System displays a warning message: “Are you sure you want to delete this Data Set?” 9c. User selects the “Yes” option 9d. System deletes the Data Set and displays message: “Data Set has been deleted” Continue at step 2.
Exceptions:	
Priority:	High

Title:	Request Data Transfer
Description:	This use case describes the process of requesting a data transfer between two people within different teams.
Primary Actor:	Formula Member
Trigger:	User wishes to retrieve data from another member/team within the system.
Preconditions:	1. User must be logged into the system 2. At least one data set exists in the system 3. At least one team must exist within the system
Normal Course:	1. User selects the “Data” option in the menu 2. System displays the data landing page 3. User selects the “Data Transfer” option 4. System returns a list of all data sets sorted by alphabetical order, with a search option for refined criteria 5. User selects the desired data set(s) 6. User selects “Request Transfer” 7. System displays transfer request dialogue a. Selected data sets b. Entry for Member ID sending request c. Entry for Member ID receiving request 8. User enters corresponding Member IDs 9. User selects “Confirm” 10. System submits request to the Member ID 11. System marks request as “Pending” and assigns a unique ID number to the transfer 12. System saves request to the database
Postconditions:	1. Request is sent to the receiving member 2. Data request is assigned a unique ID number 3. Data request is updated in the database marked as “Pending”.

Alternative Courses:	
Exceptions:	User enters a Member ID that does not exist (Occurs at step 10) 1. System displays error message: “Invalid Member ID(s). Please try again” Continue at step 9.
Priority:	High

Title:	Resolve Data Transfer Request
Description:	This use case describes the process of resolving a data transfer request.
Primary Actor:	Formula Member
Trigger:	User wishes to approve or deny an existing data transfer request.
Preconditions:	1. User must be logged into the system 2. At least one pending data transfer request is present in the user's inbox
Normal Course:	1. User selects the "Inbox" icon 2. System displays inbox page 3. User selects the "Data Requests" Option 4. System displays a list of pending data transfer requests sorted by Request ID, with a search bar and filter for refined search criteria a. Resolved b. Saved for Later c. Pending 5. User enters search criteria and selects "Search" 6. System displays all corresponding transfer requests 7. User selects a request 8. System displays the data transfer request information 9. User selects the "Approve" or "Deny" option 10. System updates the data request details to database, marking it as "Resolved" 11. System sends the updated request back to the requesting team member.
Postconditions:	1. System sends updated request back to the requesting team member 2. System marks request as "Resolved"

	3. Data request details are updated in the Formula Car Database
Alternative Courses:	If the user wishes to come back to the request without finishing (Occurs at step 8) 8a. User selects the “Save For Later” option 8b. System assigns a unique ID to the request 8a. System marks request as “Saved For Later” Continue at step 4.
Exceptions:	
Priority:	High

Title:	Save Item Receipt
Description:	This use case describes the process of adding an item receipt from purchased materials to the database.
Primary Actor:	Formula Member
Trigger:	User wishes to save a new item’s receipt to the database.
Preconditions:	1. User must be logged into the system
Normal Course:	1. User selects the “Financial” option on the menu 2. System displays the financial landing page 3. User selects the “New Item Receipt” option 4. System displays the new item receipt dialogue a. Item name b. Item description c. Item price d. Item Quantity e. Date purchased 5. User enters required information and selects “Save” 6. System saves receipt to the database
Postconditions:	1. New item receipt is saved in the Database
Alternative Courses:	
Exceptions:	
Priority:	Medium

Title:	Generate Reports and Summaries
Description:	This use case describes the process of generating a report or a summary.
Primary Actor:	Formula Member
Trigger:	User wishes to compile a report or a summary.
Preconditions:	1. User must be logged into the system
Normal Course:	1. User selects the “Project” option on the menu 2. System displays the project landing page 3. User selects the “Generate Reports & Summaries” option 4. System displays dialogue a. Report/Summary Title b. Documents/Data sets included c. Date range d. How the report/summary is presented e. File type of saved document 5. User enters all fields of the dialogue 6. User selects the “Save” option 7. System saves document in the database 8. System displays the report/summary
Postconditions:	1. Summary/report is saved in the databases
Alternative Courses:	

Exceptions:	User does not enter information into one or more of the presented fields (Occurs at step 5) 1. System displays error message: “Please complete all mandatory fields” 2. System highlights fields left blank Continue at step 4.
Priority:	High

Title:	View Reports and Summaries
Description:	This use case describes the process of viewing a report or a summary.
Primary Actor:	Formula Member
Trigger:	User wishes to view a report or a summary.
Preconditions:	1. User must be logged into the system
Normal Course:	1. User selects the “Project” option on the menu 2. System displays the project landing page 3. User selects the “View Reports & Summaries” option 4. System displays dialogue listing all previously created reports/summaries in alphabetical order a. Search bar for refined criteria b. Filter by date and name 5. User selects a document 6. User selects “View” 7. System displays the document
Postconditions:	
Alternative Courses:	
Exceptions:	

Priority:	High

Title:	Project Proposal
Description:	This use case describes the process of proposing a new project to a team or manager.
Primary Actor:	Formula Member
Trigger:	User wishes to create a new project proposal.
Preconditions:	1. User must be logged into the system
Normal Course:	1. User Selects the “Project” option on the menu 2. System displays the project landing page 3. User selects the “New Proposal” option 4. System displays the new proposal dialogue a. Proposal name b. Project description c. Budget estimate d. Timeline estimate e. Project objective f. Sending Member name g. Sending Member ID h. Receiving Team ID 5. User enters data into the dialogue and selects “Save” 6. System saves new project proposal 7. System displays a confirmation page 8. User selects the “Complete Proposal” option 9. System sends project proposal to corresponding team
Postconditions:	1. Proposal is saved in the database 2. Proposal is sent to the corresponding team manager

Alternative Courses:	User wishes to save an incomplete proposal to be finished at a later time (Occurs at step 6) 3. User selects the “Save For Later” option 4. System saves the proposal and marks it as “In Progress” in the database
Exceptions:	Use does not enter data into one or more required fields (Occurs at step 3) 1. System displays error message “Please complete all mandatory fields” with two options: a. “Save For Later” b. Edit Proposal 2. User selects corresponding option a. “Save For Later” i. System saves the proposal in the database marking it accordingly b. Edit Proposal i. Continue at step 4.
Priority:	High

Title:	Pull Data
Description:	This use case describes the process of retrieving data from the data stores for various purposes.
Primary Actor:	System
Trigger:	A part of the system requests data from data stores
Preconditions:	1. User must be logged into the system
Normal Course:	1. System requests data en masse from the data stores 2. System pulls data from one or more of the data stores 3. System sends the data to where it was requested from
Postconditions:	1. The data transfer and pull must be logged into the database
Alternative Courses:	Data request is incorrect, no data store matching name (Occurs at step 1) 1a. A “incorrect data request” alert is sent to the origin of request 1b. System ends data pull process with no message No correlating data found (occurs at step 2) 2a. A “no data found” alert is sent to the origin of request 2b. System ends data pull process with no message
Exceptions:	
Priority:	High

Title:	System Backup
Description:	This use case describes the process of automatically updating the system.
Primary Actor:	System, Formula Member
Trigger:	User makes any new changes or edits within the system.
Preconditions:	1. The automatic backup feature is enabled in the system configuration.
Normal Course:	1. User makes an edit, change, or enters new data into the system 2. System automatically completes a system backup to the database.
Postconditions:	1. System instantaneously completes the backup, preserving all user data within the Managerial Database.
Alternative Courses:	User wants to manually backup the system (Occurs at step 1) 1a. User selects the “Backup Now” option 1b. System completes a backup
Exceptions:	
Priority:	High

Title:	Archive Data
Description:	This use case describes the process of archiving data that is no longer relevant, but still needs to be saved in the database.
Primary Actor:	Formula Member
Trigger:	User wishes to archive data in the system.
Preconditions:	1. User must be logged into the system 2. At least one data set must exist within the system 3. At least one data container must exist within the system
Normal Course:	1. User selects the “Data” option on the menu 2. System displays the data landing page 3. User selects the “Archive Data” option 4. System displays the archive data dialogue a. Search bar for precise criteria b. Filter by data container, data set, date range 5. User selects data they wish to archive 6. User selects the “Archive” option 7. System displays message: “Are you sure you want to archive this data?” 8. User selects “Yes” 9. System moves selected data into the Archive Database
Postconditions:	1. Data is moved into the Archive Database
Alternative Courses:	

Exceptions:	User selects “No” (Occurs at step 10) 1. System saves selects and continues at step 6
Priority:	Medium

Title:	View Archived Data
Description:	This use case describes the process of viewing archiving data.
Primary Actor:	Formula Member
Trigger:	User wishes to view previously archived data in the system.
Preconditions:	1. User must be logged into the system 2. At least one document must exist within the database
Normal Course:	1. User selects the “Data” option on the menu 2. System displays the data landing page 3. User selects the “View Archived Data” option 4. System displays all the archive data in alphabetical order a. Search bar for precise criteria b. Filter by data container, data set, date range 5. User selects data they wish to archive 6. User selects “View” 7. System displays the archived data
Postconditions:	n/a
Alternative Courses:	
Exceptions:	
Priority:	Medium

Title:	Retrieve Data
Description:	This use case describes the process of retrieving data from a third party to be stored in the system and its databases.
Primary Actor:	System
Trigger:	External data source is connected with new data
Preconditions:	1. External data source must be verified and secured 2. The external file format must be logged and ready to be accepted 3. Virus protection must be active
Normal Course:	1. External storage is inserted into the system 2. System checks file type and readability 3. System checks for malware 4. System accepts data and feeds it to data storage 5. System Logs data transfer
Postconditions:	1. External data storage is ejected
Alternative Courses:	External storage is unreadable (Occurs in step 1) 1a. System displays “Storage Unreadable” message and ejects external storage File type is unknown (Occurs in step 2) 2a. System displays “File Type Unknown” message and ejects external storage Malware detected (Occurs in step 3) 3a. System displays “Malware Detected” message and ejects external storage

Exceptions:	
Priority:	Medium

Discussion of Proposed Solution

The Cal Poly Formula SAE Communication And Storage System was designed to streamline and enhance the operations of the Cal Poly Racing team. Our system will simplify and centralize how different teams within the Cal Poly FSAE team manage and store data, as well as track progress.

First, our system offers a variety of data management features. It allows for creating, viewing, editing, and deleting data containers (folders) and sets (files). These functions are crucial for any organizations that need to store data on any sort of software. What makes our system unique is that it can take in data from external data sources and format it to the centralized storage. Currently, dozens of teams all use separate datastores and file types which can make broader analysis very difficult and record keeping impossible. By standardizing and organizing the information within the team through the use of this system, we solve those issues. The system's design ensures that data is both easy to store and to retrieve.

