OCTOBER 30, 2019

Software Engineering

Assignment-IV

Group II

Project Plan Template

By

Jaya Krishna Mandivarapu

Kiruthiga Sekar

Nicholas Solmon Yesu

Ke Wang

Project Name: Event Me

Semester: Fall 2019

Course Name: CSC6350 Software Engineering

Group Members: Jaya Krishna Mandivarapu, Kiruthiga Sekar, Nicholas Yesu, Ke Wang

Group: Group II

Document Revision #: 1.0.0

Date of Submission: October 30, 2019

Table of Contents

1. Planning and Scheduling

Name	Email	Task	Duration	Dependency	Due Date	Note
Jaya Krishna Mandivarapu	Jmandivarapu1@student.gsu.edu	• Modified the Database Architecture • Created the Server in AWS • Exposed the API Gateway to connect to the front end • Written sample data to the database.	3 hrs.	• Nodejs • HTML/CSS • Mongodb • AWS • AWS Lamda • API Gateway	October 30, 2019	100%
Kiruthiga Sekar	ksekar2@student.gsu.edu	• Team Co-Ordinator • Created Login, SignUp Page and provided screenshots of it • Architectural View • Redrew Class Diagram • Edited Sequence Diagram and Use Case Diagram • Report Formatting	4hrs.	• MS Visio • HTML/CSS • Sublime Text	October 30, 2019	100%
Nicholas Yesu	nyesu1@student.gsu.edu	• Creation of HTML/CSS • demo pages: Create New Event, Gallery, Home Page • Created Logo, • Provided Screenshots • Upload Files to GitHub	4 hrs.	• Sublime Text • HTML/CSS • PHP • Abode Photoshop • Illustrator	October 30, 2019	100%
Ke Wang	kwang15@student.gsu.edu	• Testing • Test cases	4 hrs.	• JavaScript • Jasmine unit test environment	October 30, 2019	100%

2. Problem Statement

Our product is a completely online event planner called Event Me with versatile features that allow the user to individualize their event and not worry about rest of additional things needed for the event. User or Client can just create event with us, and we will take care of the rest of the things like decoration, food etc. needed for the event. Our product not only saves time for the clients but also will be helpful in getting the whole event package at a lower price. Functionalities for the overview of the project will first have to include our landing page that will house our different services and pages for the user, the second mandatory page will be the payment option page. This page will require the user to input a payment option or third-party payment like PayPal or Google Pay so they can secure their events on schedule. The next mandatory page will be the main function of the website which will be the create event page, this page will allow the user to create a new event and allow them to select different assets from the resources we offer, that includes type of event, assets, location, and date. Subfunctions of the event page will include edit event and delete event. The create new event/ edit existing event requires the user to make an account and sign in. The next page is our services page, this page does not require the user to sign in because the function of this page is just so the user can look and see what types of events we offer, the types of assets we offer, and so on. This gives the user an idea of what we can offer them without committing to making an account and starting the process. The next page is the gallery or (history), the functionality of this page like services is so that users can view the past events created and brought to life by EventMe.com, this pages main purpose is to create a trust and confidence with the user so they can feel more comfortable with the website before they start their first event with EventMe. As an overview we're going to be focusing on the database API and the back end so that we can hold all the user data securely and create a fluid design on the front end so the process can take under 30min to create an event fast and easy. We're planning on using HTML/CSS, JavaScript, react, (possibly MySQL), java, and Adobe Suite for UI/UX as the main languages to create and run this platform.

This product is meant for anyone who organizes events individually or professionally ranging from lower price level to higher price level. We also target customers who want to hassle-free events with low budget and simplicity. This can also include businesses who want to organize events at their offices or schools who'd like to plan an event for their students. EventMe services organizations as well as Individuals to allow them to create the event to their choosing and tailor an event to everyone.

The functionalities of this website provide a way for us to implement a fast and easy way to plan an event. The payment page will take a down payment of the event to secure your details like date and location so that you don't have to worry about scheduling after you've created your event. The Create and Event page will solve the main issue of there being so much time and planning going into an event and transforms it into a page with multiple options that allow the user to customize their event in many different ways and potentially can take up to 20-30min which is exponentially faster than going about it on your own or hiring a event planner yourself. The gallery and services pages help create a confidence with the user which solves the issue of trustworthiness of the website also gives a realistic and honest example of what the user can accomplish by using EventMe.com. This page also allows us to process our data quicker in chunks and can expedite the process. The main problem that EventMe looks to solve is the long and complicated process of planning an event, with EventMe we plan to expedite the process and make it easier for the user/client so that they don't have to manually plan the event themselves or hire an expensive event planner. EventMe streamlines and simplifies the event

planning process to optimize efficiency. (Saves time and money). Apart from solving the existing the event planning process it also creates a unique opportunity to combine the distributed field of individual things in event planning like decoration, food etc to come together and to form a unified model in the field of Event planning.

As far as a completely online solution, there is none (currently) other alternatives include professional event planners (expensive and tedious).

The solution we created allows the user to organize a quality event in under an hour; this saves the user time, money, physical trips, etc. It also opens the gateway to field of new collaborations between different departments and event will be responsible for bringing down the prices of the event planning. As the current event planning in US costs a lot of money ranging from tens of thousands to millions. Our product will have direct impact on thousands of users.

Our target customers range from 18yr-60yr old individuals that are looking for a simpler solution to event planning. Our Top-Level Objectives include creating a database and API on the back end and design intuitive UI for the pages on the front end. Our Differentiators include easy access to efficient event planning software, ability to customize details quickly and on the go, time and cost efficiency for professionals, etc. The scope of our project includes our top-level objectives as well as giving the user an easy to use interface and ability to customize their events after making it.

Our competitors are general planning software (not as big) and professional event planners. Make it clear that the system can be built, making good use of the available resources and Technology. The bulk of our front end and the back end would be HTML/CSS, Nodejs and JavaScript as well as React for database/API reasons. The online event planner system is definitely in the realm of execution based on the languages mentioned above. With these languages (HTML/CSS and JavaScript) we can create a fluid design on the front end that rivals our competition websites (not many) and makes it simpler than hiring a professional event planner who usually only offers only specific event types per planner. Our initial drafts for our pages can be built with UI/UX software before we develop it just so we can nail in our design first. The development of the front end will mainly come from the HTML/CSS & JavaScript components of our execution to allow us mimic what we created for the UI. The React/Nodejs (and possibly MySQL) will serve as our main asset to creating and maintaining our database to hold user information in reference to our novel approach. (how we will approach this project).

