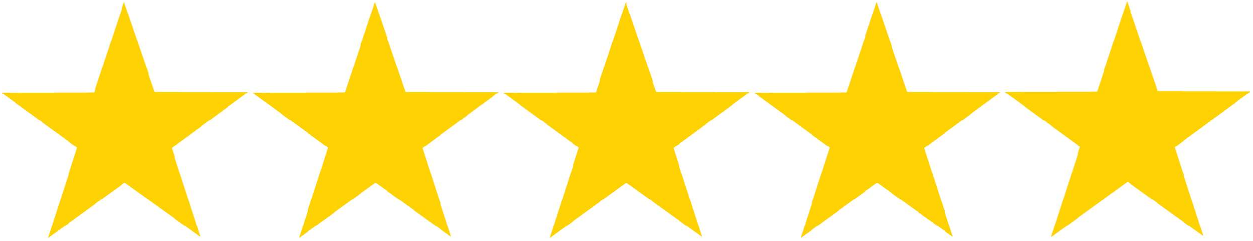


www.RateAClub.com

RAD Document

Brian Pigford, Max Cardillo, Joseph Hays, Matthew Kelley, Chad Lenz, Emmanuel Maku, Nick Myers, Austin Lane

Table of Contents

Cover Page 1

Table of Contents 2

Use Case Diagrams 4

Introduction 5

Proposed System 5

Functional Requirements 5

Non-functional Requirements 5

Glossary 6

Use Cases:

