SYSTEM REQUIREMENTS SPECIFICATION

VERSION 1.0

JUST GO BOULDER

For: University of Colorado Boulder

Just Go Boulder July 6 2018

Yasmin Alshafai
Kyle Gronberg
Kyle Ward
Ali Oustlund
Gouri Yerra

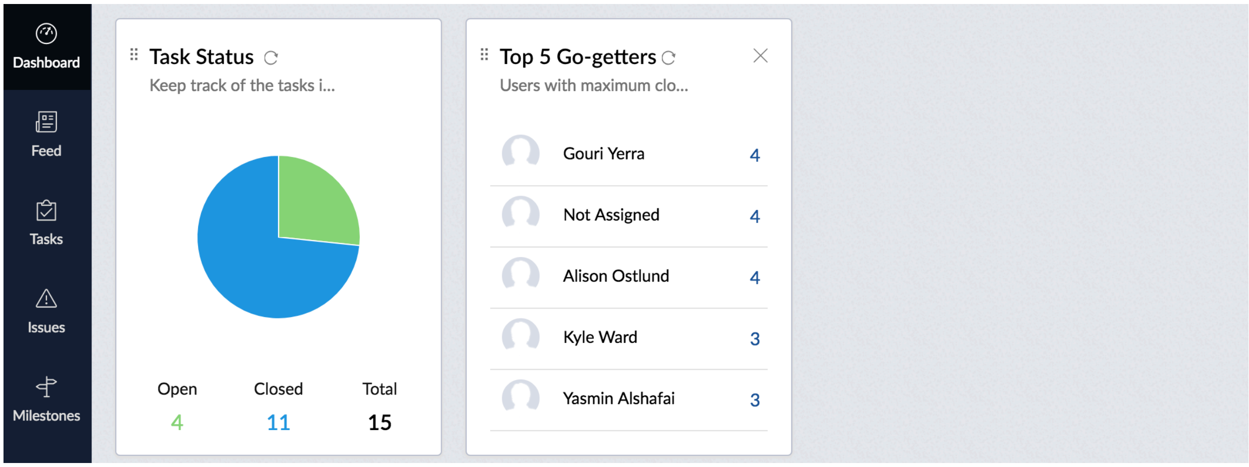
Contents

Introduction

Our system provides a look into Boulder’s activities from locals’ perspective. To give visitors and Colorado natives a look into Boulder’s nature, restaurants, shopping, and fun events. This report outlines the project’s background, goals and objectives, stakeholders, current problems and situation, use case diagrams and analysis, testing descriptions, and Entity Relational Diagram with analysis. Our GitHub repository can be found at: **https://github.com/YasminAlshafai/JustGo-Boulder**

Project Tracker

We used the Zoho website as a project tracker. This website offers features that make assigning tasks and projects to each team member easy. We used two main applications offered through Zoho, the Project application, and the Sprint application. Within the Project application, we were able to keep track of tasks and milestones. The Project feature allowed us to split the tasks up based on the frontend and backend. This way the Backend had their own set of tasks specific of what was needed for their part and the Frontend had tasks assigned specifically to their side of the webpage. This feature made it easy and organized to view each task and what needs to be done, but also check the status of each end of the project. For each task, we were able to add descriptions, timelines, status, completion(in percentage), and priority. This allowed a full overview of what has been done and what needs to be completed. This project tracker feature allowed each group member to get a clear overview of what is to be completed and what is expected from each member of the time. The sprint feature was very similar to the Project feature except it gave the group a closer look at the tasks due. The sprints we used were each a week long so there was a new sprint and set of tasks each week. This kept the group on track and kept us organized with what needs to be done. This also contributed to the organization in the Project feature and the tasks for both frontend and backend. Through Zoho and the features it provided, our group was able to stay on track and keep on the same page throughout the extent of this project. This tool contributed largely to the competition and success of this Project and team. The link to our project management tool is: https://projects.zoho.com/portal/justgoboulder#dashboard/1282423000000020308