This is interesting from a technical viewpoint because of the relationship we will build from front end to back end with the array of languages we plan to use so that we can achieve a seamless user experience. We will be using up to the mark front-end languages, which is technically robust and secure. For building the backend and API we will using AWS (Amazon Web Services) which will provide us will in build security and all the backend services will be written in Nodejs. All the data will be stored in the Cloud using NoSQL database Mongo db and our database will have auto backup options which is needed in case of any malfunctions.
(Please refer to our Technical Architecture Section for Further Information)

3. Class Diagram

The class diagram shows the object classes, their attributes and their operations in a system and the association between the classes. The class diagram for our project is shown below. A better picture can be obtained from the vsdx file in GitHub.

GitHub link: https://github.com/KiruthigaSekar/Software-Engineering/blob/master/Sprint%203/ClassDiagram1_KiruthigaSekar.vsdx

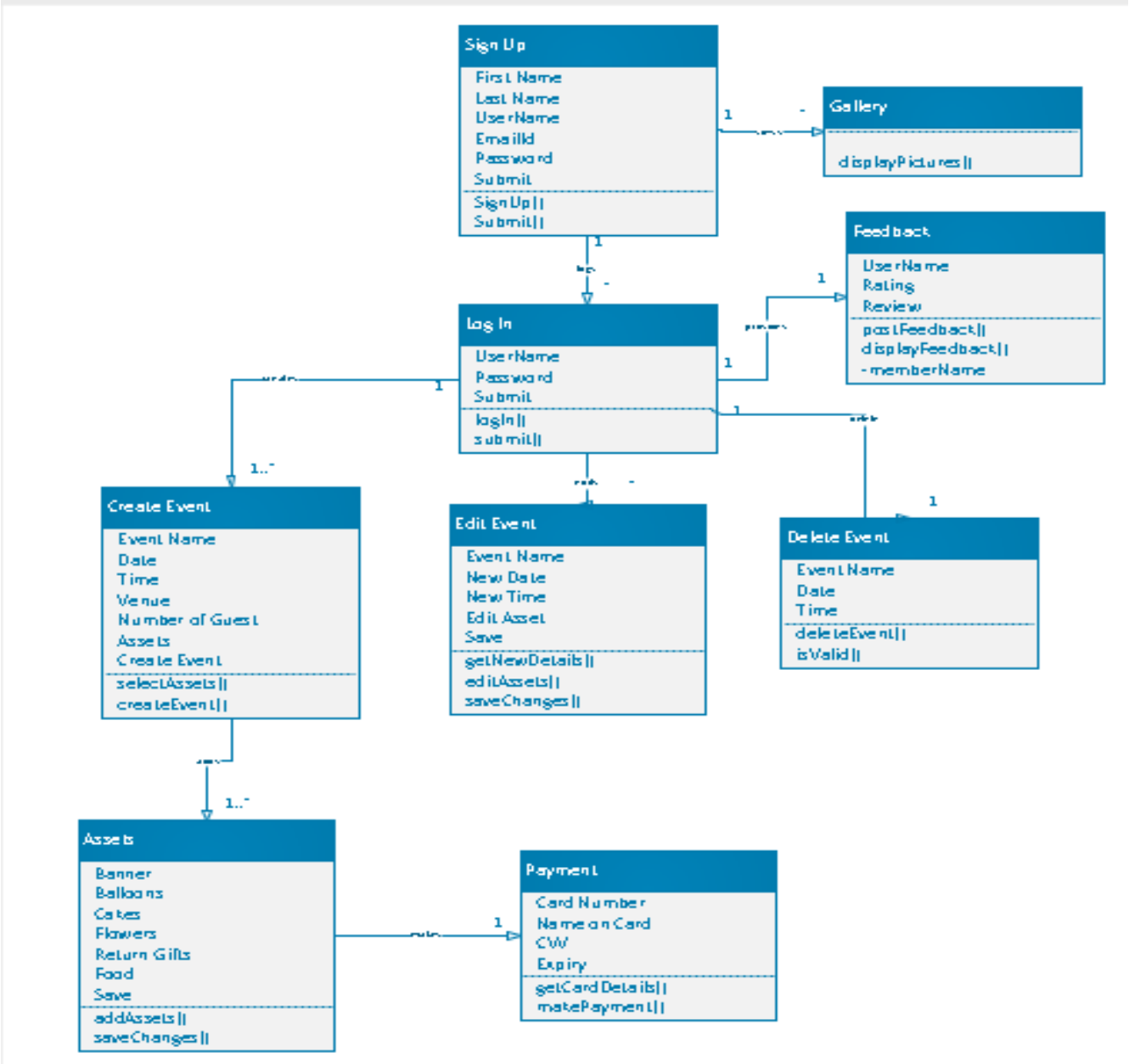


Fig 1. Class Diagram for EventMe

4. Sequence Diagram

The sequence diagram shows the interaction between any two objects in a sequential manner. It is also known as event diagrams or event scenarios. The sequence diagram for our project is shown below:

GitHub link: https://github.com/KiruthigaSekar/Software-Engineering/blob/master/Sprint%203/SequenceDiagram_EventMe.vsdx