View Main Screen 7

Create Account 8

Login 10

Logout 11

View Account Management 12

Change Password 13

Change Email 15

Change School 16

View Club Directory 17

View School Directory 18

View Club Page 19

View School Page 21

Add Comment/Rating 22

Edit Comment/Rating 23

Delete Comment/Rating 24

Add Club 25

Edit Club Info 26

View FAQ 27

Search 28

Specific Error Messages 29

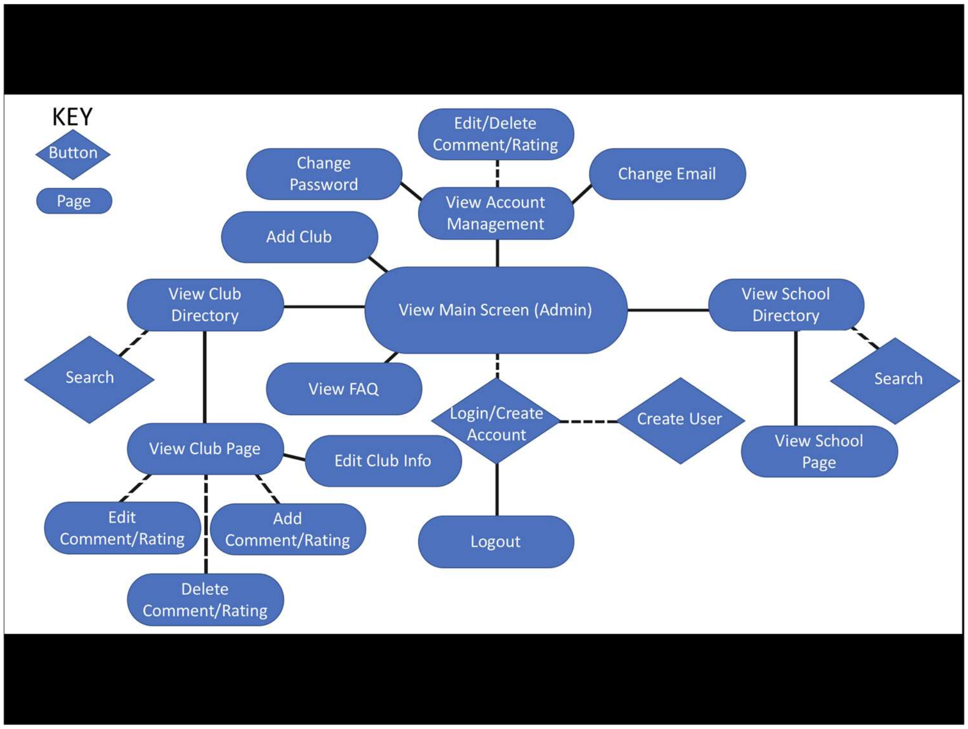
Admin Specific Error Messages 30

Gantt Chart 31

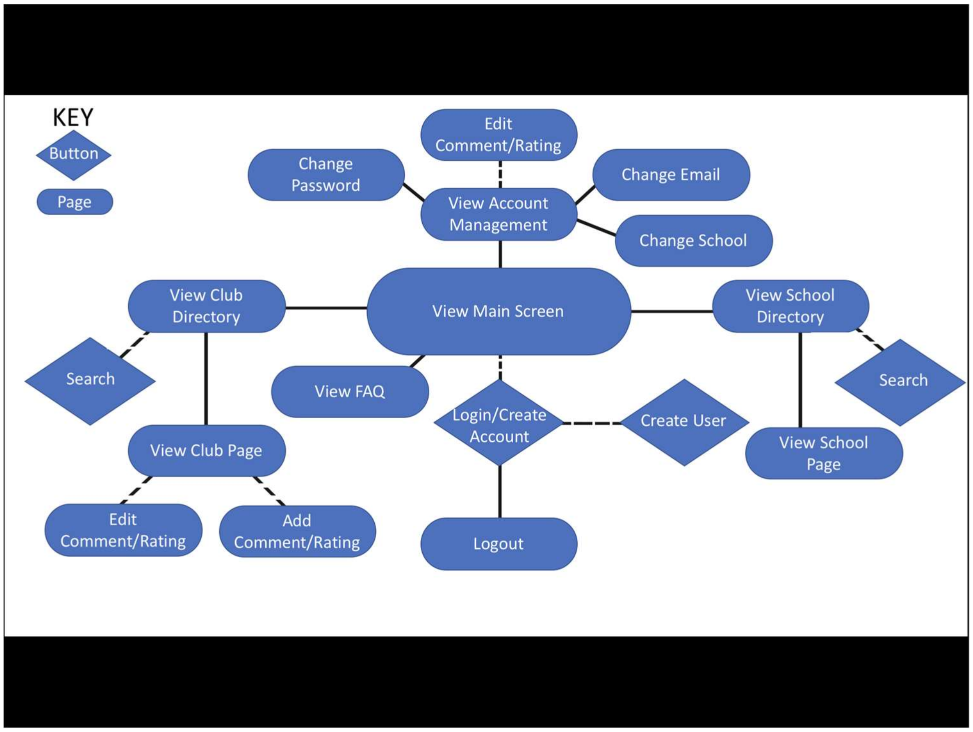
Database Design Tables 32

Use Case Diagrams

Admin User:



Regular User:



Introduction:

This documentation contains a description of the functional and nonfunctional requirements that our team is able to commit to our client in the website called RateAClub.org. We also present a potential design of the software product though a Use Case model and discuss our current progress developing the database and website.

Proposed System:

The scenario presented by the client consists of the following. The user goes to our website and has the option to create an account. They can view clubs in their school and read ratings and comments and view information on that club (club name, meeting time, club description, advisor name, and contact info) without having an account. If the user has an account, the user can view clubs they are involved in and leave ratings and comments. A list of every school and club will be available to the user and they can search for clubs and schools.

Functional Requirements:

The functional requirements we promise to our clients are listed below.

1. The website allows the user to view schools and clubs and their information and ratings/comments without an account.
2. The website allows the user to create an account.
3. The website allows the user to search for clubs and schools.
4. The website allows the user to leave ratings/comments.
5. The website allows the user to edit comments they made.
6. The website allows admins to delete accounts.
7. The website allows regular account holders to change schools.
8. The website allows the user to change their email.
9. The website allows the user to change their password.
10. The website allows the user to view frequently asked questions and their responses.

Non-functional Requirements:

1. User must create an account to leave a rating/ comment.

Glossary:

Account: a username, password, email address, and privileges associated with a user

Administrator (Admin): a user that has administrative privileges that include being able to add, edit, and delete clubs and the associated information in addition to the abilities of a standard user

Club Description: the section where miscellaneous information not covered by other fields is placed in the club entry

Comment: a section where a logged user may type information regarding their rating of a club, viewable after searching

Password: a String with at least 8 characters (one being in capital case and another being a special character and one being a number) used to authenticate and authorize actions

Rating: a integer between 1 and 5 used to describe the quality of a club, viewable after searching; also known as “stars”

User: a person using the application, which has the ability to search for clubs and to view ratings, comments, and suggestions

Username: a user-designated identifier associated with an account, used to authenticate login attempts and show which user left a particular rating

Use Cases:

Use Case: View Main Screen

Name of Use Case: *View Main Screen*

Primary Actor: User

Goal in Context: Create a landing page that a user will see when first accessing the website.

Preconditions: None

Trigger: Navigating to the site.

Scenario:

1. Website loads page consisting of header featuring a “Login/Create Account” button, the “Rate A Club” logo. If the user is logged in the “Login/Create Account” button changes to “Hi [username], Logout?” as well as an “Account Management” button in the center of the screen. There will also be “Club Directory” (invokes *View Club Directory* use case if user clicks button) and “School Directory” (invokes *View School Directory* use case if user clicks button) buttons. An “Add Club” button is shown if the user is logged in as an admin (invokes *Add Club* use case if user clicks button).
2. Website retrieves top 10 rated schools from the GoDaddy database then populates a table on the left of the page under the header and “Club Directory”, “School Directory” and “Add Club” (if user is an admin) buttons.
3. Website retrieves top 10 rated clubs from the GoDaddy database then populates a table on the right of the page under the header and “Club Directory”, “School Directory” and “Add Club” (if user is an admin) buttons.
4. Website loads a footer consisting of a “FAQ” link.
5. User clicks “Login/Create Account” button under the “Login/Create Account” drop-down menu (invokes *Login* use case and can invoke *Create Account* use case)
6. User clicks “Hi [username], Logout?” button if user is logged in (invokes *Log Out* use case).
7. User clicks “Account Management” button if user is logged in (invokes *Account Management* use case).

Exceptions: None

Use Case: Create Account

Name of Use Case: *Create Account*

Primary Actor: User

Goal in Context: Provides further capabilities within the website i.e. leave comments/ratings and enables "Account Management" button.

Preconditions: User is not logged in.

Trigger: The user selects create account from the header (see *View Main Screen* use case).

Scenario:

1. The user is taken to a page with seven textboxes with the labels (First Name, Last Name, Email, School (optional), Username, Password, and Confirm Password).
2. A line of text stating, "Password must be 8 characters long and contain a capital, a number and a special character." is shown under the Password textbox.
3. User enters all required information and clicks the "Submit" button.
4. Website checks that username and email do not already exist and that the passwords match and meet the set requirements (invokes *Change Password* use case).
5. Website redirects the user to the home page and logs the user in.

Exceptions:

1. Passwords do not match.
    a. Solution: Display error in black text in a red box under the "Submit" button "Password and Confirm Password do not match"
2. Password does not meet the special character requirement.
    a. Solution: Display error "No special character found in password".
3. Password does not meet the number requirement.
    a. Solution: Display error "No number found in password".
4. Password does not meet the length requirement.
    a. Solution: Display error "Password too short".
5. Password does not meet the capitalization requirement.
    a. Solution: Display error "No capital letter found in password".
6. Email already exists is the GoDaddy database.
    a. Solution: Display error "Email taken. Choose another".