Task Status

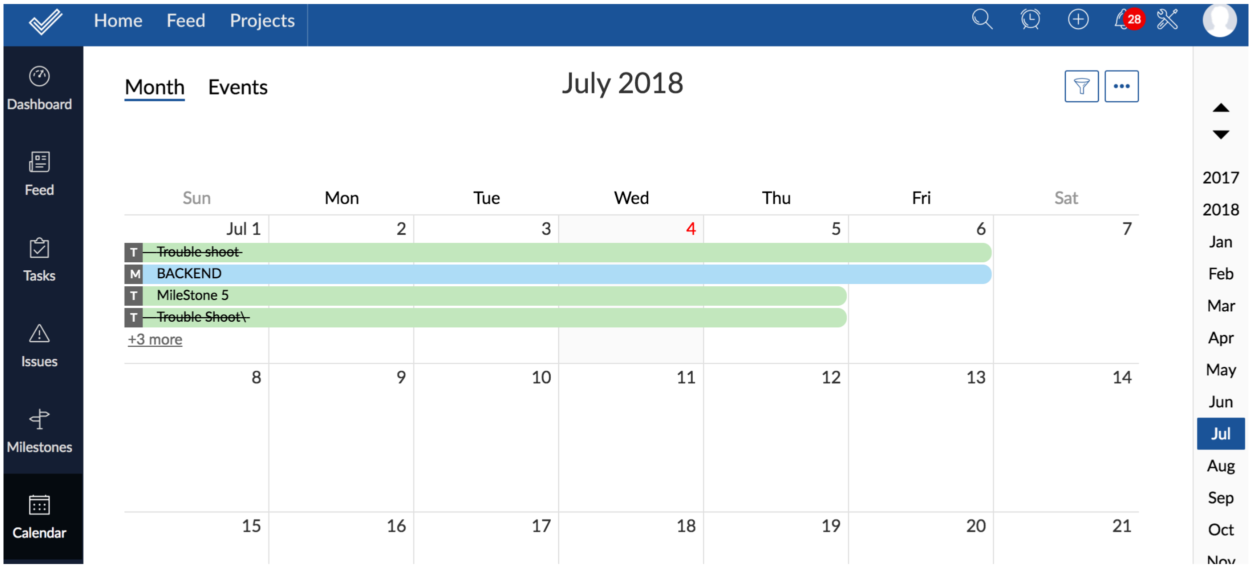
Keep track of the tasks i...

Open	Closed	Total
4	11	15

Top 5 Go-getters

Users with maximum clo...

Gouri Yerra	4
Not Assigned	4
Alison Ostlund	4
Kyle Ward	3
Yasmin Alshafai	3



July 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
Jul 1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21