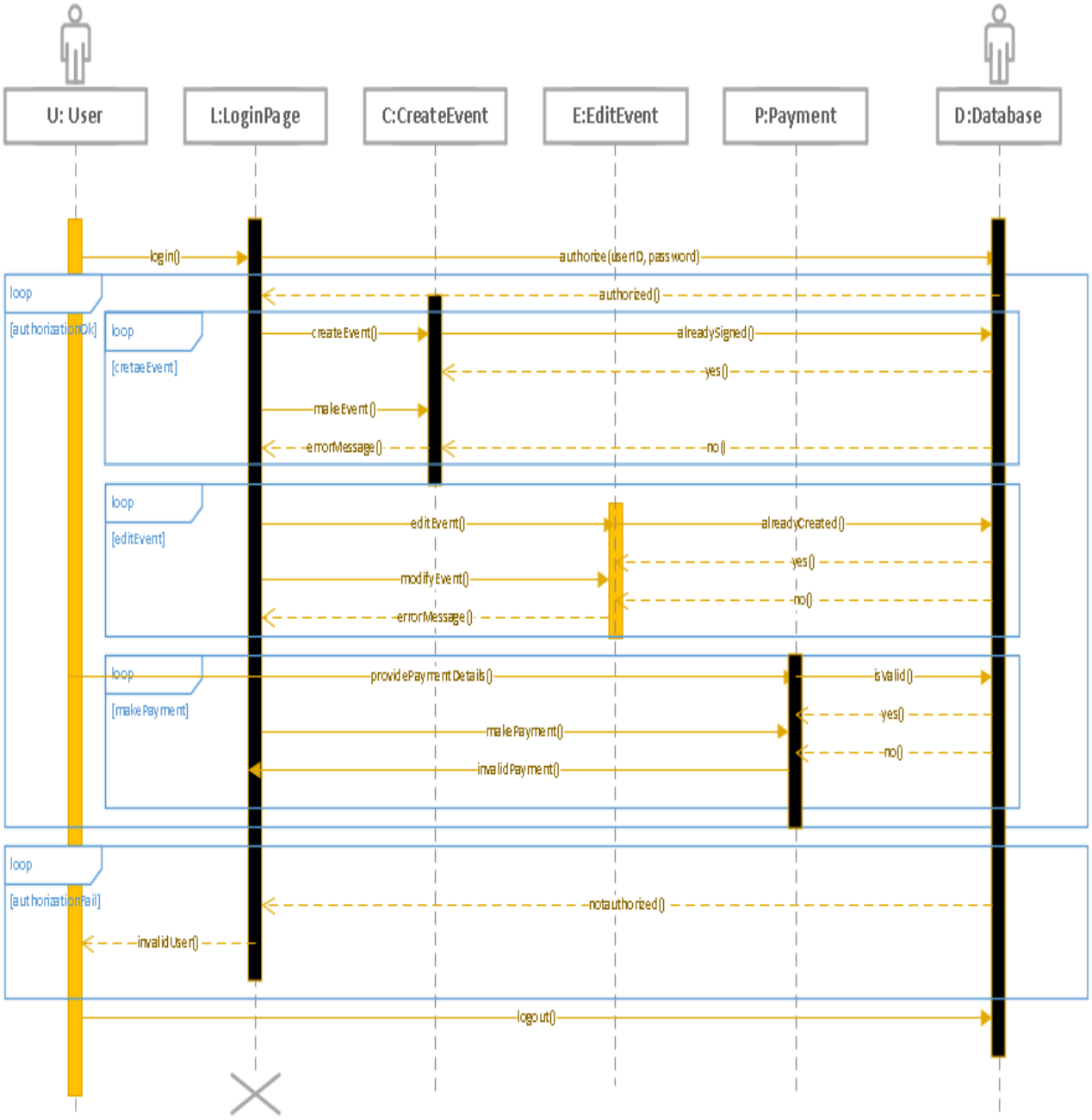


Fig 2. Sequence Diagram of EventMe

5. Architecture Diagram

Architecture is the first step of the design. It partitions the requirements onto self-contained sus-systems. It is important to plan how these sus-systems communicate and co-operate.

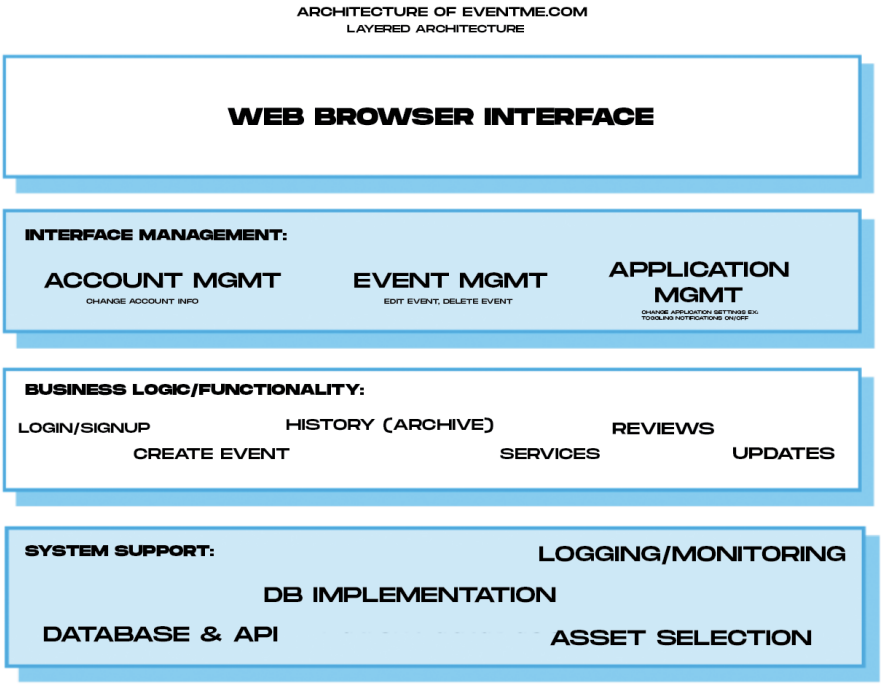


Fig 3. Architecture design of EventME

5.1 Logical View

It decomposes the system structure into software components and connectors. It also maps use cases onto the components.