7. First Name field is left blank.
    a. Solution: Display error “No first name entered”.
8. Last Name field is left blank.
    a. Solution: Display error “No last name entered”.
9. Email field is left blank.
    a. Solution: Display error “No Email Entered”.
10. Username field is left blank.
    a. Solution: Display error in black text in a red box under the “Submit Button” “No Username Entered”
11. Password and Confirm Password field are left blank.
    a. Solution: Display error “No password entered”.
12. Username already exists in the GoDaddy database.
    a. Solution: Display error “Username taken”.

Use Case: Login

Name of Use Case: *Login*

Primary Actor: User

Goal in Context: Provide a simple user interface for users and admins to log into their accounts.

Preconditions: The user must already have an account.

Trigger: Click on the "Login/Create Account" button in the Login drop-down menu (See *View Main Screen* use case).

Scenario:

1. Website displays two text boxes with the labels (Username and Password).
2. Website displays buttons labeled (Login, Create User).
3. User enters a correct username and password.
4. If user clicks "Login" button. Website validates credentials against GoDaddy database. Website logs user in and loads mainpage (invokes *View Main Screen* use case).
5. If user clicks "Create User" button (invokes *Create User* use case).

Exceptions:

1. Username is entered incorrectly.
    a. Solution: Display error "Invalid username entered".
2. Password is entered incorrectly.
    a. Solution: Display error "Invalid Password entered".
3. Username not entered.
    a. Solution: Display error "No username".
4. Password not entered.
    a. Solution: Display error "No password"

Use Case: Logout

Name of Use Case: *Logout*

Primary actor: Logged In User

Goal in Context: User may log out of their session while on the website.

Preconditions: Must have an account, and must be already logged in.

Trigger: The user clicks the “Hi [username], Logout?” button in on the top right of the screen. (see *View Main Screen* use case).

Scenario:

1. User clicks “Hi [username], Logout?” button. Website logs user out and loads main page (invokes *View Main Screen* use case).

Exceptions: None

Use Case: View Account Management

Name of Use Case: *View Account Management*

Primary Actor: Logged In User

Goal in Context: Allows the user to see their account information such as username, email, school. If the user is an admin, that is also shown. There are also options to change password, email, and school (if the user is not an admin). The user can also edit comments or delete them (if they are an admin).

Preconditions: The user has an account. The user must be logged in.

Trigger: The user clicks the "Account Management" button (see *View Main Screen* use case).

Scenario:

1. Website loads a page which shows their username, email address, school under the header. If the user is an admin it shows "You are an admin for your school." as well as the "Change Password" (invokes *Change Password* use case) and "Change Email" (invokes *Change Email* use case) links. A "Change School" (invokes *Change School* use case) link is shown if the user is not an admin. All of the user's comments are shown on the left side of the page with an "Edit" (invokes *Edit Comment/Rating* use case) button, and a "Delete" (invokes *Delete Comment/Rating* use case) button (if the user is an admin).
2. User clicks "Change Password" link (invokes *Change Password* use case).
3. User clicks "Change Email" link (invokes *Change Email* use case).
4. Non-admin user clicks "Change School" link (invokes *Change School* use case).
5. User clicks "Edit" button (invokes *Edit Comment/Rating* use case).
6. Admin user clicks "Delete" button (invokes *Delete Comment/Rating* use case).

Exceptions: None

Use Case: Change Password

Name of Use Case: *Change Password*

Primary Actor: Logged In User

Goal in Context: Allows the user to change their password.

Preconditions: The user has an account. The user must also be logged in.

Trigger: The user clicks the “Change Password” link (see *View Account Management* use case).

Scenario:

1. The user gets taken to a page with three text boxes with the labels “Current Password:”, “New Password” and “Confirm Password” and a “Change Password” button.
2. A line of text stating “Password must be 8 characters long and contain a capital, a number and a special character.” will appear under the “New Password” textbox.
3. The user enters their current password and their new password twice into the appropriate fields.
4. User clicks “Change Password” button. And the website verifies passwords are the same and satisfies the requirements and updates the password in the GoDaddy database.
5. There will then be a confirmation prompt that states “Password changed”. The website places the user back on the “Account Management” page (invokes *View Account Management* use case), the password is updated in the Godaddy database.

Exceptions:

1. Passwords do not match.
    a. Solution: Display error in black text in a red box under the “Change Password
    b. User” button “Password and Confirm Password do not match”
2. Password does not meet the special character requirement.
    a. Solution: Display error “No special character found in password”.
3. Password does not meet the number requirement.
    a. Solution: Display error “No number found in password”.
4. Password does not meet the length requirement.
    a. Solution: Display error “Password too short”.
5. Password does not meet the capitalization requirement.
    a. Solution: Display error “No capital letter found in password”.

6. Password & Confirm Password fields are left blank.
    a. Solution: Display error “No password entered”.

Use Case: Change Email

Name of Use Case: *Change Email*

Primary Actor: Logged In User

Goal in Context: Allows the user to change their email.

Preconditions: The user has an account. The user must also be logged in.

Trigger: The user clicks the "Change Email" link (see *View Account Management* use case).

Scenario:

1. The user gets taken to a page with that displays "Current Email:" and their email pulled from the database with one text box with the label "New Email" and a "Change Email" button below.
2. User clicks "Change Email" button. And the website updates the password in the GoDaddy database.
3. There will then be a confirmation prompt that states "Email changed". The website places the user back on the "Account Management" page (invokes *View Account Management* use case), the email is updated in the Godaddy database.