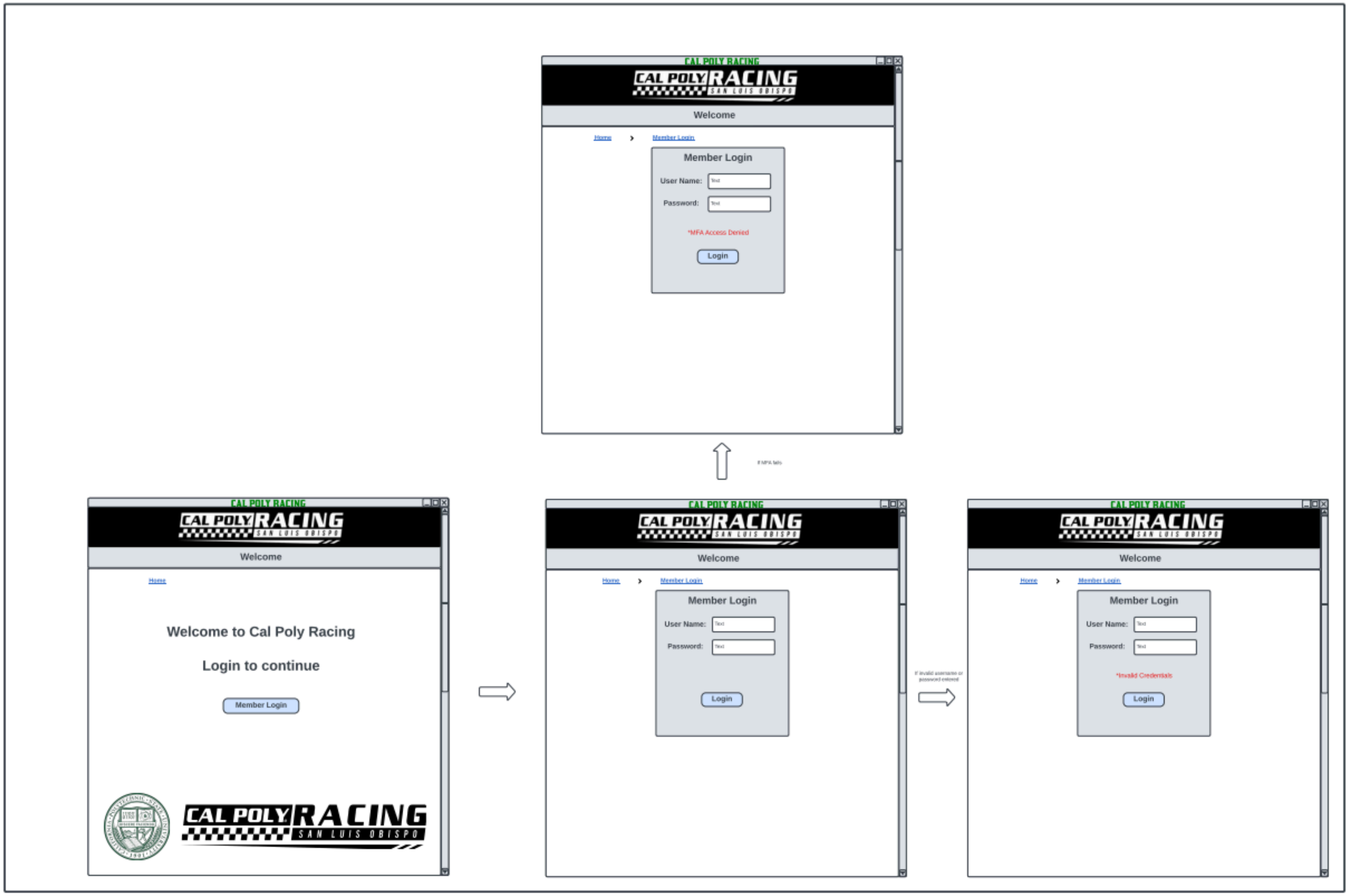
Second, a key aspect of the system is its user-specific access. The system controls what information each team member can see and modify, based on their assigned role and subsequent responsibilities. This not only makes the system more efficient by ensuring that a user sees only relevant information, but also adds a layer of security, guaranteeing sensitive data is only accessible to those who are authorized to view it. With this, the system also utilizes a multi-step data transfer request feature which requires the approval from another team lead in order for a member to gain access to any information that is not within the scope of their credentials. Our data security is further improved through multi-factor authentication, a method that adds an extra layer of protection against unauthorized access by requiring members to authenticate any logins with their personal devices in addition to their credentials.

Additionally, the system has specific features for member and team management, allowing for the creation and modification of member profiles and team structures. This ensures that all member information is current and that the team's organizational structure is transparent and easy to navigate. This structure helps the team minimize time spent figuring out who is responsible for a certain project along with who to contact if needed. This is especially useful when the team needs to request data and finding who is in charge of specific data such as budgeting or material usage is critical.

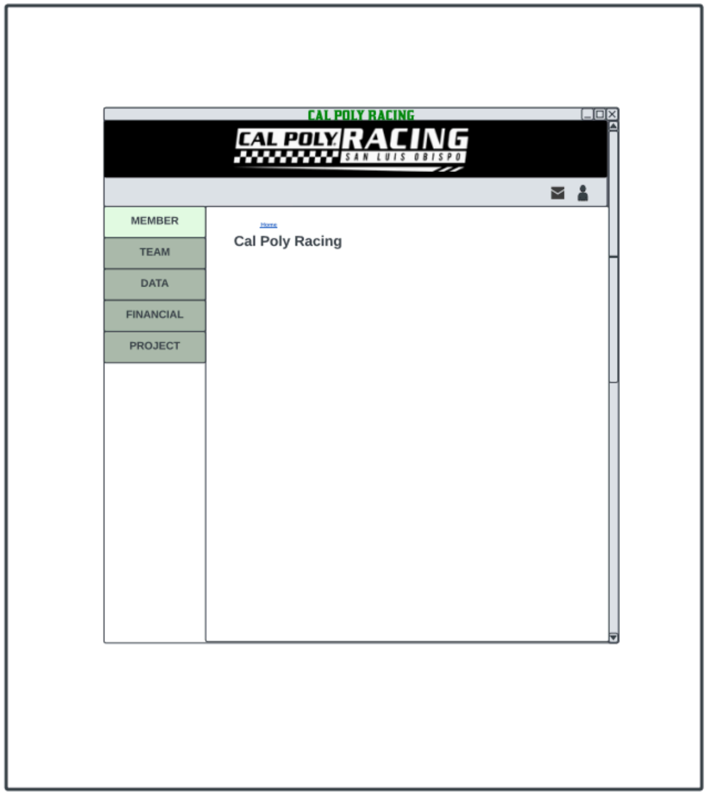
Finally, the system also handles reports and archiving. One of the use cases we created can be used as a tool for summarizing data and creating reports. This feature is extremely important for the organization in order to keep a close eye on their budget, material usage, man hours spent, team allocation and workloads, and spending. The system also has a backup and archiving feature which ensures that all data is securely stored and can be easily retrieved when needed. The backup timing and what is being backed up can be programmed to specifics, manually done, or both, depending on the needs of the user.