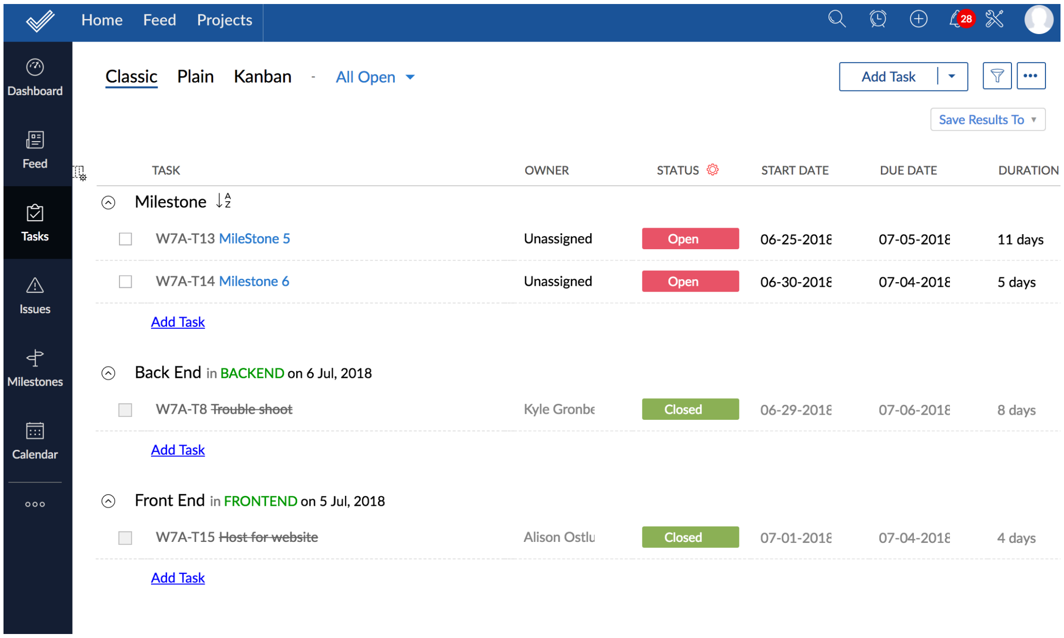
T ~~Trouble shoot~~

M BACKEND

T MileStone 5

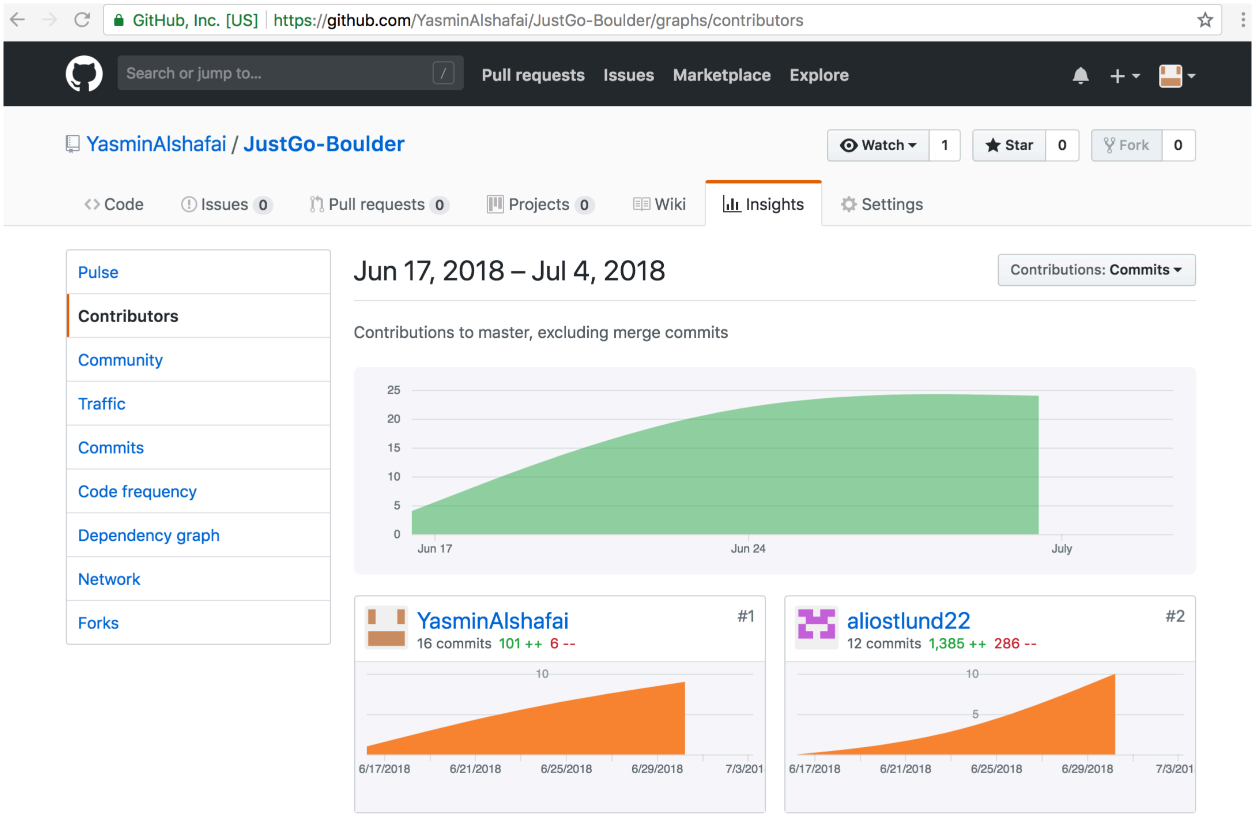
T ~~Trouble Shoot\~~

+3 more



TASK	OWNER	STATUS	START DATE	DUE DATE	DURATION
Milestone					
□ W7A-T13 MileStone 5	Unassigned	Open	06-25-2018	07-05-2018	11 days
□ W7A-T14 Milestone 6	Unassigned	Open	06-30-2018	07-04-2018	5 days
Add Task					
Back End in BACKEND on 6 Jul, 2018					
□ W7A-T8 ~~Trouble shoot~~	Kyle Gronbe	Closed	06-29-2018	07-06-2018	8 days
Add Task					
Front End in FRONTEND on 5 Jul, 2018					
□ W7A-T15 ~~Host for website~~	Alison Ostlu	Closed	07-01-2018	07-04-2018	4 days
Add Task					

GitHub Commits per Team member

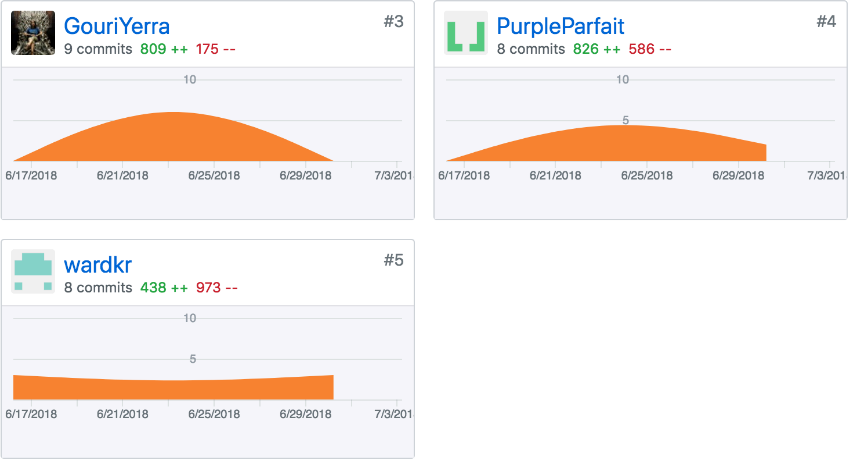


Jun 17, 2018 – Jul 4, 2018

Contributions to master, excluding merge commits

YasminAlshafai #1
16 commits 101 ++ 6 --

aliostlund22 #2
12 commits 1,385 ++ 286 --



Stakeholders

Our site might attract interest of new students coming in Boulder who want to study in CU boulder, travel agencies, and locals who are interested in knowing about live shows, promo codes for restaurants, or getting direction for a hiking trail. And eventually when we expand our website nation wide, tourists might be interested in knowing about different places and events about their desired destination.

Current Situation and Problems/Opportunities

The project's current situation is a working HTML and SQL database, however, the backend is not connected to the frontend. Some modifications with CRUD capabilities for the admin page, to add the create ability. We are still working on implementing all of the features that we would like the website to have. Our plan for the future is to create the subscription button to create a mailing list, customize the emails that the user will receive, create a comment box for each attraction, implement a bigger directory of events and attractions, and last add a list of other cities from other cities because we would like this to go nation wide.

Sprint Meetings Documentation

At the end of week two we created three weekly sprints. We assigned clear tasks so that Gouri and Ali were responsible of front end, Kyle G. and Yasmin were responsible of creating the PHP and SQL code and Kyle W. was responsible of delivering supporting data for the website and populating the database. We had meeting with the entire group and other meeting with just two or three members. The first meeting consisted of us creating templates using Wireframe which we successfully created and we updated the project management tool.