GitHub link: https://github.com/KiruthigaSekar/Software-Engineering/tree/master/Sprint%204

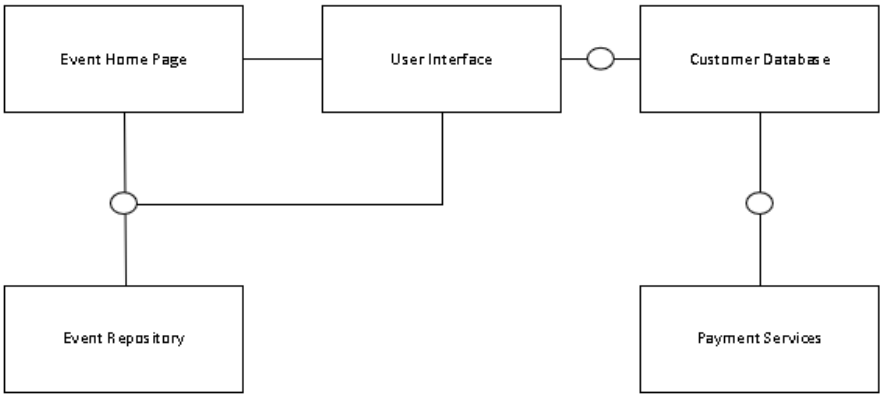


Fig 4. Logical View of EventMe

5.2 Process View

The Process View models the dynamic aspects of the architecture and the behavior of its parts. It describes how the process/threads should communicate with each other.

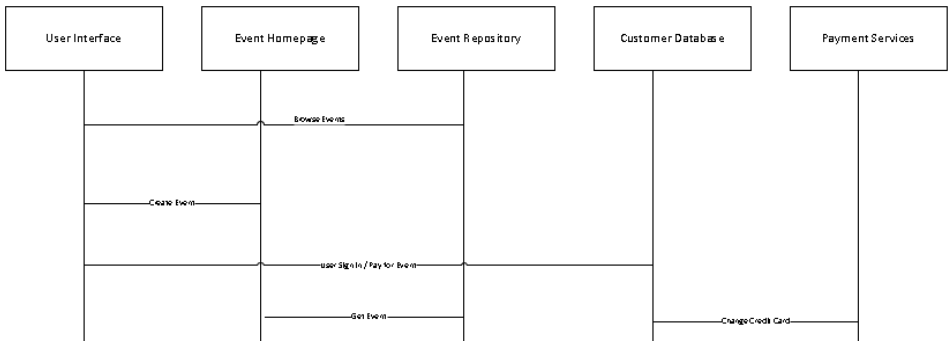


Fig 5. Process View of EventMe

5.3 Development View

The Development View is the static organization of the software code artifacts. Mapping between the elements in the logical view and the code artifacts.

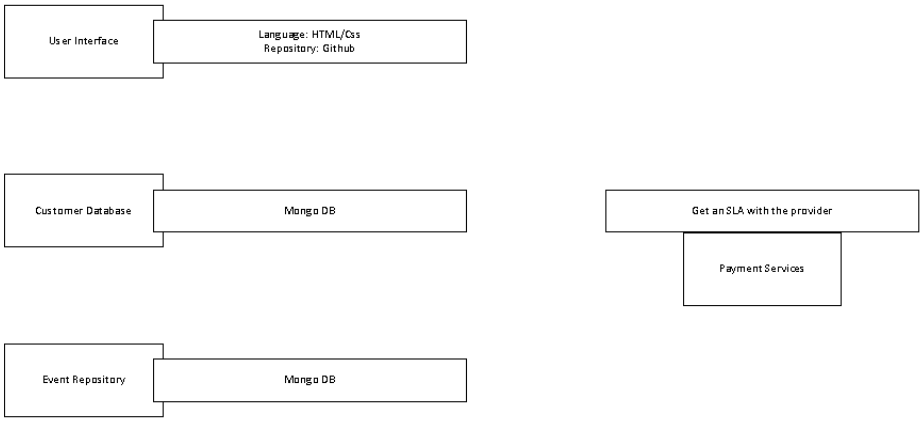


Fig 6. Development View of EventMe

5.4 Physical View

The Physical View is the hardware environment where the software will be deployed. This view maps the logical and the physical entities of the system.

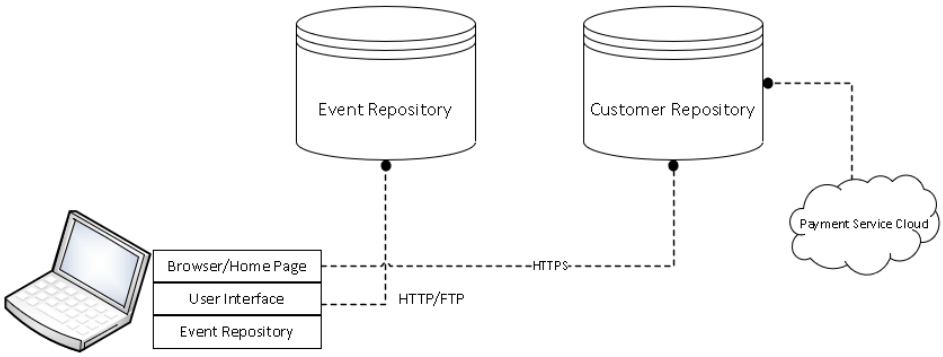


Fig 7. Physical View of EventMe

6. Use Case

Use Case Diagram:

The Use Case Diagram shows the interaction between the system and its environment. The modified use case diagram is shown below:

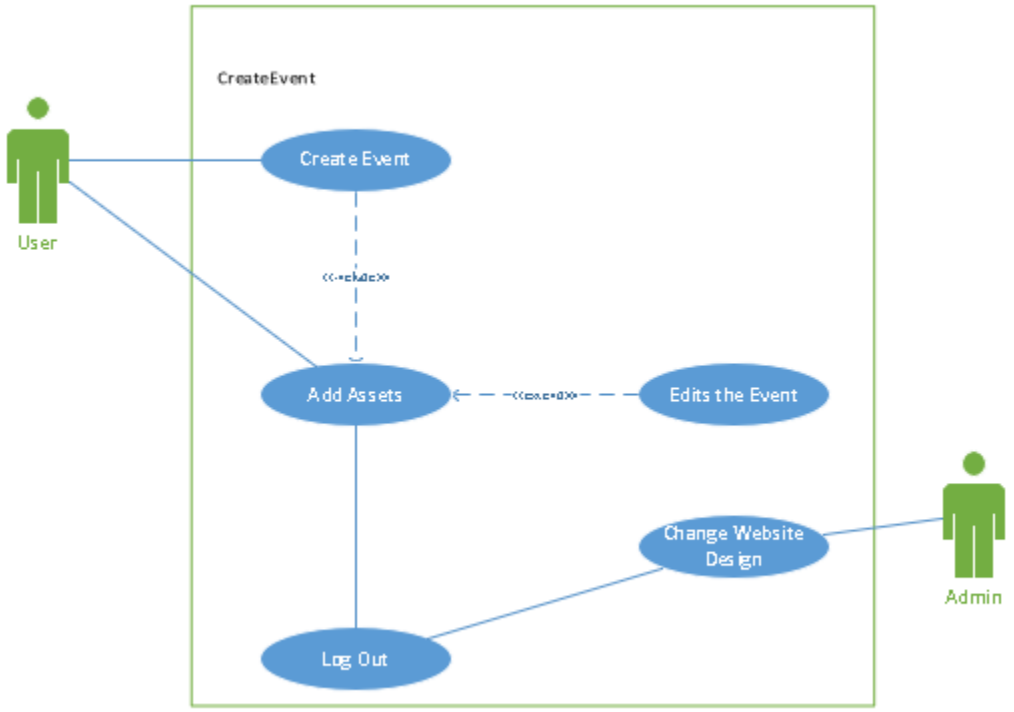


Fig8. Use Case of Create Event

Use Case #1

Name of the Use Case: Sign-Up
Actors: User, Admin, Database
Description:

- The user will be provided with the login/sign up page, once accessing the EventMe site
- The user will sign up (create account) by providing the details such as Name, Age, Gender, Email ID, Username, Password
- After filling the required fields, the user will hit the Sign-Up button.
- If the account is successfully created, the user will be directed to the Create Event Page
- If the account is not created, (in case of invalid email id), an error message will be prompted to the user
- The information provided by the user are stored in the database
- Admin can change the look and feel of the website and anything in the database.

Exception Path: If the user does not have an account with EventMe, they will be prompted to Sign Up
Alternate Path: There is only one path to this case
Pre-Condition: User must have reached out to the website

Use Case #2

Name of the Use Case: Log In/Log Out
Actors: User, Admin, Database
Description:

- The user after creating the account, must be able to log in to the system
- The user must provide their email id and password and hit the log in button
- The user will be directed to the home page
- The user can create a new event or can modify the existing event after logging in
- The user should be able to provide the upper limit of the budget of the event, so that the expense does not exceed the limit
- The user can also provide a feedback of the service provided

Exception Path: Once the user is signed up to EventMe, he/she can log in to the system
Alternate Path: There is only one path to this case, unless error message is prompted
Pre-Condition: Valid account must have already been made

Use Case #3

Name of the Use Case: Create Event
Actors: User, Admin, Database
Description:

- Once the user is successfully logged into the system, they must be able to create an event
- When the user clicks on the create event tab, a list of all the events are displayed like birthday party, wedding, corporate events, get together, conference, expo and so on
- After selecting the event, the user must hit the next button
- The user must click on the desired event
- The system will be directed to the next page, where they can select the assets and theme

Exception Path: The user cannot click on the next button, without selecting any event.
Alternate Path: There is only one path to this case
Pre-Condition: The user must have a valid account and signed in successfully

Use Case #4

Name of the Use Case: Assets, Event Details
Actors: User, Admin, Database
Description:

- When the user selects an event from the create event page, the system will direct the user to the next page
- The next page is the Assets and Event Details page
- In this page, the user can select the assets based on the theme of the event/party
- The user must make sure that they do not exceed the budget of the party
- The price of each asset is displayed in this page, such that the user is aware it
- When the required assets are selected
- The user must provide the details about the event, such as the venue, date and time and then hit the next button

Exception Path: The user must select at least on asset and provide the details about the event to proceed to the next page.

Alternate Path: There is only one path to this case
Pre-Condition: The user must select an event in order to get to this page

Use Case #5

Name of the Use Case: Budget Upper Limit
Actors: User, Admin, Database
Description:

- When the user is in the assets and event details page, they must be able to provide the budget of the event, before selecting the assets
- When the user tends to pick items, exceeding the budget, a warning message is displayed to the user, prompting that they are attempting to select an item above the budget
- At this point, the user can either increase the budget of the event or can proceed to check out

Exception Path: Warning message should be displayed, if the user exceeds the budget
Alternate Path: There is only one path to this case
Pre-Condition: The user must have selected an event
Post-Condition: The user can select the assets they require for the event after specifying the budget

Use Case #6

Name of the Use Case: Edit / Delete Event
Actors: User, Admin, Database
Description:

- When the user successfully created an event and specified the assets required for the event, they must be able to edit/delete the event
- Any editing or deleting the event should be done 48 hours prior to the event date
- The user will not be able to edit/delete the event when there is only 48 hours left for the event
- The user can change the assets and the event details

Exception Path: Waring will be prompted, if the user tends to edit/delete within 48 hours of the event
Alternate Path: There is no alternate path to this case
Pre-Condition: The user can edit/delete before 48 hours

Use Case #7

Name of the Use Case: Event Progress Notification
Actors: User, Database
Description:

- User Logs in to EventMe.com (Assuming user has went through sign up process)
- User will then navigate to MyEvent Page (Account Page)
- Notification is given on this page if the event status has been updated. (Assuming an event has already been created)

Exception Path: If the user does not have an account with EventMe, or the User has not already created an event there will not be notifications.
Alternate Path: There is only one path to this case
Pre-Condition: Sign Up & Completion of Create New Event

Use Case #8

Name of the Use Case: View Progress
Actors: User, Database
Description:

- User Logs in to EventMe.com (Assuming user has went through sign up process)
- User will then navigate to MyEvents Page (Account Page)
- User can view status report of all Events created by the user on this page.

Exception Path: If the user does not have an account with EventMe, or the User has not already created an event there will not be a way for the user to see event progress if one doesn't exist already.
Alternate Path: There is only one path to this case
Pre-Condition: Sign Up & Completion of Create New Event

Use Case #9

Name of the Use Case: View All Events (History/Gallery)
Actors: User, Database
Description:

- User visits EventMe.com
- Visits Home Page
- User will then navigate to History Page (made available to user on the home page)
- User can view all the past events that EventMe.com has done in form of a gallery page and short descriptions.