Design

Login with MFA

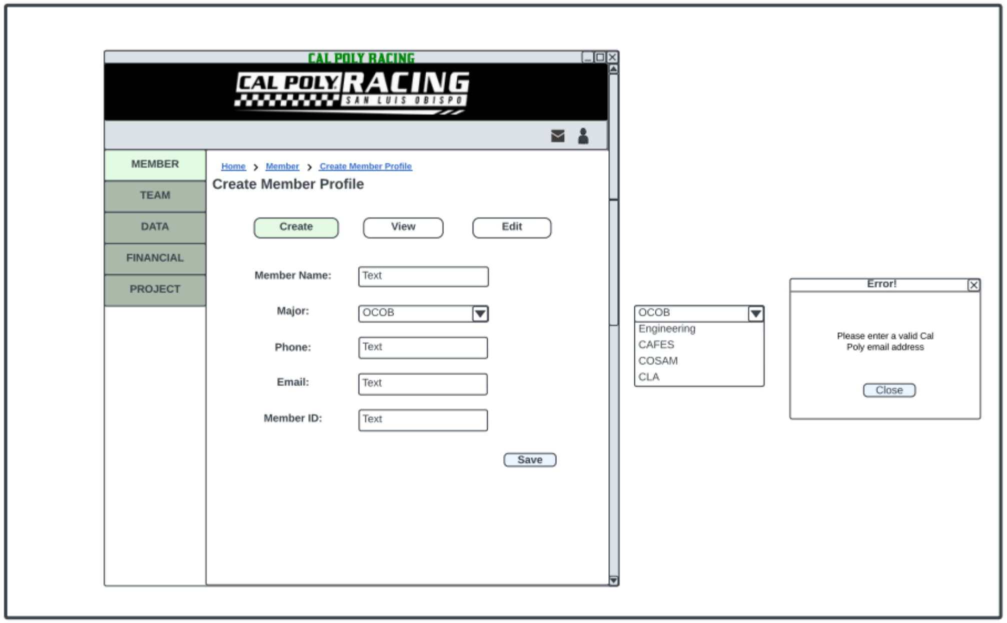


Home



Cal Poly Racing

Create Member Profile



Create Member Profile

Member Name: Text

Major: OCOB

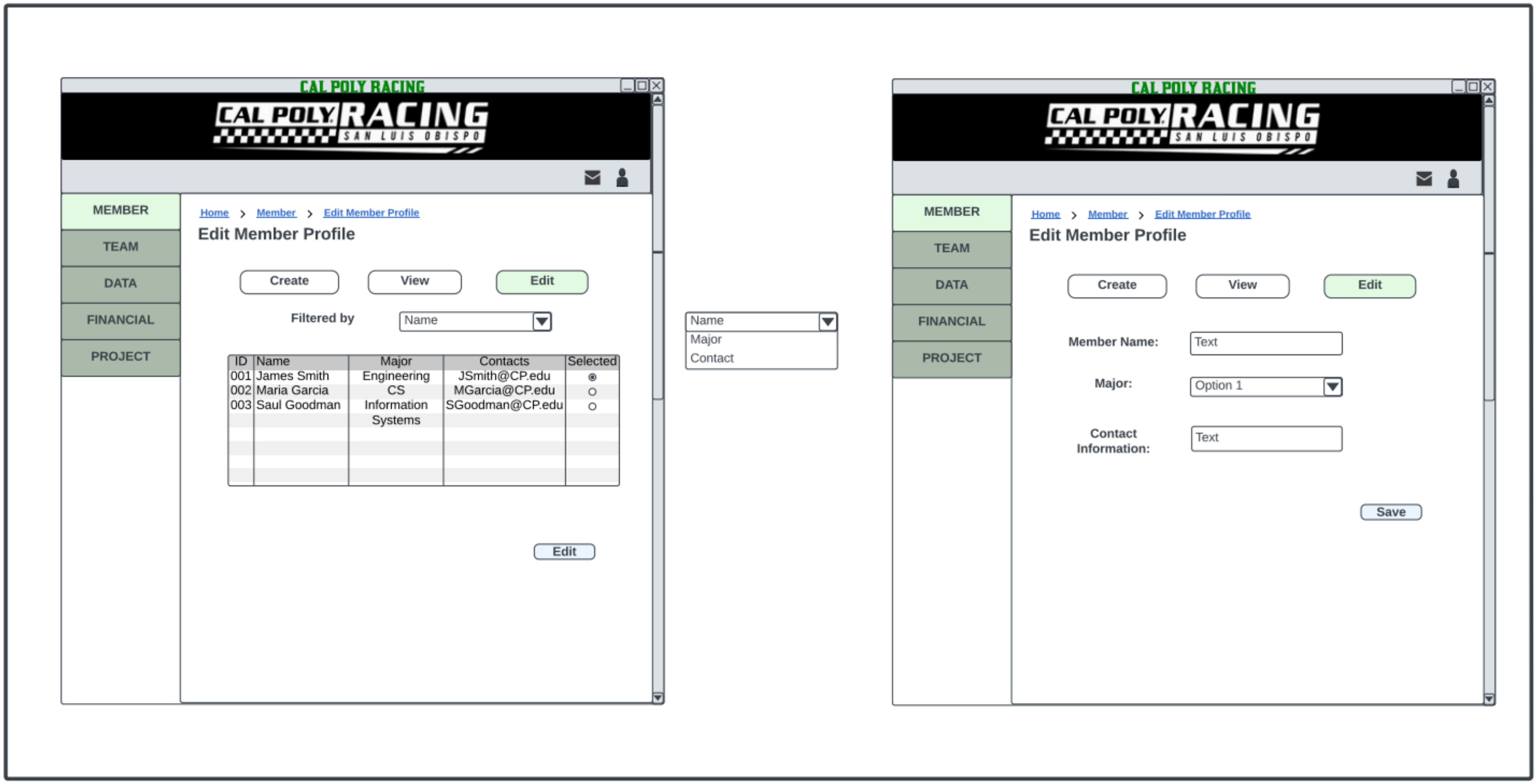
Phone: Text

Email: Text

Member ID: Text

Please enter a valid Cal Poly email address

Edit Member Profile



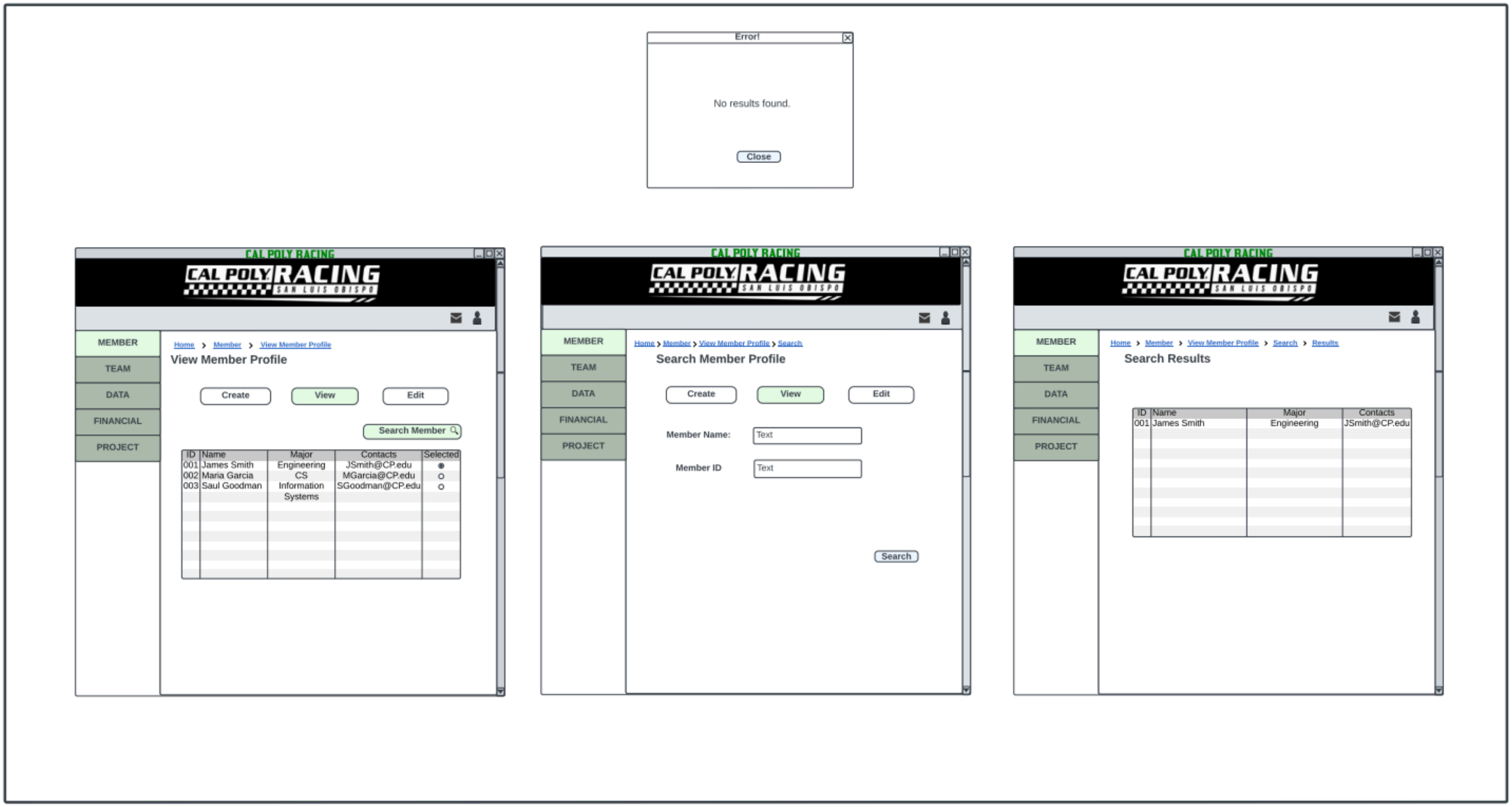
ID	Name	Major	Contacts	Selected
001	James Smith	Engineering	JSmith@CP.edu	●
002	Maria Garcia	CS	MGarcia@CP.edu	○
003	Saul Goodman	Information Systems	SGoodman@CP.edu	○

Member Name: Text

Major: Option 1

Contact Information: Text

View Member Profile


Error!
Close
CAL POLY RACING
CAL POLY RACING SAN LUIS OBISPO
MEMBER
TEAM
DATA
FINANCIAL
PROJECT
Home › Member › View Member Profile
Create
View
Edit
Search Member
CAL POLY RACING
CAL POLY RACING SAN LUIS OBISPO
MEMBER
TEAM
DATA
FINANCIAL
PROJECT
Home › Member › View Member Profile › Search
Create
View
Edit
Search
CAL POLY RACING
CAL POLY RACING SAN LUIS OBISPO
MEMBER
TEAM
DATA
FINANCIAL
PROJECT
Home › Member › View Member Profile › Search › Results

No results found.

View Member Profile

ID	Name	Major	Contacts	Selected
001	James Smith	Engineering	JSmith@CP.edu	●
002	Maria Garcia	CS	MGarcia@CP.edu	○
003	Saul Goodman	Information Systems	SGoodman@CP.edu	○

Search Member Profile

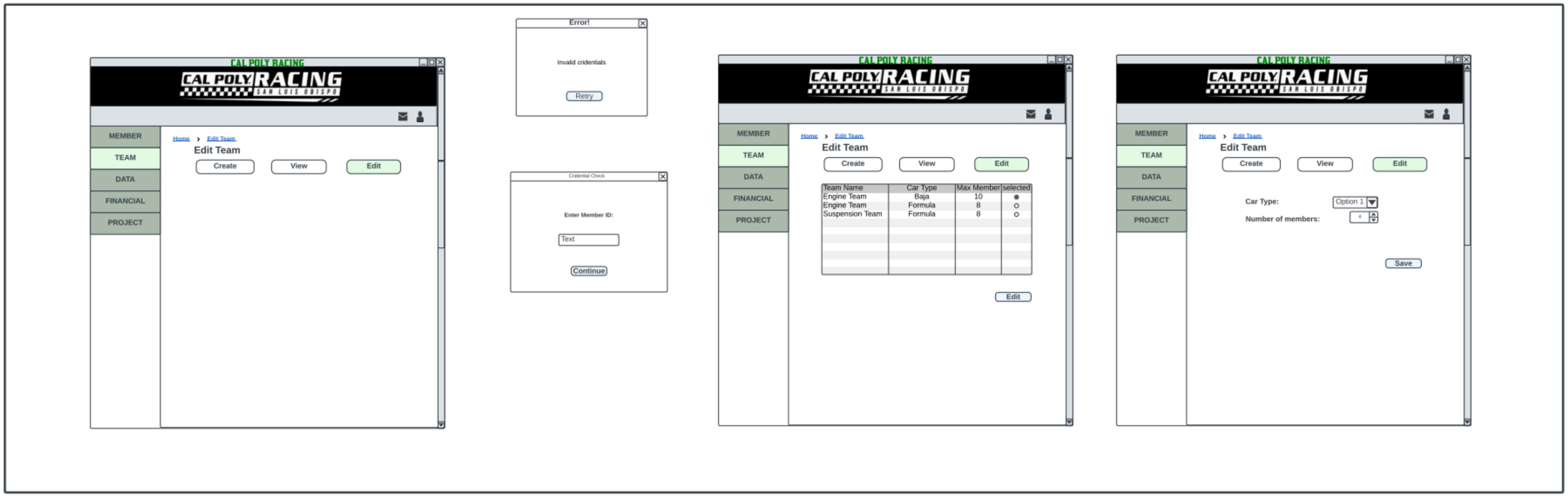
Member Name: Text

Member ID Text

Search Results

ID	Name	Major	Contacts
001	James Smith	Engineering	JSmith@CP.edu

Edit Team



Invalid cridentials

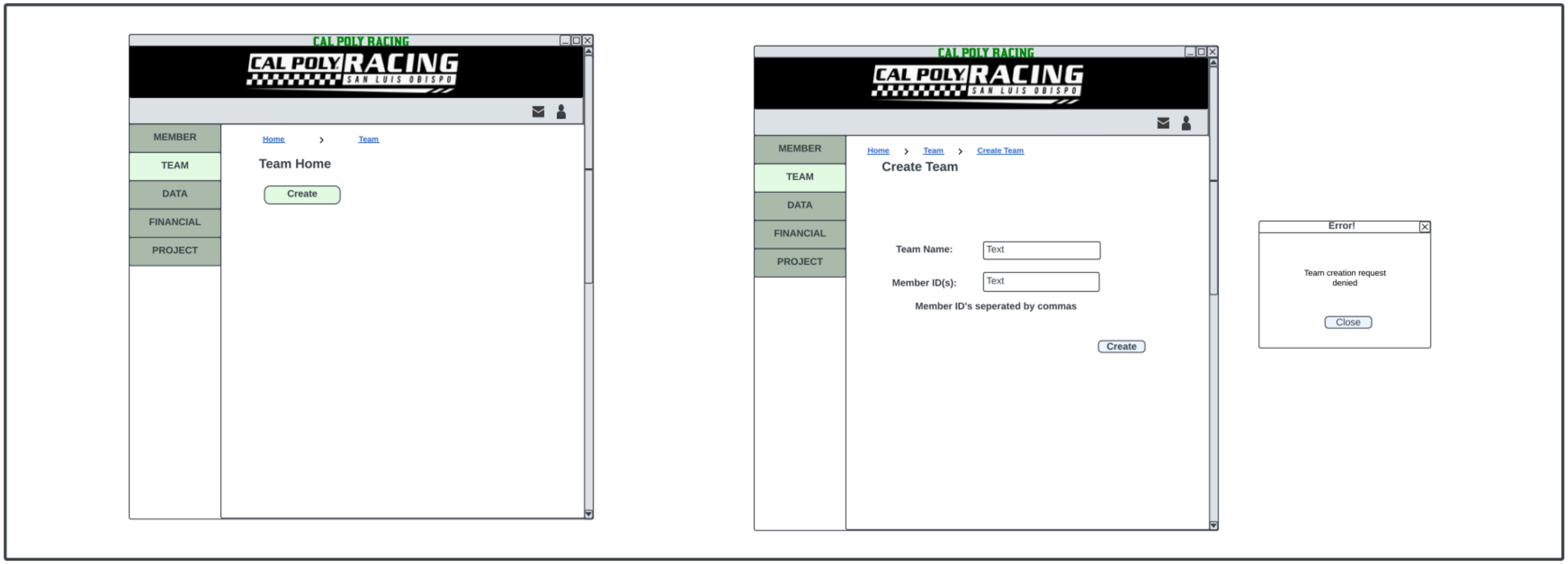
Enter Member ID: Text

Team Name	Car Type	Max Member	selected
Engine Team	Baja	10	●
Engine Team	Formula	8	○
Suspension Team	Formula	8	○