During the second sprint Yasmin and Kyle G. researched PHP to create the login feature and link our database with the front end. Gouri and Ali were able to have the HTML code ready and Kyle W. took the responsibility of plugging in the data he accumulated about the attractions.

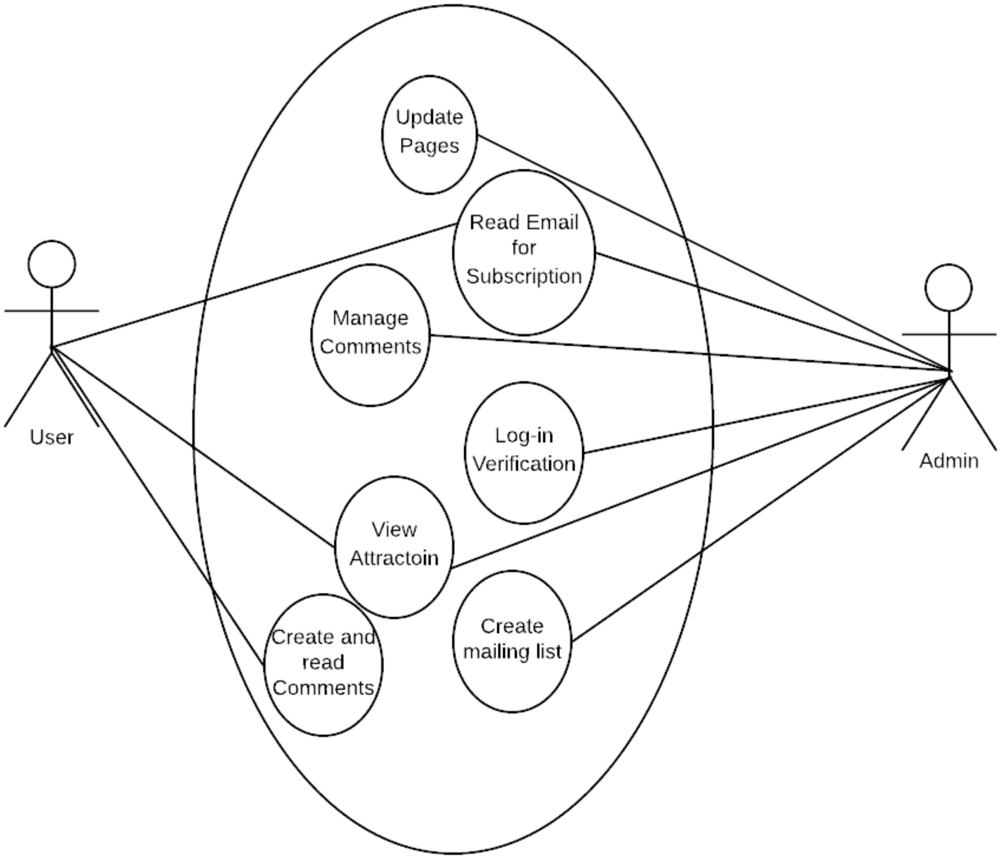
During the last sprint Kyle W. and Kyle G. were responsible of creating the admin page, in which admins are able to manage the database. Ali and Gouri finished up the tabs and added some finishing touched such as the weather, direction and social media links in the attractions. Yasmin was responsible of documentation and finishing the final project report.

Deployment

We decided to use the local host from XXAMP/Apache to host our website, due to budget constraints XXAMP was the best choice. The local host choice also allowed us to run multiple programming languages in parallel.

For now, all of our files are on Github and users will be able to download all of the files and run the web page. In the future, we are planning to create a domain name and host the website so that users are able to go to the webpage directly.

Use Case Diagram



Actors	
Actor Name	**Definition / Description**
User	Generic user who is able to navigate the website, subscribe to the mailing list, receive emails, and post comments. In the future we want to implement a feature where users are

Admin	Administrators are those who manage the front and back end of the website and they are given extra privileges to manage features of this software.