Exceptions:

1. New email field is left blank.
    a. Solution: Display error "No email entered".

Use Case: Change School

Name of Use Case: *Change School*

Primary Actor: Logged In Non-Admin User

Goal in Context: Allows the user to change their school.

Preconditions: The user has an account. The user must also be logged in.

Trigger: The user clicks the "Change School" link (see *View Account Management* use case).

Scenario:

1. The user gets taken to a page with that displays "Current School:" and their school pulled from the database with a drop down menu populated with all the schools in the GoDaddy database with the label "New School" and a "Change Schools" button below.
2. User clicks the drop down menu and selects a school.
3. There will then be a confirmation prompt that states "School changed successfully". The website places the user back on the "Account Management" page (invokes *View Account Management* use case), the school is updated in the Godaddy database.

Exceptions: None

Use Case: View Club Directory

Name of Use Case: *View Club Directory*

Primary Actor: User

Goal in Context: Display a list of all the clubs on the GoDaddy database and allow users to search for a specific club (invokes *Search* use case).

Preconditions: None

Trigger: User clicks "Club Directory" button (see *View Main Screen* use case).

Scenario:

1. The user gets taken to a page with a "Search Clubs" bar with a "Search" button beside it, centered under the header. Below the bar is a table labeled "ALL CLUBS" that is populated with every club listed with their school and overall rating pulled from the GoDaddy database.
2. User enters text into the "Search Clubs" bar and clicks "Search" button (invokes *Search* use case).
3. User clicks on a row in the "ALL CLUBS" table (invokes *View Club Page* use case).

Exceptions: None

Use Case: View School Directory

Name of Use Case: *View School Directory*

Primary Actor: User

Goal in Context: Display a list of all the schools on the GoDaddy database and allow users to search for a specific school (invokes *Search* use case).

Preconditions: None

Trigger: User clicks "School Directory" button (see *View Main Screen* use case).

Scenario:

1. The user gets taken to a page with a "Search Schools" bar with a "Search" button beside it, centered under the header. Below the bar is a table labeled "ALL SCHOOLS" that is populated with every school listed with their overall rating pulled from the GoDaddy database.
2. User enters text into the "Search Schools" bar and clicks "Search" button (invokes *Search* use case).
3. User clicks on a row in the "ALL SCHOOLS" table (invokes *View School Page* use case).

Exceptions: None

Use Case: View Club Page

Name of Use Case: *View Club Page*

Primary Actor: User

Goal in Context: View information about the club and view all comments/ratings and club information for that club.

Preconditions: None

Trigger: User clicks a club from the "ALL CLUBS" table on the "Club Directory" page or the school page (see *View School Page* use case). The user can also click a club on the "TOP 10 RATED CLUBS" table on the homepage (see *View Main Screen* use case).

Scenario:

1. The user gets taken to a page with a "Contact Info" box that shows "Email:", a box that shows the school's name which is linked to the school page (invokes *View School Page* use case) and the name of the club under the school name, and an "Overall Rating" box that displays the average rating displayed as a number of 1-5 stars for the club, pulled from the GoDaddy database, all displayed under the header.
2. All comments/ratings for the club are also pulled from the GoDaddy database and displayed on the left side of the screen. The "Rate This Club" button is shown above the comments. If the user has left a comment, an "Edit" button is shown (invokes *Edit Comment/Rating*). If the user is an admin and the club is part of their school, every comment has a "Delete" button underneath it (invokes *Delete Comment/Rating* use case).
3. The "Club Information" (Meeting Location, Meeting Time, Club Description, and Club Advisor) is also pulled from the GoDaddy database and shown on the right side of the page. If the user is an admin and the club is part of their school, a "Edit Club Info" button (see *Edit Club Info* use case) is displayed.
4. User clicks "Rate This Club" button (invokes *Add Comment/Rating* use case).
5. User clicks "Edit" button (invokes *Edit Comment/Rating* use case).
6. Admin user for school the club is associated with clicks "Delete" button (invokes *Delete Comment/Rating*
7. Admin user for school the club is associated with clicks "Edit Club Info" button (invokes *Edit Club Info* use case).

Exceptions:

1. The user is not logged in and tries to leave a comment/rating.
    a. Solution: Display error message stating, “You must be logged in to leave a rating!”.

Use Case: View School Page

Name of Use Case: *View School Page*

Primary Actor: User

Goal in Context: View information about the school and view all comments/ratings and all the clubs that school has.

Preconditions: None

Trigger: User clicks a school from the "ALL SCHOOLS" table on the "School Directory" page. The user can also click a club on the "TOP 10 RATED SCHOOLS" table on the homepage (see *View Main Screen* use case). The user can also click the school's name that is displayed as a link on the club page (see *View Club Page* use case).

Scenario:

1. The user get taken to a page that with a picture from the GoDaddy database, the school name, and an "Overall Rating" box that displays the average rating displayed as a number of 1-5 stars for the club, pulled from the GoDaddy database, all displayed under the header.
2. All the clubs for the school and their average ratings are pulled from the GoDaddy database and displayed in the "ALL CLUBS" table on the left side of the page.
3. The "School Information" (Number of Users, Number of Clubs, and Phone Number) is also pulled from the GoDaddy database and shown on the right side of the page.
4. User clicks a row on the "ALL CLUBS" table (invokes *View Club Page* use case).

Exceptions: None

Use Case: Add Comment/Rating

Name of Use Case: *Add Comment/Rating*

Primary Actor: Logged in User

Goal in Context: Allows the user to leave a comment/rating.

Preconditions: Haven't left previous rating/comment for that club.

Trigger: User clicks "Rate This Club" button (see *View Club Page* use case).