Car Type: Option 1

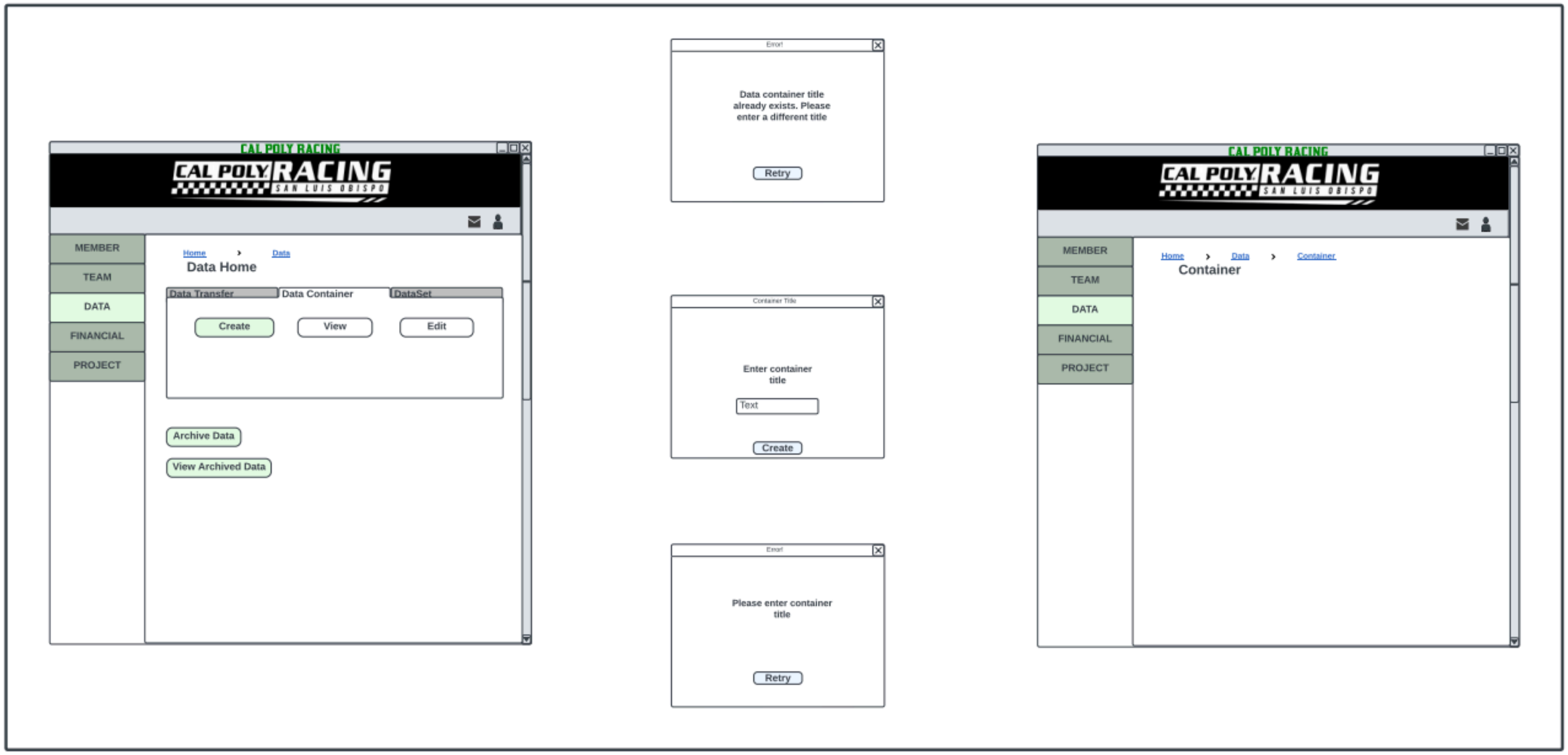
Number of members: 4

Create Team



Team creation request denied

Create Data Container

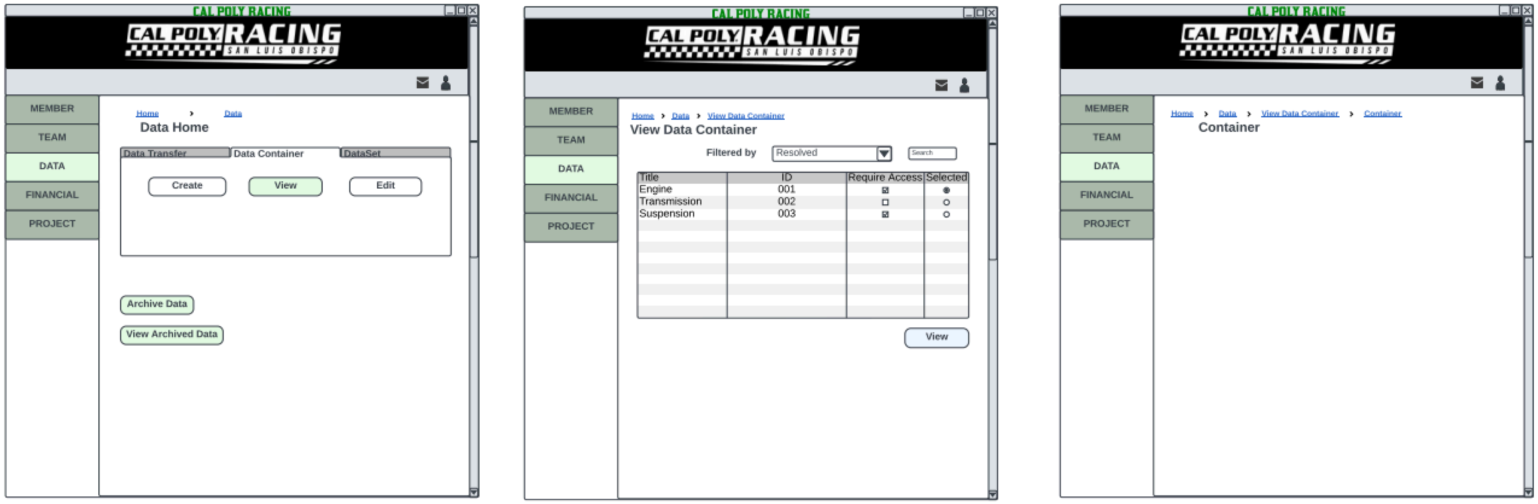


Data container title already exists. Please enter a different title

Enter container title

Please enter container title

View Data Container



Home › Data

Data Home

Home › Data › View Data Container

View Data Container

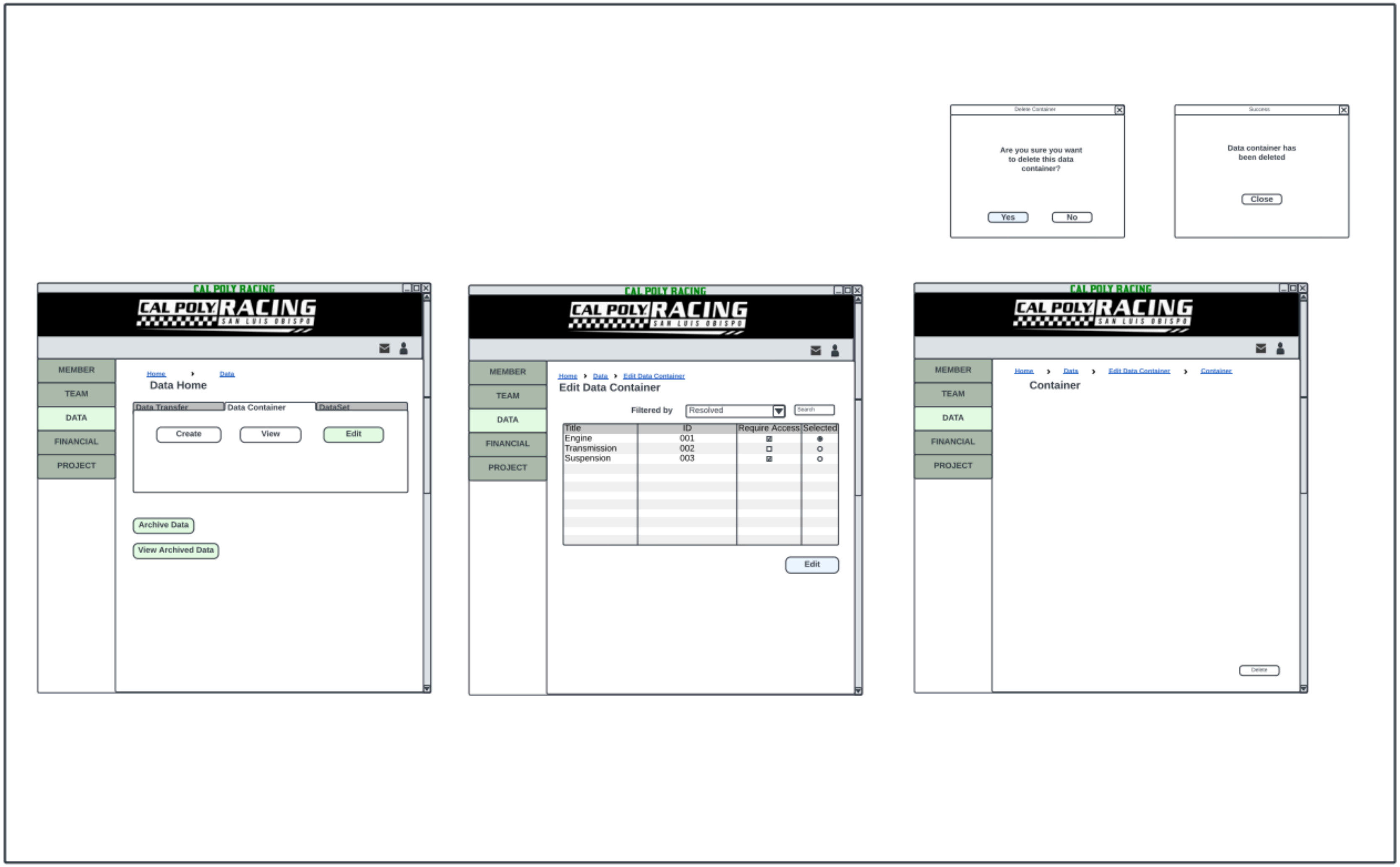
Filtered by: Resolved

Title	ID	Require Access	Selected
Engine	001	☒	●
Transmission	002	□	○
Suspension	003	☒	○