Use case Description			
Use Case ID	**Priority**	**Use Case Name**	**Description**
UC1	High	Update Pages	Admins are able to update the attraction tab, to change the addresses, pictures, descriptions related to attractions
UC2	High	Read emails from subscription list	Users and admins are able to read emails the mailing list associated with each attraction.
UC3	Medium	Manage comments	Admin are able to Create, Read, Update, and Delete comments associated with each attraction page
UC4	High	Log-in Verification	Admins are able to log in to a separate page in which they are able to manage the database, to add attractions, change picture, addresses, and delete some attractions if needed.
UC5	High	View attractions	Admins and users are able to navigate and view the attractions with all details associated with them such as directions, addresses, and a brief description.
UC6	High	Create mailing list	Admins are able to create the mailing list and the actual email to be sent users.
UC7	Medium	Create and read comments	Users are only able to write and read the comments they post on each page

Testing Description

User Acceptance test 1:

System Test Case:
JustGo Boulder
User Story U11: Admins login to make edits

Purpose: Verify the user story U11 Instructions:	
Test Run Information: Tester Name: Yasmin Alshafai Date(s) of Test: 06-28-2018 Location/server being used: Local Host	**Prerequisites for this test:** Have to be an Admin with assigned username and password
	Software Versions: **Application:** **Browser used:** Google Chrome **Version 67.0.3396.99** **Database:** JustGo Boulder **Operating System**: macOS High Sierra
	Required Configuration:
Notes and Results: The php code was tested using a local host, I used XAMPP for a local Host and put my testing files in the htdocs file. We still need to create usernames and passwords for all of the group members and modify the style.css file.	

Test Script Steps/Results					
Step	Test Step/Input	Expected Results	Actual Results	Requirements Validated	Pass/Fail
Admin feature to login to be able to edit pages and comments					
1.	Navigate to the Login tab	Login tab appears on far right			
2.	Press on Login tab	User directed to a login page			
3.	Enter Assigned Username	Able to enter text			
4.	Enter Assigned Password	Able to enter text			
5.	Click on login	Directed to a webpage where they are able edit the data.			

Alternative Flow 1: User is not an Admin					
1.	User will plug in a username and password	An error message pops to explain that the user is not admin and that they are not allowed to			

		login			
2.					
Alternative Flow2: Admin forgot username or password					
1.	User might insert wrong username or password	Error message will pop explaining that the username/password does not match what we have in the system and it will prompt the user to send an email to other admins to help update their login information			

User Acceptance Test 2:

Gouri Yerra
Alison Ostlund
System Test Case:

Purpose: Verify the User story for FrontEnd Purposes
User Story: End User Wants to: Find areas with restaurants Get directions to attractions

Step	Test Step/Input	Expected Results	Actual Results
1	Enter keyword to search bar	Able to enter text	
2.	Press the search button	Multiple results returned and all instances of keyword appear	
3.	Click on a attraction on Navigation Bar	Information about that Boulder attraction will pop up including its history, images, and more details.	
4.	Look at slideshow / click on next arrow	After a set time frame the images on slide show will automatically rotate/ click on next arrow and next image in set will appear	

5.	Click on “get directions” to find how to get to each attraction	Will pull up a google maps navigator where you can enter your information	
6.	Press Enter	Google Maps will show you the fastest way to get to that location from your entered location. Directions in text are also displayed	

Alternative Paths:

Step	Test Step/Input	Alternative Path	Actual Results
1	Enter keyword to search bar		
2.	Press the search button	Keyword does not exist on webpage/ error message will show up saying search doesn't exist	
3.	Click on a attraction on Navigation Bar	The page is broken so an error page will appear instead of the attraction page	
4.	Look at slideshow / click on next arrow	Will be set to one still picture in the slide show with no alternative pictures.	

5.	Click on “get directions” to find how to get to each attraction		
6.	Press Enter	The google maps page is broken so the link will be broken, will direct user to an error message that will inform them the link is broken.	