Scenario:

1. Website displays a box that has "Rating" stars that the user can select 1-5 , and a text box to leave a comment within. Underneath the text box is a "Create Rating" button.
2. User clicks "Create Rating" and their rating/comment or only rating is added into the GoDaddy database.

Exceptions:

1. If user enters a comment, but not a rating.
   a. Solution: Display error "You must leave a rating."

Use Case: Edit Comment/Rating

Name of Use Case: *Edit Comment/Rating*

Primary Actor: Logged in User

Goal in Context: Allow user to edit comments/ratings they have posted.

Preconditions: User has previously entered a comment/rating.

Trigger: User clicks edit button. (see *View Club Page* or the *View Account Management* use cases).

Scenario:

1. Website displays a box that has "Rating" stars that the user can select 1-5, and a text box that displays their previous comment and allows them to change it. Underneath the text box is a "Edit Rating" button.
2. User clicks "Edit Rating" and their rating/comment or rating only is added into the GoDaddy database.

Exceptions:

1. If user enters a comment, but not a rating.
    a. Solution: Display error "You must leave a rating."

Use Case: Delete Comment/Rating

Name of Use Case: *Delete Comment/Rating*

Primary Actor: Logged in Admin User

Goal in Context: Allows admins to delete comments/ratings for their school.

Preconditions: User has previously entered a comment/rating.

Trigger: User clicks delete button. (see *View Club Page* or the *View Account Management* use cases).

Scenario:

1. User clicks "Delete" and the rating/comment is deleted from the GoDaddy database.

Exceptions: None

Use Case: Add Club

Name of Use Case: *Add Club*

Primary Actor: Admin User

Goal in Context: Allow admins to add a club to the GoDaddy database

Preconditions: Logged in

Trigger: Admin clicks the Add Club button (see *View Main Screen* use case).

Scenario:

1. The user get taken to a page that displays their current school, along with six text boxes labeled (Club Name, Contact Email, Meeting Time, Meeting Location, Club Description, and Advisor Name). Underneath all of the boxes is a "Submit" button.
2. Admin enters all required information and clicks the "Submit" button. A confirmation message stating, "Club added" is displayed. The information is added to the GoDaddy database.

Exceptions:

1. Nothing is entered in the Club Name field.
    a. Solution: Display error in black text in a red box under the "Submit Button" "No Club Name entered".
2. Nothing is entered in the Contact Email field.
    a. Solution: Display error in black text in a red box under the "Submit Button" "No Contact Email entered".
3. Nothing is entered in the Club Description field.
    a. Solution: Display error in black text in a red box under the "Submit Button" "No Club Description entered".
4. Club Name name already exists in the GoDaddy database.
    a. Solution: Display error "Club name already taken"

Use Case: Edit Club Info

Name of Use Case: *Edit Club Info*

Primary Actor: Admin User

Goal in Context: Allow admins to change the information about a club that is a part of the school there are an admin for.

Preconditions: Club must have been created. The admin's school must be the school that the club is in.

Trigger: Admin clicks the "Edit Club Info" button (see *View Club Page* use case).

Scenario:

1. The admin gets taken to a page that displays five text boxes labeled (Contact Email, Meeting Time, Meeting Location, Club Description, and Advisor Name) that are populated with data pulled from the GoDaddy database.
2. Admin enters all required information and clicks the "Submit" button. The information is updated in the GoDaddy database.

Exceptions:

1. Nothing is entered in the Contact Email field.
    a. Solution: Display error in black text in a red box under the "Submit Button" "No Contact Email entered".
2. Nothing is entered in the Club Description field.
    a. Solution: Display error in black text in a red box under the "Submit Button" "No Club Description entered".

Use Case: View FAQ

Name of Use Case: *View FAQ*

Primary Actor: User

Goal in Context: Allow users to see FAQs.

Preconditions: User must be on any page of the website.

Trigger: User clicks the “FAQ” button (see *View Club Page* use case).

Scenario:

1. The user is taken to the“FAQ” page that displays a list of frequently asked questions and their answers.

Exceptions: None

Use Case: Search

Name of Use Case: *Search*

Primary Actor: User

Goal in Context: Allow users to search for a school or club.

Preconditions: User must be on either the "Club Directory" (see *View Club Directory* use case) or "School Directory" (see *View School Directory* use case) pages, referred to as "Either Page" for the remainder of this use case.

Trigger: User clicks the "Search" button (see *View Club Directory* or *View School Directory* use cases).

Scenario:

1. The user enters text into the "Search Clubs" or "Search Schools" bar.
2. User clicks the "Search" button.
3. A list of Clubs or Schools containing the searched text is populated in the table below the "Search Clubs" or "Search Schools" bar.

Exceptions:

1. The user searches for a club or school that is not in the GoDaddy database.
   a. Display "Search Not found." under the the ALL CLUBS or ALL SCHOOLS table.

Specific Error Messages

The user enters passwords that do not match when creating an account or changing password.

“Password and Confirm Password do not match” (see *Create Account, Change Password*)

The user enters a password without a special character when creating an account or changing password.

“No special character found in password” (see *Create Account, Change Password*)

The user enters a password without a number when creating an account or changing password.

“No number found in password” (see *Create Account, Change Password*)

The user enters a password that is too short when creating an account or changing password.

“Password too short” (see *Create Account, Change Password*)

The user enters a password without a capital letter when creating an account or changing password.

“No capital letter found in password” (see *Create Account, Change Password*)

The user tries to create an account with an email that is already used when creating an account.

“Email taken. Choose another” (see *Create Account*)

The user doesn’t enter a first name when creating an account.

“No first name entered” (see *Create Account*)

The user doesn’t enter a last name when creating an account.

“No last name entered” (see *Create Account*)

The user doesn’t enter an email when creating an account.