Home › Data › View Data Container › Container

Container

Edit Data Container



Home › Data › Edit Data Container

Edit Data Container

Filtered by Resolved ▼ Search

Title	ID	Require Access	Selected
Engine	001	☒	●
Transmission	002	□	○
Suspension	003	☒	○

Home › Data › Edit Data Container › Container

Container

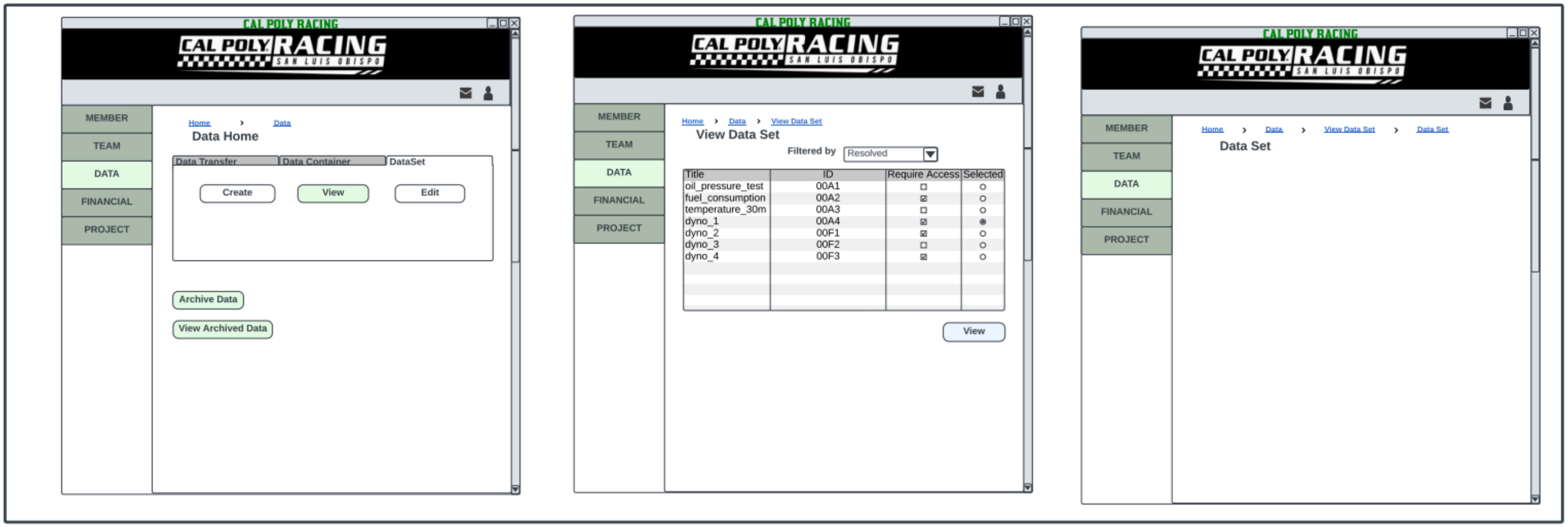
Delete Container

Are you sure you want to delete this data container?

Success

Data container has been deleted

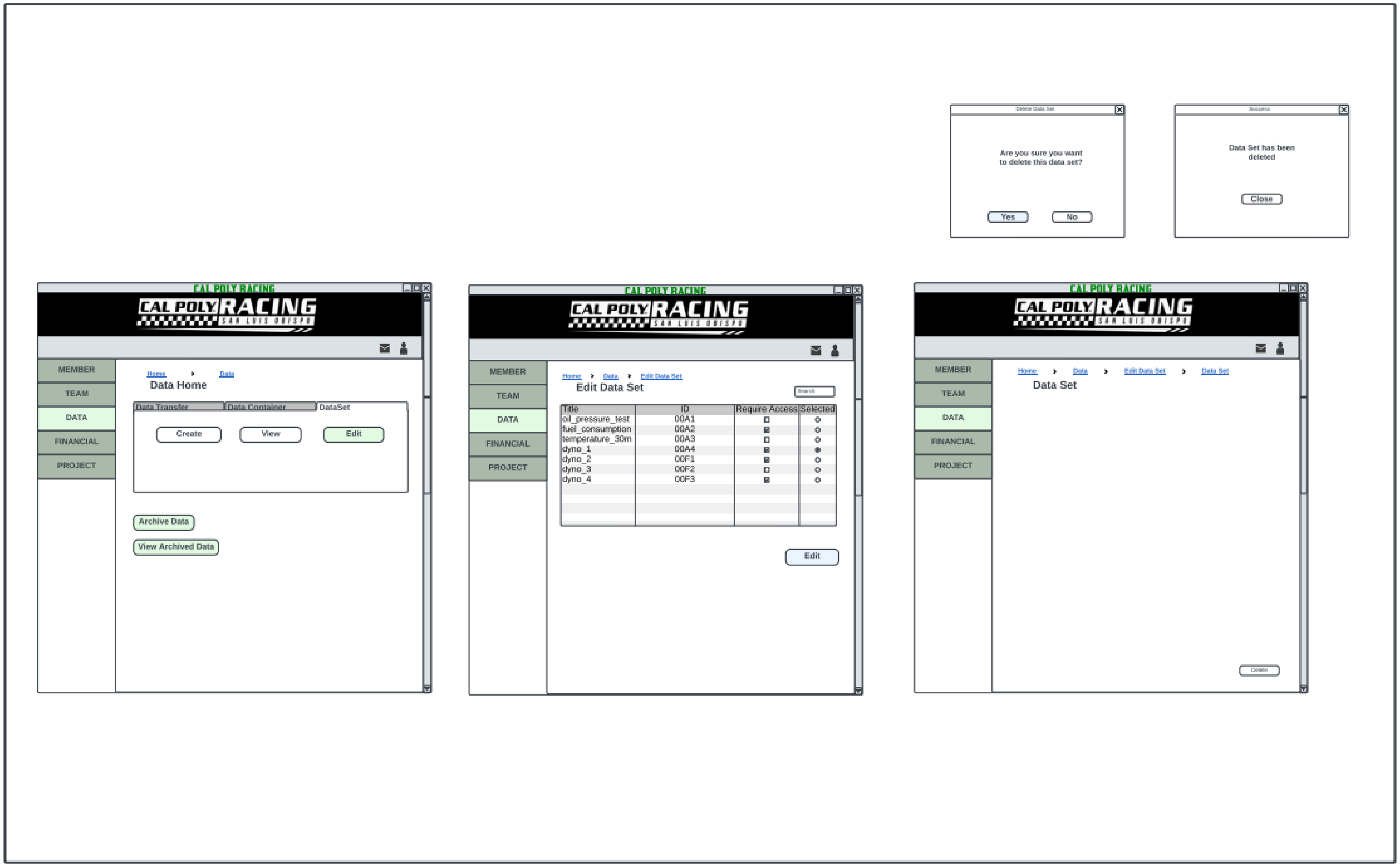
View Data Set



Filtered by Resolved

Title	ID	Require Access	Selected
oil_pressure_test	00A1	□	○
fuel_consumption	00A2	☒	○
temperature_30m	00A3	□	○
dyno_1	00A4	☒	●
dyno_2	00F1	☒	○
dyno_3	00F2	□	○
dyno_4	00F3	☒	○

Edit Data Set

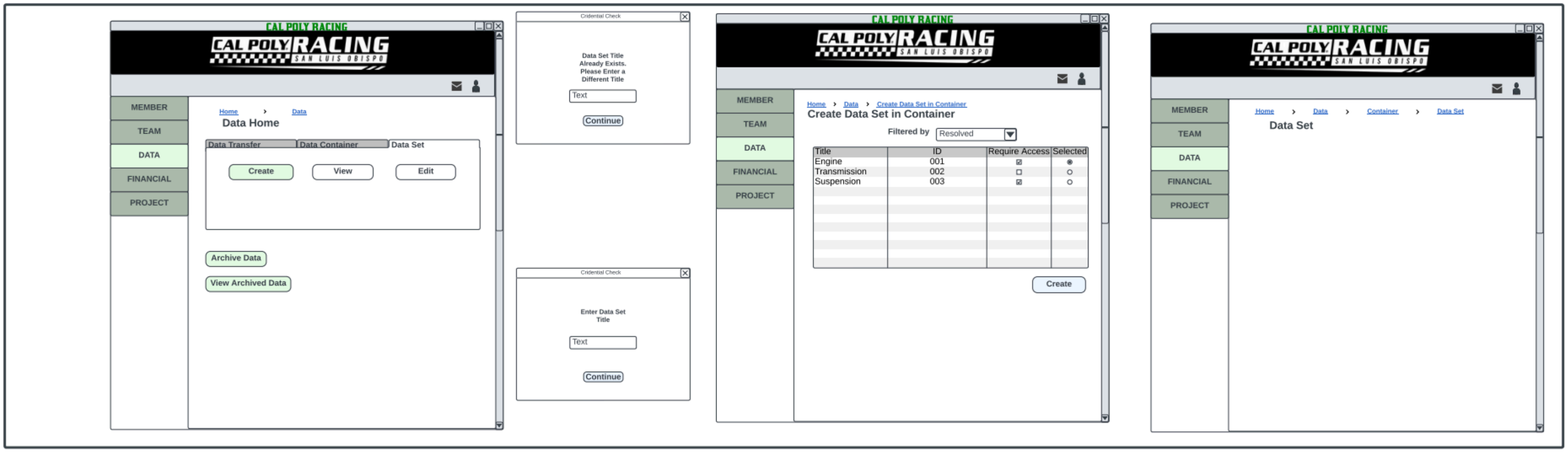


Title	ID	Require Access	Selected
oil_pressure_test	00A1	□	○
fuel_consumption	00A2	☒	○
temperature_30m	00A3	□	○
dyno_1	00A4	☒	●
dyno_2	00F1	☒	○
dyno_3	00F2	□	○
dyno_4	00F3	☒	○

Are you sure you want
to delete this data set?