Automated Tests:

For The Front End portion, our main tools for automated tests will be User Inputs, such as keyboard actions and mouse clicks. This actions will show whether the correct output or expected output is produced. For example, if a tab on the navigation bar is clicked and the user isn't redirected to the desired page, the test case fails.

User Acceptance Test 3:

JustGo Boulder

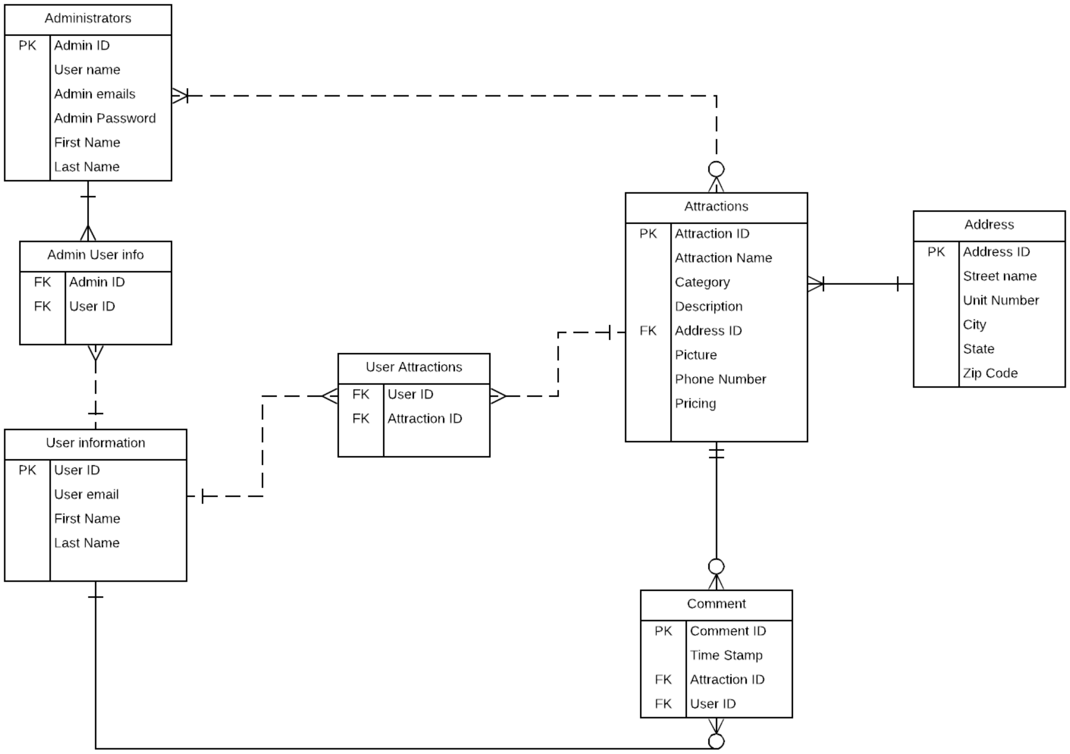
User story U7: Subscribing to an attraction

Purpose: Verify the user story U7 (all parts)	
Test Run Information: Tester Name: Kyle Ward Date(s) of Test: Location/server used: Local host	**Prerequisites for this test:** none **Application:** **Database:** JustGo Boulder **Browser:** Google Chrome Version 67.0.3396.99 **Operating System:** Windows 10 **Required Configuration:** (browser setup, security, ID roles) No special setup needed
Notes and Results:	