Exception Path: If the user does not have access to the EventMe.com site
Alternate Path: There is only one path to this case
Pre-Condition: none / access to EventMe.com

Use Case #10

Name of the Use Case: Payment (First Time User)
Actors: User
Description:

- User will navigate to the Create New Event Page from Home Page and begin the process
- Once the user has completed filling out all the information, they need for the Event they have created they will be forwarded to the Payment page.
- On Payment page User is prompted to add their credit/debit card info and select the type of payment they would like to do such as "Pay All at Once", "Finance over 3 Months", and etc.

Exception Path: If the user does not have an account with EventMe, or the User has not already created an event they will not be able to enter in payment information. If the user enters in incorrect payment information this will also result in an error.
Alternate Path: User can visit their account page on EventMe.com and add new payment information or edit existing payment info.
Pre-Condition: User should have already gone through the Create Event Path and added their initial payment information to the account.

Use Case #11

Name of the Use Case: Feedback/Reviews (View)
Actors: User
Description:

- User will navigate to the Home Page.
- User can now click on Feedback/Reviews tab
- Now the user is able to view reviews from other EventMe.com Account holders about how their event turned out.

Exception Path: If the user does not have access to EventMe.com
Alternate Path: User can visit their account page on EventMe.com and view/write a review on the event that they have completed.
Pre-Condition: none/ access to EventMe.com

Use Case #12

Name of the Use Case: Feedback/Reviews (Write)
Actors: User
Description:

- User will sign in and navigate to account page.
- User can navigate from account page to a "Write a Review" subpage which will allow them to view their own events created with EventMe.com
- Under the user's event (Only if it has been completed) there will a button to write a review on the event that they've selected.
- Once the button has been selected the user can write their review in a drop down text field on that page.

Exception Path: If the user does not have an account to EventMe.com or If they have not completed an Event with EventMe.com
Alternate Path: Only Path
Pre-Condition: Created an event, completed an event with EventMe.com, have an account.

7. System Modelling

The Uses Case diagram of the system is provided below.

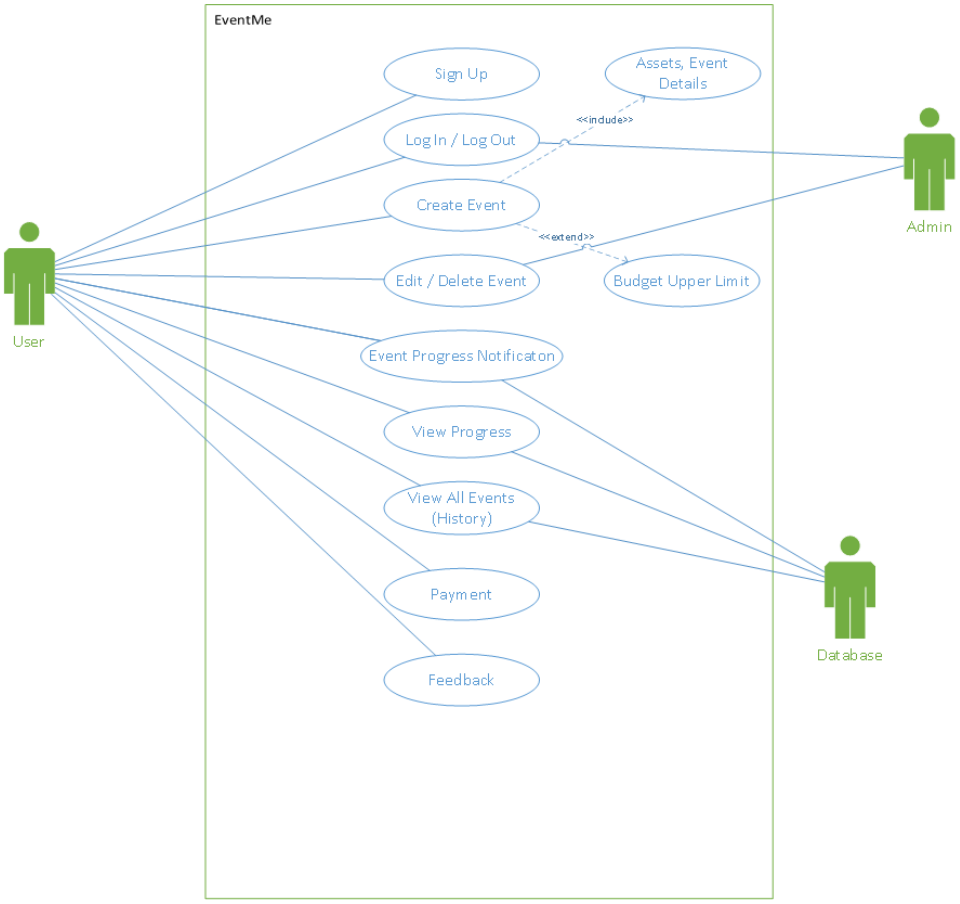


Fig 9. Uses Case Diagram of EventMe

8. System Requirements with Test Cases

Requirement# 1
Use Case# 1
Name: Sign up
Introduction: The user will be able to register an account with us so that they can login and use our services.
Input: First Name, Last Name, Email, Password, Confirm Password, Phone Number, Home Address
Requirement description: The user will be able to register an account with us if all requirements are fulfilled.

1.All fields are completed
2.All the requirements from the system are met.

Output: A screen to show that the registration has been successfully completed. Then direct to login page.

Test Cases:
Test Case ID: 1.1
Description: Sign Up with email and password with legal format.
Test Inputs: Email and password not currently in Database table linked to a User account.
Expected Results: Successful! Then redirect to log in page.
Dependencies: None
Initialization: Email and password with legal format
Test Steps:

1. Submit a valid email and password that is not in our Database System
2. Click submit and wait for successful message to be redirected to Login Page.

Test Case ID: 1.2
Description: Sign Up with an existing User's email and password
Test Inputs: Email and password that is already been registered in Database table
Expected Results: Unsuccessful, an error message to show this username has already been used, please try again.
Dependencies: None
Initialization: Existing email and password
Test Steps:

1. Submit an existing email and password in database table.
2. An error message to show this username has already been used, please try again.