Data Set has been
deleted

Create Data Set



Data Set Title Already Exists. Please Enter a Different Title

Text

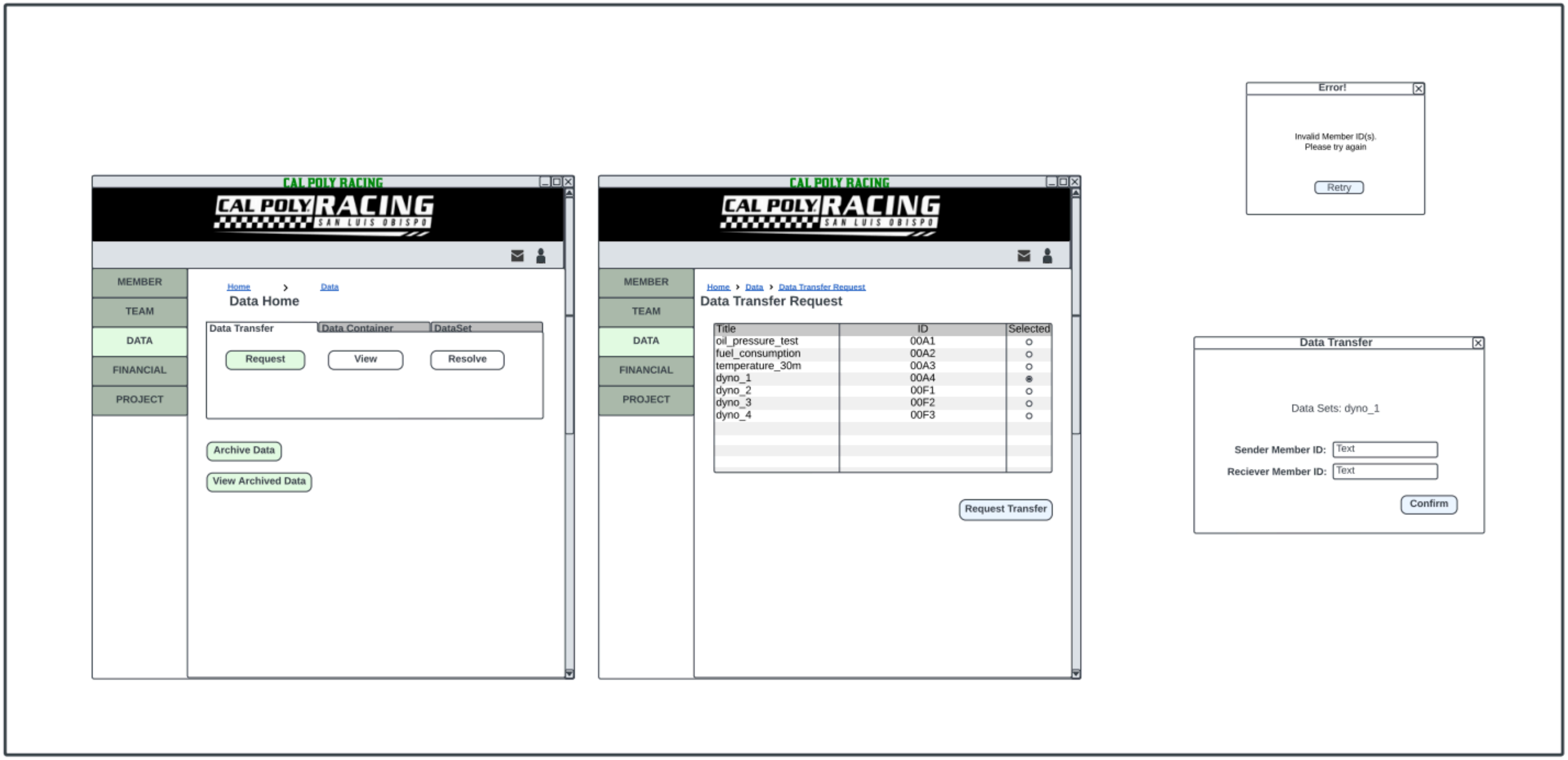
Enter Data Set Title

Text

Filtered by Resolved

Title	ID	Require Access	Selected
Engine	001	☑	●
Transmission	002	□	○
Suspension	003	☑	○

Request Data Transfer



Title	ID	Selected
oil_pressure_test	00A1	○
fuel_consumption	00A2	○
temperature_30m	00A3	○
dyno_1	00A4	●
dyno_2	00F1	○
dyno_3	00F2	○
dyno_4	00F3	○