Test Script Steps/Results					
step	**Test step/input**	**Expected results**	**Actual results**	**Req val**	**pass /fail**
Subscribe to a specific attraction present in the database					
1	Type in url	Navigate to home page			
2	Click on category buttons	Drop down menu appears with all attractions in that category			
3	Select attraction	Navigate to that specific attraction page			
4	Enter in user info into subscription form	Form displays input			
5	Click submit button	Form info is sent to the database and a confirmation email is received by the user			
6	Wait for email	Receive emails with info about			

		upcoming events and specials regarding that specific attraction			
7	If user has multiple subscriptions	The user will receive one compiled email containing all events and specials for every subscribed attraction			
Alternative Flow 1: Form not filled out correctly					
	Incorrect email	Display error message “Please Enter A Valid Email” and return to enter information into form			
Alternative Flow 2: Form does not display correctly					
	Error displaying form	Display an error message saying “Site is under maintenance please come back soon”			
Alternative Flow 3: Attraction pictures do not display					
	Error displaying an attraction photo	Display an error message saying “”Image not Found”			

Entity Relational Diagram



Class	Class Description
Administrators	An entity to store information the administrators of the website
Admin User Info	An entity to store information about the users
User Information	An associative entity to connect administrators and users. Due to security reasons users will only be able to see the admins’ emails.
Attraction	Entity to store everything related to attractions
User Attraction	An associative entity to connect users to attractions
Comment	An Entity to store info related to comments
Addresses	Addresses were a multi valued attribute, for this reason, we created a separate table for it.

Class	Attribute	Attribute Description
Administrators		
	Admin ID	The primary key for this table, all other attributes depend on this key
	Admin Username	User name assigned to admins to login
	Admin Emails	Admins emails for users to contact them
	Admin Password	Passwords assigned to admins to login
	First Name	Admins first name
	Last Name	Admin last name
Admin User Info		
	Admin ID	A foreign key to reference the admins table
	User ID	A foreign key to reference the users table
User Information		
	User ID	The primary key for this table, all other attributes depend on this key
	User email	Users email for the subscription list
	First name	Users first name
	Last name	Users last name
User attraction		
	User ID	A foreign key to reference the users table
	Attraction ID	A foreign key to reference the attraction table
Attractions		
	Attraction ID	The primary key for this table, all other attributes depend on this key
	Attraction name	The name of the attraction
	Category	The category to help us with the drop down menus
	Description	A brief description for each attraction

	Address ID	A foreign key to reference the address table
	picture	A link for the pictures
	Phone number	Phone number to contact the attraction
	Pricing	We want to create a pricing schema to show users how expensive the attraction is
comments		
	Comment ID	The primary key for this table, all other attributes depend on this key
	Time stamp	To show the date, time, and users on each attraction
	Attraction ID	A foreign key to reference the attraction table
	User ID	A foreign key to reference the users table
Address		
	Address ID	The primary key for this table, all other attributes depend on this key, all other attributes are just parts of the address.

Supplementary Specification

Design Requirements & Constraints		
Requirement / Constraint Name	Priority	Requirement/Constraint Descriptions
Including all information in the data base	High	This will allow us to be efficient in which we only have one place to go to to update everything, instead of hardcoding attraction using HTML.

Security Requirements & Constraints		
Requirement / Constraint Name	Priority	Requirement/Constraint Descriptions
User authentication	High	Protecting user’s account information, the system must be able to identify a user by combination of username and password.

Implementation Requirements & Constraints		
Requirement / Constraint Name	Priority	Requirement/Constraint Descriptions
Sharing and modifying code	High	We used GitHub to be able to work together and make sure to test before pushing commits.
Deciding on programming languages	Medium	We had to agree which languages would serve us best on presenting what we want to implement, we also put timing into perspective.

Interface Requirements & Constraints		
Requirement / Constraint Name	Priority	Requirement/Constraint Descriptions
Simplified user interface	Medium	Users will able to find everything related to one attraction in on e page, they will also be able to navigate from one attraction to another very easily.
Mobile friendly interface	Medium	One requirement for the future is to use bootstrap tools to make our website mobile friendly so users are able to view our contents from any device.

Conclusion

In the end, this project was a great opportunity to get a hands on experience of different software tools out there. It gave us experience in working as a team and assigning different tasks and also helping each other when we face problems.

The website is a minimum viable product at the moment, but we hope to work on it and produce a better interactive platform. We hope that this website would serve and help people to find all activities in one stop.