Test Case ID: 1.3
Description: Sign Up with email and password with illegal format.
Test Inputs: Email and password are not in correct format
Expected Results: Unsuccessful, an error message to show this username has illegal format, please try again.
Dependencies: None
Initialization: Email and password with illegal format
Test Steps:

1. Submit email and password with incorrect format.
2. An error message to show this username has illegal format, please try again.

Requirement# 2
Use Case# 2
Name: Log In/Log Out
Introduction: After successfully sign up an account with us, users will be able to login and use our services.
Input: Email and password.
Requirement description: The user will be able to login if all requirements are met.

1.All fields are completed
2.All the requirements from the system are met.
3.All entered information are matched to our database validation.

Output: Directed to Create Event page.

Test Cases:
Test Case ID: 2.1
Description: Login with correct email and its dependent password.
Test Inputs: Email and password with correct format and registered in database table.
Expected Results: Successful! Then redirect to main user page.
Dependencies: None
Initialization: correct email and its dependent password
Test Steps:

1. Submit a valid email and its dependent password that is in our Database System
2. Click submit and wait for successful message to be redirected to main page

Test Case ID: 2.2
Description: Login with incorrect email and password.
Test Inputs: Email and password that is not registered in database table.
Expected Results: Unsuccessful! Then redirect to login page.
Dependencies: None
Initialization: invalid email and password
Test Steps:

1. Submit an invalid email and password that is not in our Database System
2. Click submit and return to login page

Requirement# 3
Use Case# 3
Name: Create Event
Introduction: After successfully login, users will be able to create events with us and this page will guide them through to create an event with us.
Input: No input.
Requirement description: The user will be able to create an event with us if all requirements are fulfilled.

1.At least one event is selected.
2.At least one asset and theme are selected for each.

Output: Directed to Assets, Event Details page.

Test Cases:
Test Case ID: 3.1
Description: The user selects a type of event he or she wants to create.
Test Inputs: None
Expected Results: Direct to Assets, Event Details page.
Dependencies: The user must select one event
Initialization: None
Test Steps:

1. Click the event that user wants to create
2. Click submit button

Test Case ID: 3.2
Description: The user did not select any event
Test Inputs: None
Expected Results: Error to show that you must select an event.
Dependencies: None
Initialization: None
Test Steps:

1.Leave the field empty
2.Click submit button

Requirement# 4
Use Case# 4
Name: Assets, Event Details
Introduction: This will help users to customize the event.
Input: No input
Requirement description: The user will be able customize the event with us if all requirements are fulfilled.

1. The user must select at least on asset.
2. select the assets based on the theme of the event.

Output: Directed to payment page.

Test Cases:
Test Case ID: 4.1
Description: The user selects assets, budget and provides details about the event
Test Inputs: None
Expected Results: Event has been created, direct to payment page
Dependencies: Must fill out all the information.
Initialization: None
Test Steps:

1. Fill out all the information.
2. Click submit button

Test Case ID: 4.2
Description: The user does not select all assets, budget or fail to provide details about the event
Test Inputs: None
Expected Results: Error message shows that all information must be provided!

Dependencies: Fail to fill out all the information.
Initialization: None

Test Steps:

1. Leave blanks on the information sections.
2. Click submit button

Requirement# 5
Use Case# 5
Name: Budget Upper Limit
Introduction: This will help users to keep down their budget so that they don't exceed the budget limit.
Input: Buget Limit
Requirement description: The user will be able to complete creating event if the service cost that they select is under the budget limit.
Output: No output

Test Cases:
Test Case ID: 5.1
Description: After the user selects assets, budget and provides details about the event, Buget is under cost.
Test Inputs: Budget
Expected Results: Event has been created, direct to payment page
Dependencies: Must fill out the budget limit.
Initialization: Budget =or<cost
Test Steps:

1. Fill out all the information.
2. Click submit button

Test Case ID: 5.2
Description: After the user selects assets, budget and provides details about the event, Buget is over limit.
Test Inputs: Budget
Expected Results: Over the budget limit, please lower your cost
Dependencies: Must fill out the budget limit.
Initialization: Budget >cost
Test Steps:

1. Fill out all the information.
2. Click submit button

Requirement# 6
Use Case# 6
Name: Edit / Delete Event
Introduction: When the user successfully created an event and specified the assets required for the event, they must be able to edit/delete the event.
Input: No input
Requirement description: After successfully created an event and specified the assets, the user can only Edit / Delete Event 48 hours prior to the event date
Output: No output.

Test Cases:
Test Case ID: 6.1
Description: the user Edits Event 48 hours prior to the event date
Test Inputs: None
Expected Results: Successful! Your event has been Edited!
Dependencies: An event must be created and has not been participated
Initialization: None
Test Steps:

1. Edit assets for an event that has been created
2. Click submit button

Test Case ID: 6.2
Description: the user deletes Event 48 hours prior to the event date
Test Inputs: None
Expected Results: Successful! Your event has been deleted!
Dependencies: An event must be created and has not been participated
Initialization: None
Test Steps:

1. Delete an event that has been created
2. Click submit button

Test Case ID: 6.3
Description: the user edits/deletes Event less than 48 hours before the event date
Test Inputs: None
Expected Results: Unsuccessful! You must edits/deletes Event 48 hours prior to the event!
Dependencies: An event must be created and has not been participated
Initialization: None
Test Steps:

1. Edit/delete Event less than 48 hours before the event date
2. Click submit button

Requirement# 7
Use Case# 7
Name: Event Progress Notification
Introduction: After an existed user login and directed to the account page, a notification is given on this page if the event status has been updated.
Input: No input.
Requirement description: The user must have successfully created an event with us, and the event status is updated.
Output: The event status.

Test Cases:
Test Case ID: 7.1
Description: The event status is updated
Test Inputs: None
Expected Results: An updated notification is shown on the page
Dependencies: An event must be created and has not been participated
Initialization: None

Test Steps:

1. Click Notification button

Test Case ID: 7.2
Description: The user has no past event created and goes to notification page
Test Inputs: None
Expected Results: Nothing is shown on the event status page
Dependencies: None
Initialization: None
Test Steps:

1. Click Notification button

Requirement# 8
Use Case# 8
Name: View Progress
Introduction: User can view status report of all Events created by the user on this page.
Input: No input
Requirement description: The user must have successfully created an event with us, and the event status is updated.
Output: Status Report.