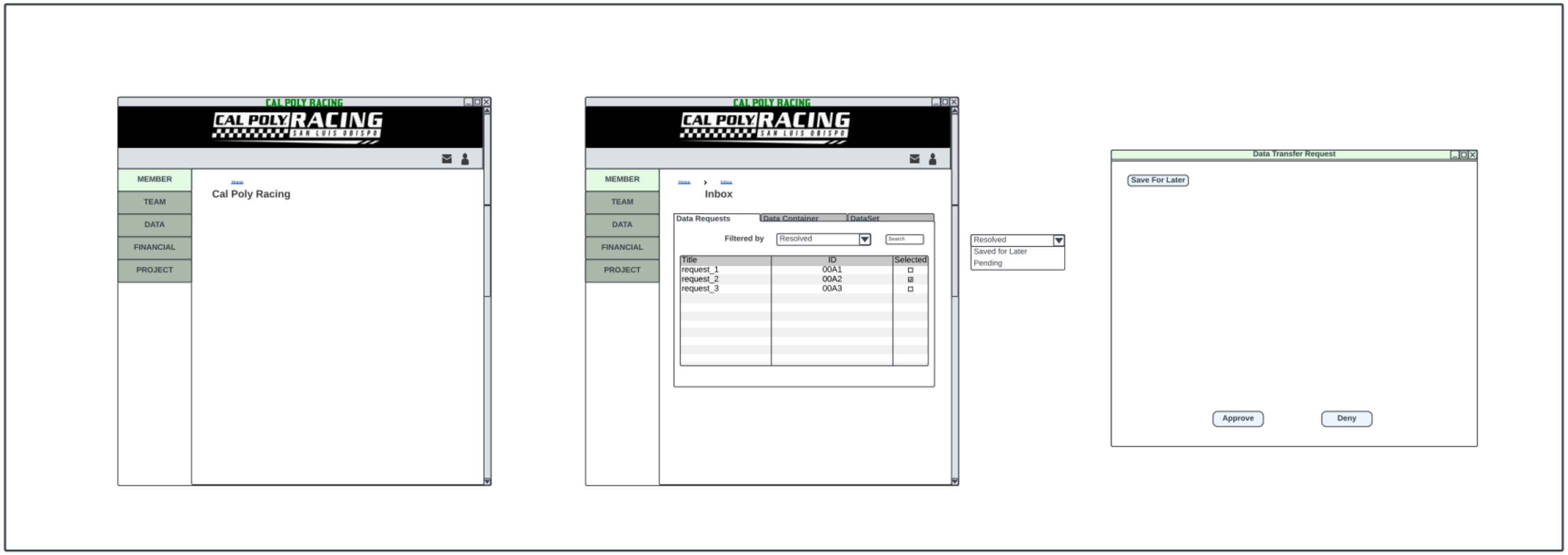
Invalid Member ID(s).
Please try again

Data Sets: dyno_1

Sender Member ID: Text

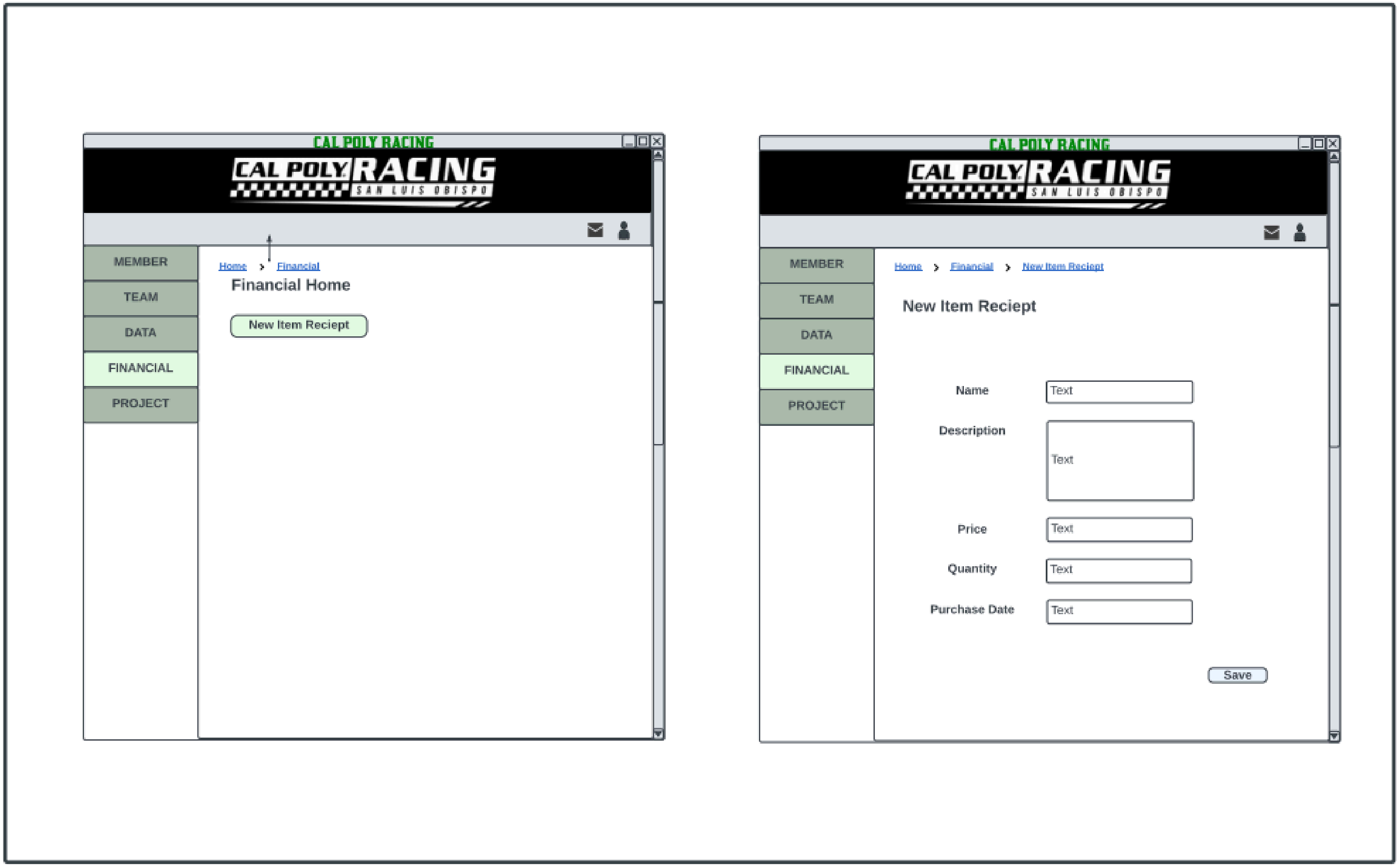
Reciever Member ID: Text

Resolve Data Transfer Request

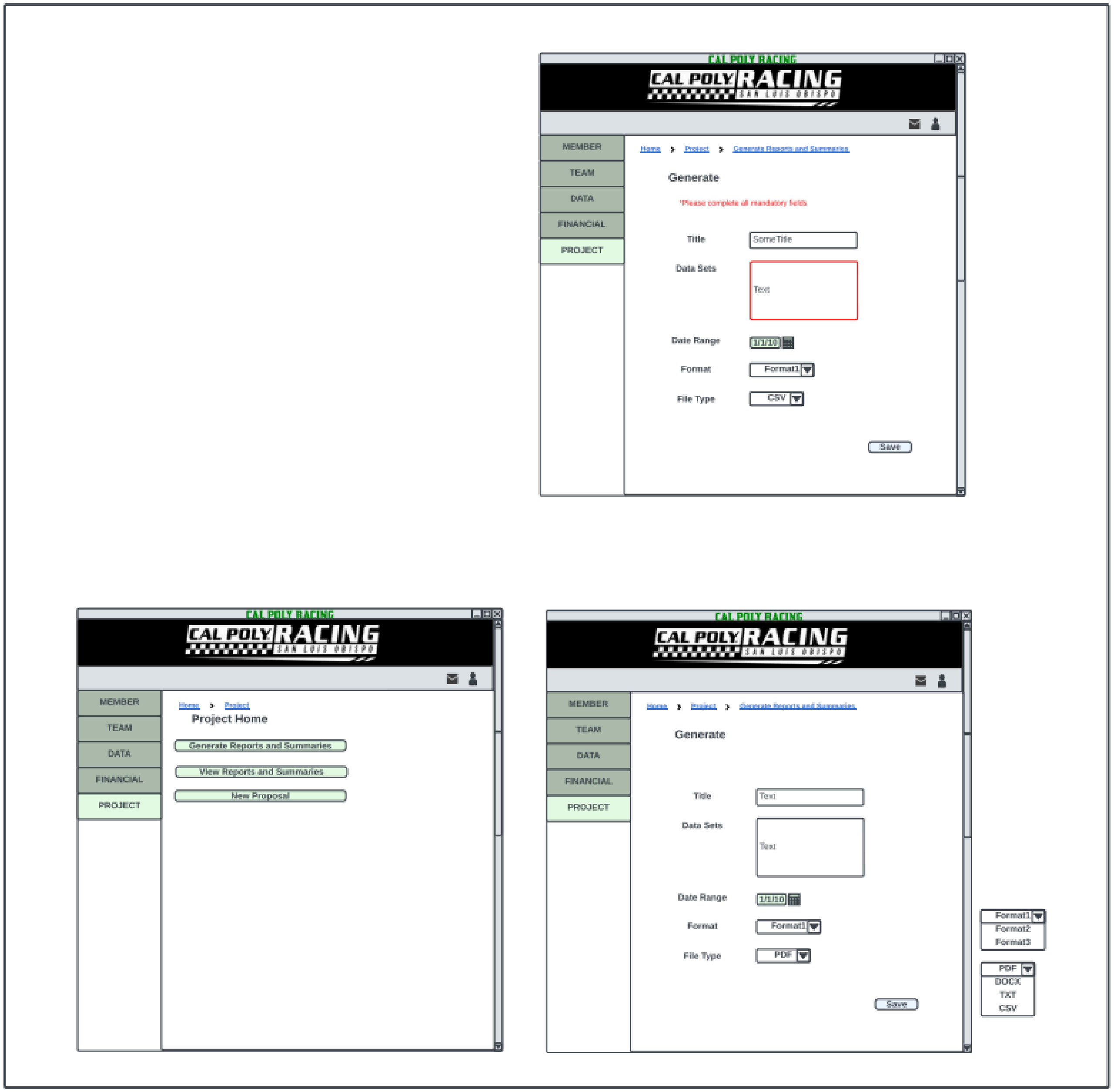


Title	ID	Selected
request_1	00A1	□
request_2	00A2	☑
request_3	00A3	□

Save Item Reciept



Generate Reports and Summaries


CAL POLY RACING
CAL POLY RACING SAN LUIS OBISPO
MEMBER
TEAM
DATA
FINANCIAL
PROJECT
Home > Project > Generate Reports and Summaries

Generate

*Please complete all mandatory fields

Title: SomeTitle

Data Sets: Text

Date Range: 1/1/10

Format: Format1

File Type: CSV

Save
CAL POLY RACING
CAL POLY RACING SAN LUIS OBISPO
MEMBER
TEAM
DATA
FINANCIAL
PROJECT
Home > Project

Project Home

Generate Reports and Summaries
View Reports and Summaries
New Proposal
CAL POLY RACING
CAL POLY RACING SAN LUIS OBISPO
MEMBER
TEAM
DATA
FINANCIAL
PROJECT
Home > Project > Generate Reports and Summaries