“No Email Entered” (see *Create Account*)

The user doesn’t enter an username when creating an account.

“No Username Entered” (see *Create Account*)

The user doesn’t enter a password and confirmation password when creating an account or changing password.

“No password entered” (see *Create Account, Change Password*)

The user tries to create an account with an username that is already used when creating an account.

“Username taken” (see *Create Account*)

The user enters their username incorrectly when logging in.

"Invalid username entered" (see *Login*)

The user enters their password incorrectly when logging in..

"Invalid Password entered" (see *Login*)

The user doesn't enter a username when logging in..

"No username" (see *Login*)

The user doesn't enter a password when logging in..

"No password" (see *Login*)

The user doesn't enter a new email when changing their email.

"No email entered" (see *Change Email*)

The user is not logged in and tries to leave a comment/rating.

"You must be logged in to leave a rating!" (see *View Club Page*)

The user enters a comment without leaving a rating.

"You must leave a rating." (see *Add Comment/Rating, Edit Comment/Rating*)

The user searches for something that returns no results.

"Search Not found" (see *Search*)

Admin Specific Error Messages

The admin doesn't enter a club name when creating a club.

"No Club Name entered" (see *Add Club*)

The admin doesn't enter a contact email when creating a club or changing club info.

"No Contact Email entered" (see *Add Club, Edit Club Info*)

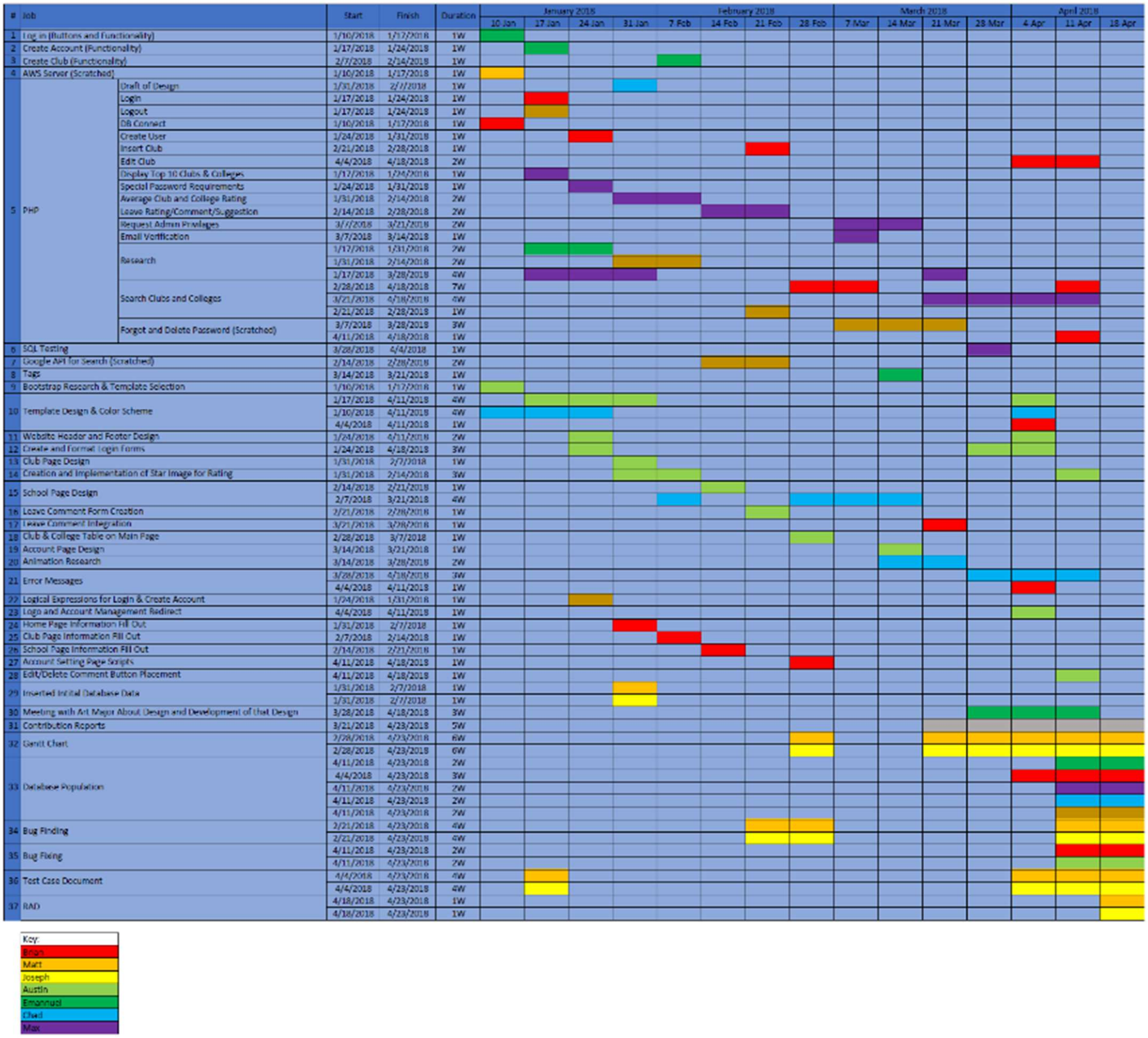
The admin doesn't enter a club description when creating a club or changing club info.

"No Club Description entered" (see *Add Club Edit Club Info*)

The admin tries to create a club with a name that already exists for their school.