Generate

Title: Text

Data Sets: Text

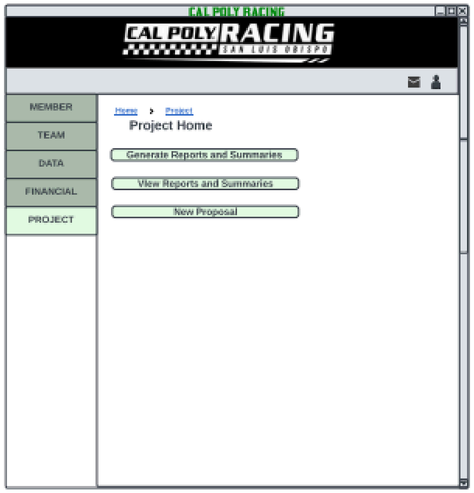
Date Range: 1/1/10

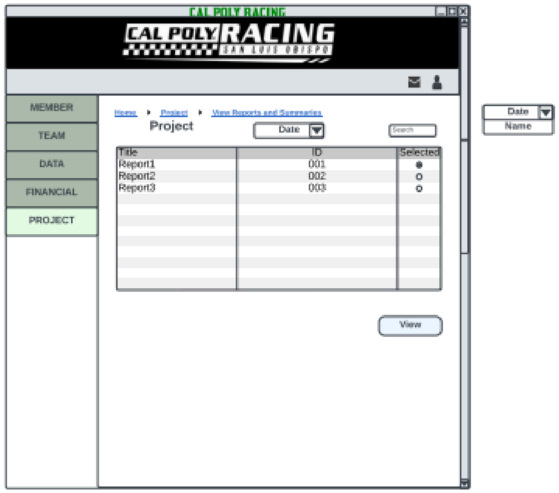
Format: Format1

File Type: PDF

Save
Format1
Format2
Format3
PDF
DOCX
TXT
CSV

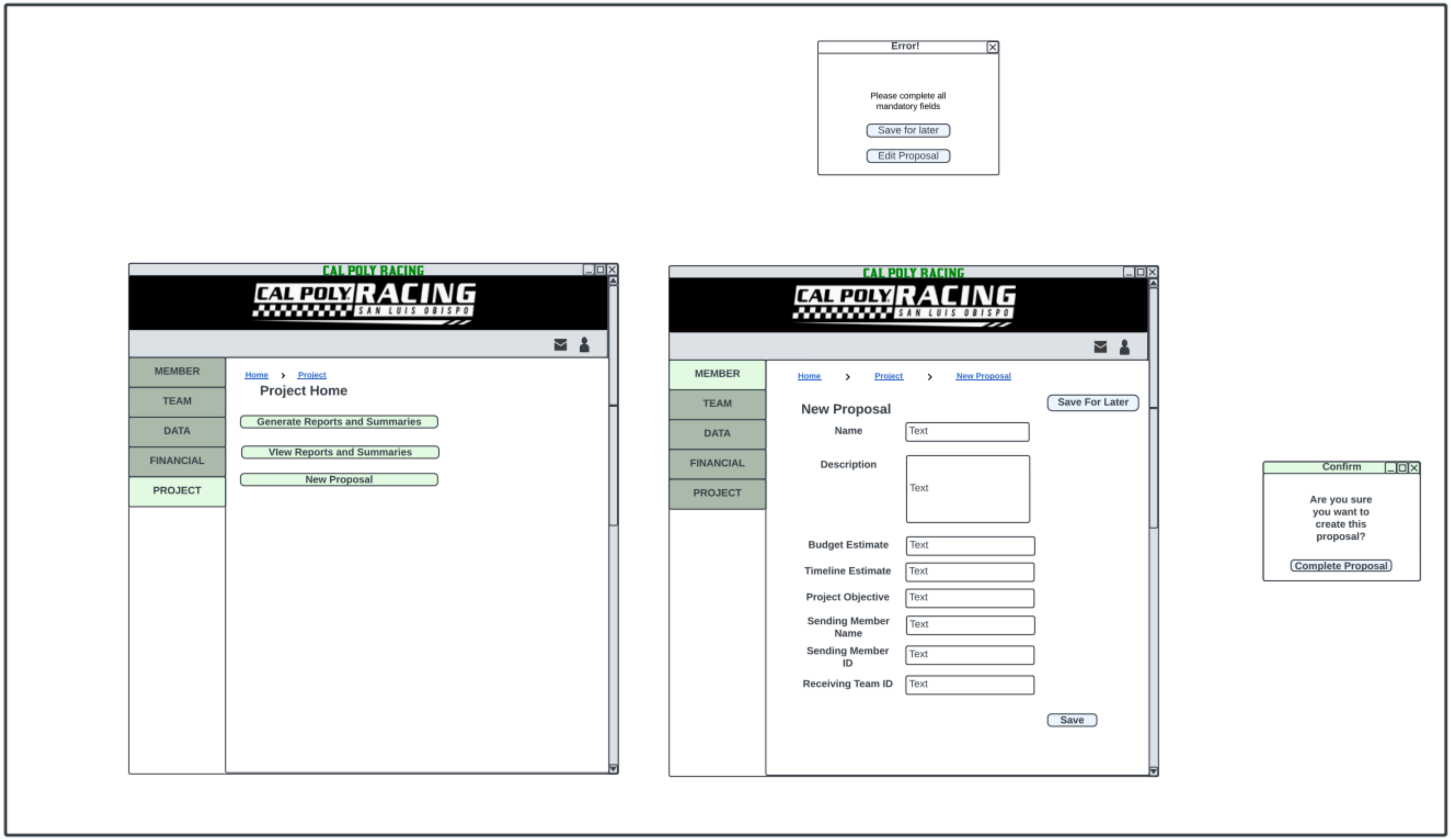
View Reports and Summaries





Title	ID	Selected
Report1	001	●
Report2	002	○
Report3	003	○

Project Proposal



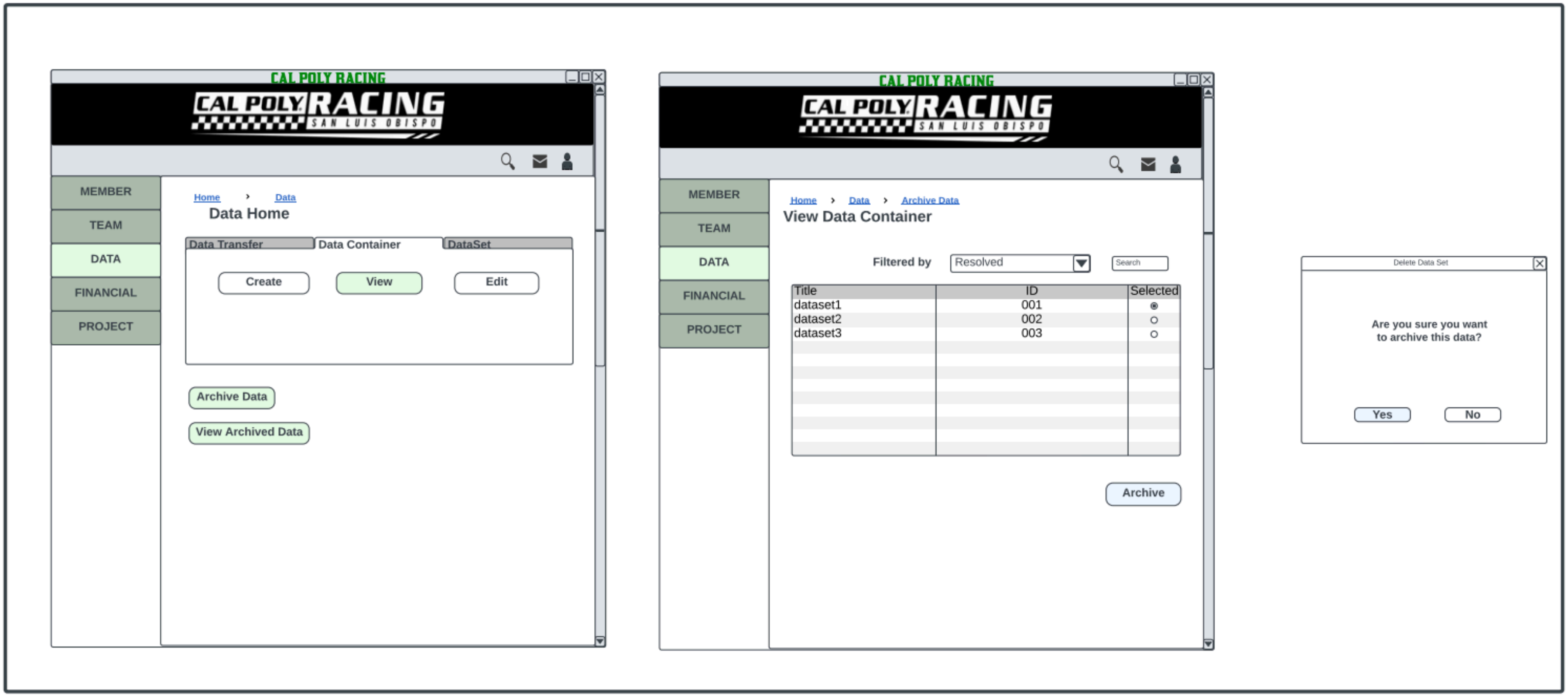
Please complete all mandatory fields

New Proposal

Name	Text
Description	Text
Budget Estimate	Text
Timeline Estimate	Text
Project Objective	Text
Sending Member Name	Text
Sending Member ID	Text
Receiving Team ID	Text

Are you sure you want to create this proposal?

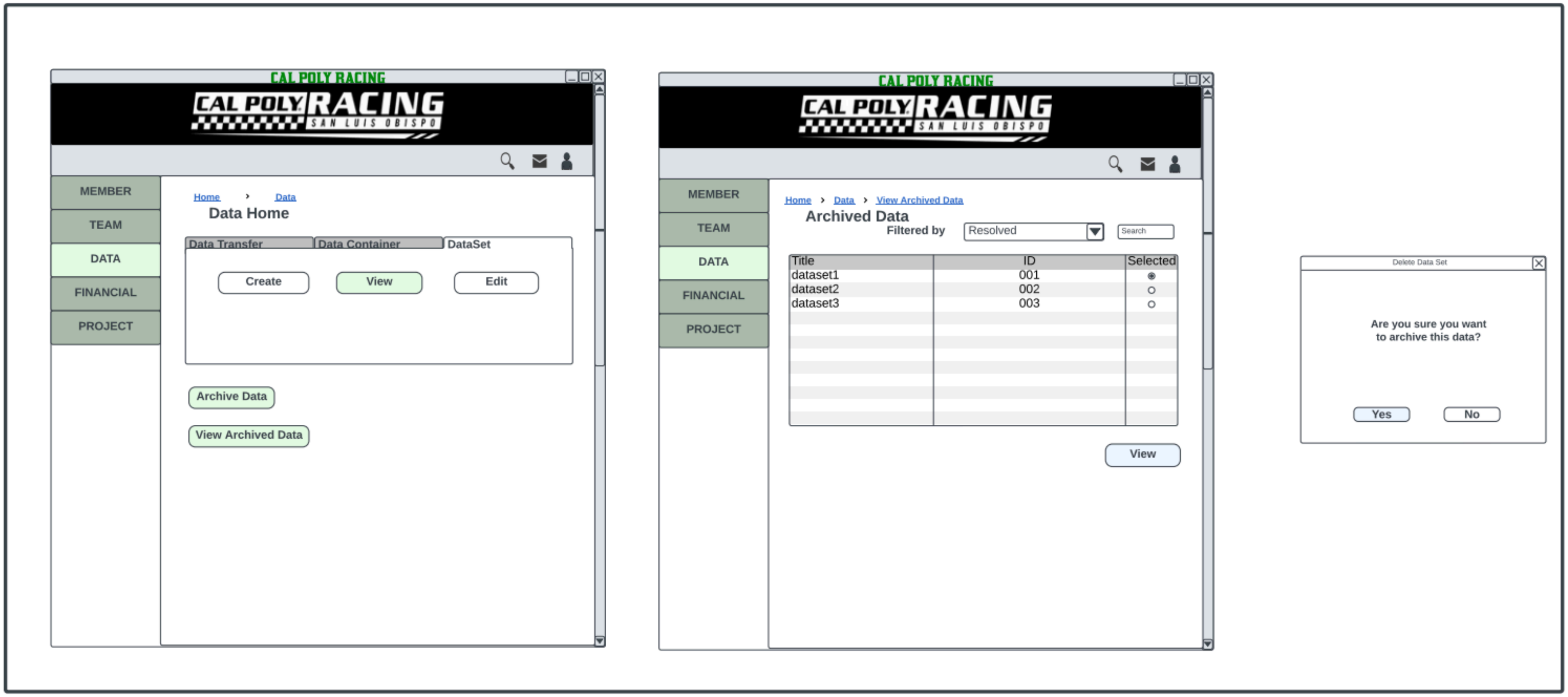
Archive Data



Title	ID	Selected
dataset1	001	●
dataset2	002	○
dataset3	003	○

Are you sure you want to archive this data?

View Archived Data

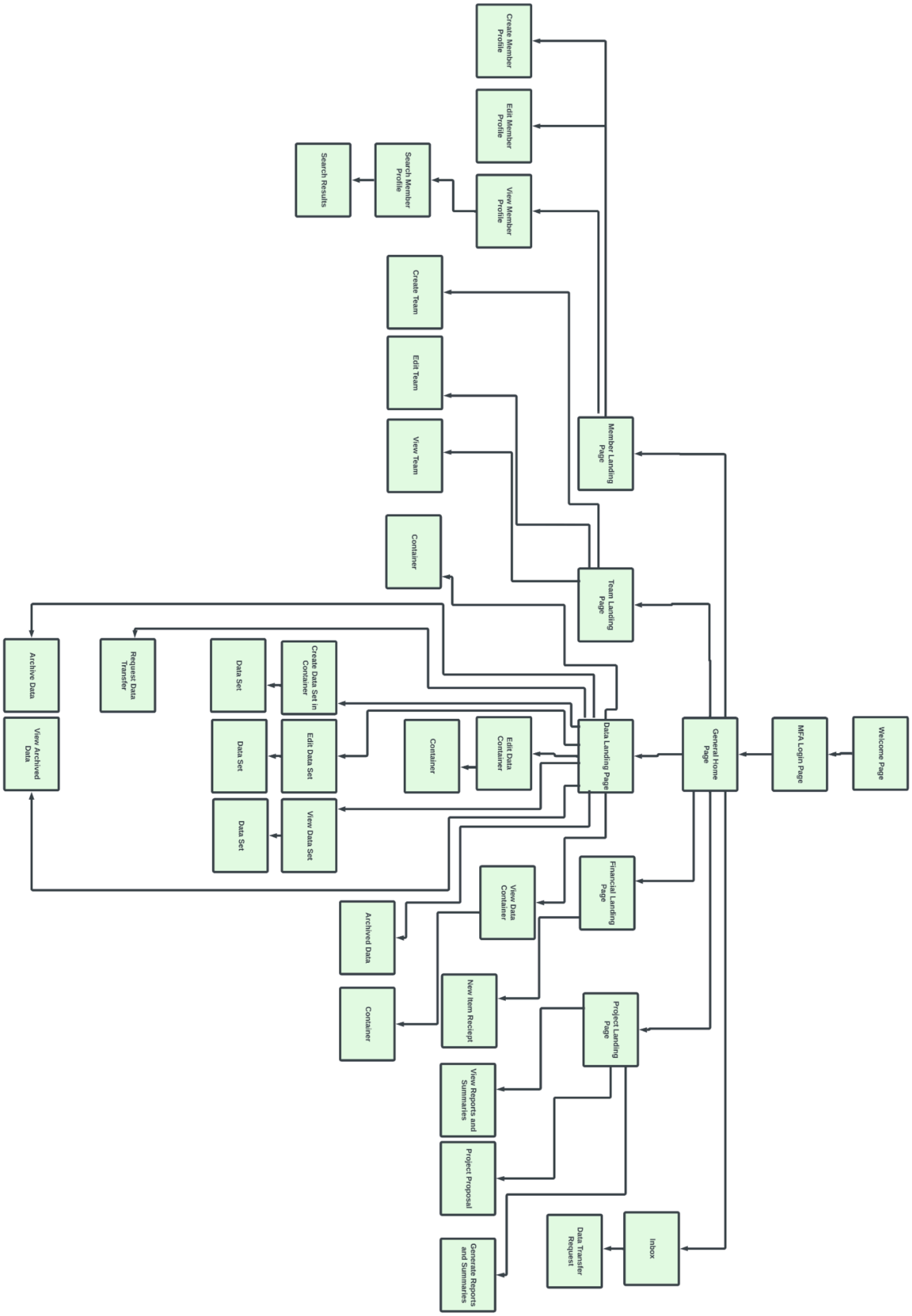


Filtered by Resolved

Title	ID	Selected
dataset1	001	●
dataset2	002	○
dataset3	003	○

Are you sure you want to archive this data?

UI Site Map:


Welcome Page
MFA Login Page
General Home Page
Data Landing Page
Member Landing Page
Team Landing Page
Financial Landing Page
Project Landing Page
Inbox
Data Transfer Request
Create Member Profile
Edit Member Profile
View Member Profile
Search Member Profile
Search Results
Create Team
Edit Team
View Team
Container
Archive Data
View Archived Data
Request Data Transfer
Create Data Set in Container
Data Set
Edit Data Set
Data Set
View Data Set
Data Set
Edit Data Container
Container
View Data Container
Archived Data
Container
New Item Reciept
View Reports and Summaries
Project Proposal
Generate Reports and Summaries

Appendix

- Meeting on 10/12/23
    - Interview, document analysis
    - Kira, Preston, Victor, Tran
    - Victor
- Meeting on 10/31/23
    - Interview/Initial planning and outline, brainstorming, and outcome analysis to decide on system requirements
    - Kira, Preston, Victor, Tran
    - Victor
- Meeting on 11/7/23
    - Interview/System development discussion and analyze current system technology for proper improvement
    - Kira, Preston, Victor, Tran
    - Victor
- Meeting on 11/28/23
    - Interview/Progress update follow up
    - Kira, Preston, Victor, Tran
    - Victor
- Meeting on 12/4/23
    - Interview/Finalize project
    - Kira, Preston, Victor, Tran
    - Victor