"Club name already taken" (see *Add Club*)

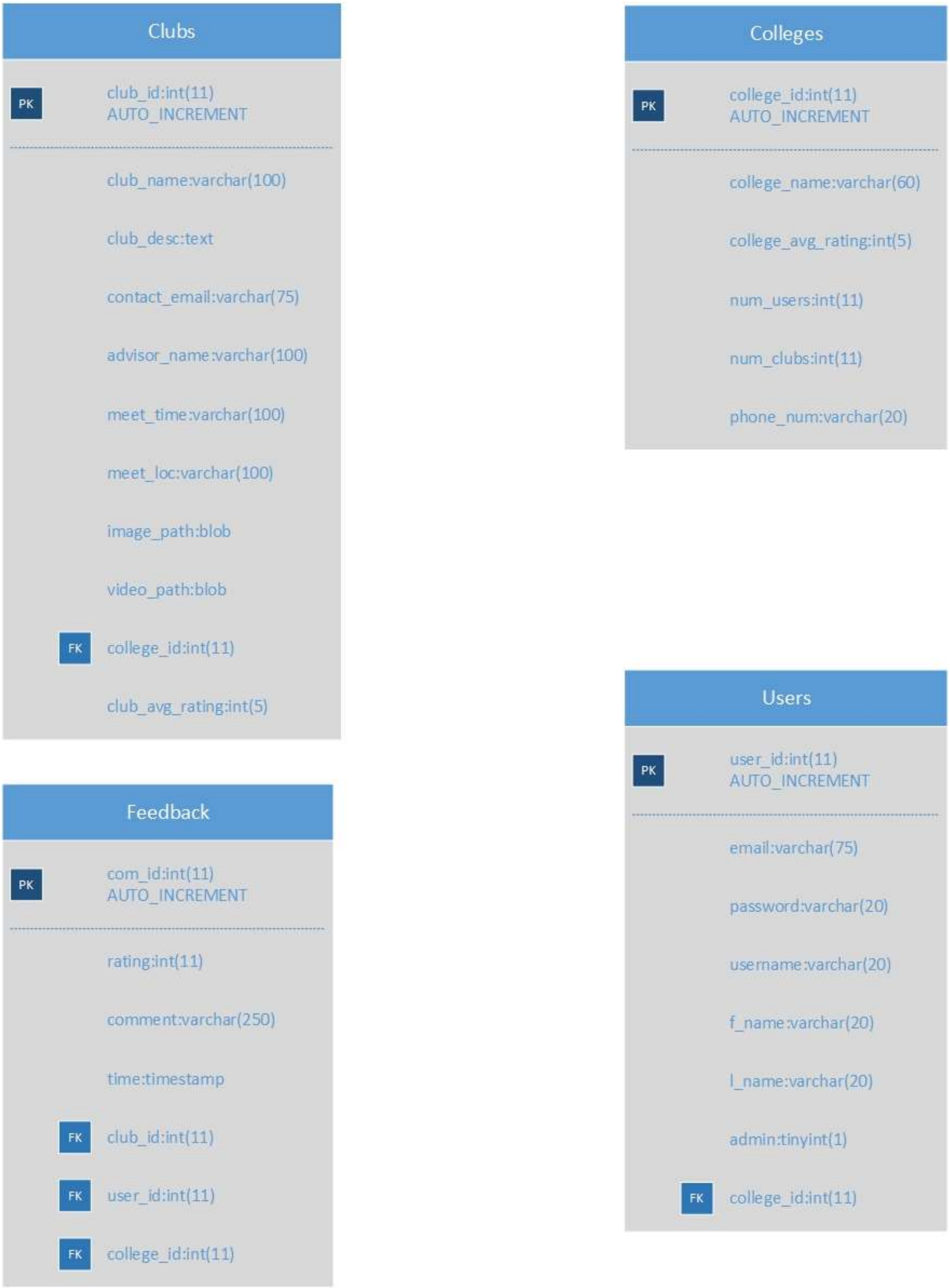
Gantt Chart



#	Job		Start	Finish	Duration	January 2018				February 2018				March 2018				April 2018		
						10 Jan	17 Jan	24 Jan	31 Jan	7 Feb	14 Feb	21 Feb	28 Feb	7 Mar	14 Mar	21 Mar	28 Mar	4 Apr	11 Apr	18 Apr
1	Log in (Buttons and Functionality)		1/10/2018	1/17/2018	1W	🟩														
2	Create Account (Functionality)		1/17/2018	1/24/2018	1W		🟩													
3	Create Club (Functionality)		2/7/2018	2/14/2018	1W					🟩										
4	AWS Server (Scratched)		1/10/2018	1/17/2018	1W	🟧														
5	PHP	Draft of Design	1/31/2018	2/7/2018	1W				🟦											
		Login	1/17/2018	1/24/2018	1W		🟥													
		Logout	1/17/2018	1/24/2018	1W		🟫													
		DB Connect	1/10/2018	1/17/2018	1W	🟥														
		Create User	1/24/2018	1/31/2018	1W			🟥												
		Insert Club	2/21/2018	2/28/2018	1W							🟥								
		Edit Club	4/4/2018	4/18/2018	2W													🟥	🟥	
		Display Top 10 Clubs & Colleges	1/17/2018	1/24/2018	1W		🟪													
		Special Password Requirements	1/24/2018	1/31/2018	1W			🟪												
		Average Club and College Rating	1/31/2018	2/14/2018	2W				🟪	🟪										
		Leave Rating/Comment/Suggestion	2/14/2018	2/28/2018	2W						🟪	🟪								
		Request Admin Privileges	3/7/2018	3/21/2018	2W									🟪	🟪					
		Email Verification	3/7/2018	3/14/2018	1W									🟪						
		Research	1/17/2018	1/31/2018	2W		🟩	🟩												
			1/31/2018	2/14/2018	2W				🟫	🟫										
		Search Clubs and Colleges	1/17/2018	3/28/2018	4W		🟪	🟪	🟪							🟪				
			2/28/2018	4/18/2018	7W								🟥	🟥					🟥	
			3/21/2018	4/18/2018	4W											🟪	🟪	🟪	🟪	
			2/21/2018	2/28/2018	1W							🟫								
		Forgot and Delete Password (Scratched)	3/7/2018	3/28/2018	3W									🟫	🟫	🟫				
			4/11/2018	4/18/2018	1W														🟥	
6	SQL Testing		3/28/2018	4/4/2018	1W												🟪			
7	Google API for Search (Scratched)		2/14/2018	2/28/2018	2W						🟫	🟫								
8	Tags		3/14/2018	3/21/2018	1W										🟩					
9	Bootstrap Research & Template Selection		1/10/2018	1/17/2018	1W	🟢														
10	Template Design & Color Scheme		1/17/2018	4/11/2018	4W		🟢	🟢	🟢									🟢		
			1/10/2018	4/11/2018	4W	🟦	🟦	🟦										🟦		
			4/4/2018	4/11/2018	1W													🟥		
11	Website Header and Footer Design		1/24/2018	4/11/2018	2W			🟢										🟢		
12	Create and Format Login Forms		1/24/2018	4/18/2018	3W			🟢									🟢	🟢		
13	Club Page Design		1/31/2018	2/7/2018	1W				🟢											
14	Creation and Implementation of Star Image for Rating		1/31/2018	2/14/2018	3W				🟢	🟢									🟢	
15	School Page Design		2/14/2018	2/21/2018	1W						🟢									
			2/7/2018	3/21/2018	4W					🟦			🟦	🟦	🟦					
16	Leave Comment Form Creation		2/21/2018	2/28/2018	1W							🟢								
17	Leave Comment Integration		3/21/2018	3/28/2018	1W											🟥				
18	Club & College Table on Main Page		2/28/2018	3/7/2018	1W								🟢							
19	Account Page Design		3/14/2018	3/21/2018	1W										🟢					
20	Animation Research		3/14/2018	3/28/2018	2W										🟦	🟦				
21	Error Messages		3/28/2018	4/18/2018	3W												🟦	🟦	🟦	
			4/4/2018	4/11/2018	1W													🟥		
22	Logical Expressions for Login & Create Account		1/24/2018	1/31/2018	1W			🟫												
23	Logo and Account Management Redirect		4/4/2018	4/11/2018	1W													🟢		
24	Home Page Information Fill Out		1/31/2018	2/7/2018	1W				🟥											
25	Club Page Information Fill Out		2/7/2018	2/14/2018	1W					🟥										
26	School Page Information Fill Out		2/14/2018	2/21/2018	1W						🟥									
27	Account Setting Page Scripts		4/11/2018	4/18/2018	1W								🟥							
28	Edit/Delete Comment Button Placement		4/11/2018	4/18/2018	1W														🟢	
29	Inserted Initial Database Data		1/31/2018	2/7/2018	1W				🟧											
			1/31/2018	2/7/2018	1W				🟨											
30	Meeting with Art Major About Design and Development of that Design		3/28/2018	4/18/2018	3W												🟩	🟩	🟩	
31	Contribution Reports		3/21/2018	4/23/2018	5W											⬜	⬜	⬜	⬜	⬜
32	Gantt Chart		2/28/2018	4/23/2018	6W								🟧			🟧	🟧	🟧	🟧	🟧
			2/28/2018	4/23/2018	6W								🟨			🟨	🟨	🟨	🟨	🟨
33	Database Population		4/11/2018	4/23/2018	2W														🟩	🟩
			4/4/2018	4/23/2018	3W													🟥	🟥	🟥
			4/11/2018	4/23/2018	2W														🟪	🟪
			4/11/2018	4/23/2018	2W														🟦	🟦
			4/11/2018	4/23/2018	2W														🟫	🟫
34	Bug Finding		2/21/2018	4/23/2018	4W							🟧	🟧						🟧	🟧
			2/21/2018	4/23/2018	4W							🟨	🟨						🟨	🟨
35	Bug Fixing		4/11/2018	4/23/2018	2W														🟥	🟥
			4/11/2018	4/23/2018	2W														🟢	🟢
36	Test Case Document		4/4/2018	4/23/2018	4W		🟧											🟧	🟧	🟧
			4/4/2018	4/23/2018	4W		🟨											🟨	🟨	🟨
37	RAD		4/18/2018	4/23/2018	1W															🟧
			4/18/2018	4/23/2018	1W															🟨

Key:
🟥 Brian
🟧 Matt
🟨 Joseph
🟢 Austin
🟩 Emmanuel
🟦 Chad
🟪 Max

Database Design Tables



Clubs

- PK club_id:int(11) AUTO_INCREMENT
- club_name:varchar(100)
- club_desc:text
- contact_email:varchar(75)
- advisor_name:varchar(100)
- meet_time:varchar(100)
- meet_loc:varchar(100)
- image_path:blob
- video_path:blob
- FK college_id:int(11)
- club_avg_rating:int(5)

Colleges

- PK college_id:int(11) AUTO_INCREMENT
- college_name:varchar(60)
- college_avg_rating:int(5)
- num_users:int(11)
- num_clubs:int(11)
- phone_num:varchar(20)

Feedback

- PK com_id:int(11) AUTO_INCREMENT
- rating:int(11)
- comment:varchar(250)
- time:timestamp
- FK club_id:int(11)
- FK user_id:int(11)
- FK college_id:int(11)

Users

- PK user_id:int(11) AUTO_INCREMENT
- email:varchar(75)
- password:varchar(20)
- username:varchar(20)
- f_name:varchar(20)
- l_name:varchar(20)
- admin:tinyint(1)
- FK college_id:int(11)