Test Cases:
Test Case ID: 8.1
Description: The user views status report of all Events.
Test Inputs: None
Expected Results: Status report is available.
Dependencies: At least one event must be created and has not been participated
Initialization: None
Test Steps:

1. The event status report is shown on the page

Requirement# 9
Use Case# 9
Name: View All Events (History/Gallery)
Introduction: User can view all the past events that EventMe.com has done in form of a gallery page and short descriptions.
Input: No input
Requirement description: The user must have a valid account with us.
Output: Events in form of a gallery page and short descriptions.

Test Cases:
Test Case ID: 9.1
Description: The user views all the past events that EventMe.com has done in form of a gallery page and short descriptions.
Test Inputs: None
Expected Results: Past events that EventMe.com has done in form of a gallery page and short descriptions.

Dependencies: Must have account with us.
Initialization: None

Test Steps:
1. Click on View All Events

Test Case ID: 9.2
Description: The user has no account with us and want to view all the past events that EventMe.com has done
Test Inputs: None
Expected Results: Error message: Please login first
Dependencies: none
Initialization: None
Test Steps:
1. Click on View All Events

Requirement# 10
Use Case# 10
Name: Payment (First Time User)
Introduction: This will help users to complete the payment with us after completed filling out all the information they need for the Event they have created
Input: Type of the card, First and last name on the card, Credit or debit card number, Security code, Expiration date, and billing address.
Requirement description: The user will be able to make a payment with us if all requirements are fulfilled.
1.All fields are completed
2.Database will verify all the information is correct and secured.
Output: A screen shows that the event has been successfully created.

Test Cases:
Test Case ID: 10.1
Description: The user fills out all the required information and all the information are valid.
Test Inputs: All the information about credit/debit card.
Expected Results: Payment has been processed
Dependencies: Bank account.
Initialization: Information must be correct in order to make payment
Test Steps:
1. All fields are completed and correct.
2. Click submit

Test Case ID: 10.2
Description: The user fails to fill out all the required information
Test Inputs: Leave some blanks
Expected Results: Payment has not been processed
Dependencies: Bank account.
Initialization: All fields must be filled

Test Steps:

1. Some blanks on the fields
2. Click submit

Test Case ID: 10.3
Description: The user provides incorrect information about credit/debit card.
Test Inputs: Incorrect information about credit/debit card.
Expected Results: Payment has not been processed
Dependencies: Bank account.
Initialization: Information must be correct in order to make payment
Test Steps:

1. Incorrect information is provided
2. Click submit

Requirement# 11
Use Case# 11
Name: Feedback/Reviews (View)
Introduction: User can view reviews from other EventMe.com Account holders about how their Event is turned out.
Input: No input
Requirement description: The user must have access to our website.
Output: Reviews from other EventMe.com Account holders.

Test Cases:
Test Case ID: 11.1
Description: View reviews from other EventMe.com Account holders about how their Event is turned out.
Test Inputs: None
Expected Results: Show reviews
Dependencies: The user must have access to our website.
Initialization: none
Test Steps:

1. Click view review.

Test Case ID: 11.2
Description: The user has no account with us and click show review button
Test Inputs: None
Expected Results: Error: Please login
Dependencies: None
Initialization: none
Test Steps:

1. Click view review.

Requirement# 12
Use Case# 12
Name: Feedback/Reviews (Write)
Introduction: User can write reviews about how their Event is turned out for other EventMe.com Account holders to see.
Input: Reviews in English.
Requirement description: The user must have access to our website and created event(s), paid the bills and participated.
Output: The review will be available for other users to see.

Test Cases:
Test Case ID: 12.1
Description: The user wants to write a review about an event.
Test Inputs: Reviews
Expected Results: Review has been posted!
Dependencies: The user must have access to our website and has created in one or more than one event.
Initialization: none
Test Steps:

1. Click write review and write reviews.
2. Click post.

Test Case ID: 12.2
Description: The user leave the field empty.
Test Inputs: None
Expected Results: error: Please leave you feedback
Dependencies: The user must have access to our website and has created in one or more than one event.
Initialization: none
Test Steps:

1. Click write review and write reviews.
2. Click post.

9. Testing

Test ID	**1.1**
Purpose of Test	Sign Up with email and password with legal format
Test Environment	I will be using unit testing JavaScript with Jasmine to test the sign-up page

Test Step	describe ("A Sign up is just a function", signup() {
	it("and also is a spec", function() {
	a = true;
	expect(a).toBe(true); });
	});

Test input	Email and password not currently in Database table linked to a User account.
Expected Result	Successful! Then redirect to log in page.
Likely problems/Bug Revealed	N/A

Test ID	**1.2**
Purpose of Test	Sign Up with an existing User's email and password
Test Environment	I will be using unit testing JavaScript with Jasmine to test the sign-up page

Test Step	describe ("A Sign up is just a function", signup() {
	it("and also is a spec", function() {
	a = true;
	expect(a).toBe(true); });
	});

Test input	Email and password that is already been registered in Database table
Expected Result	Unsuccessful, an error message to show this username has already been used, please try again.
Likely problems/Bug Revealed	N/A

Test ID	**1.3**
Purpose of Test	Sign Up with email and password with illegal format.
Test Environment	I will be using unit testing JavaScript with Jasmine to test the sign-up page

Test Step	describe ("A login is just a function", signup() {
	it("and also is a spec", function() {
	a = true;
	expect(a).toBe(true); });
	});

Test input	Email and password are not in correct format
Expected Result	Unsuccessful, an error message to show this username has illegal format, please try again.
Likely problems/Bug Revealed	N/A

Test ID	**2.1**
Purpose of Test	Login with correct email and its dependent password.
Test Environment	I will be using unit testing JavaScript with Jasmine to test the sign-up page

Test Step	describe ("A login is just a function", signup() {
	it("and also is a spec", function() {
	a = true;
	expect(a).toBe(true); });
	});

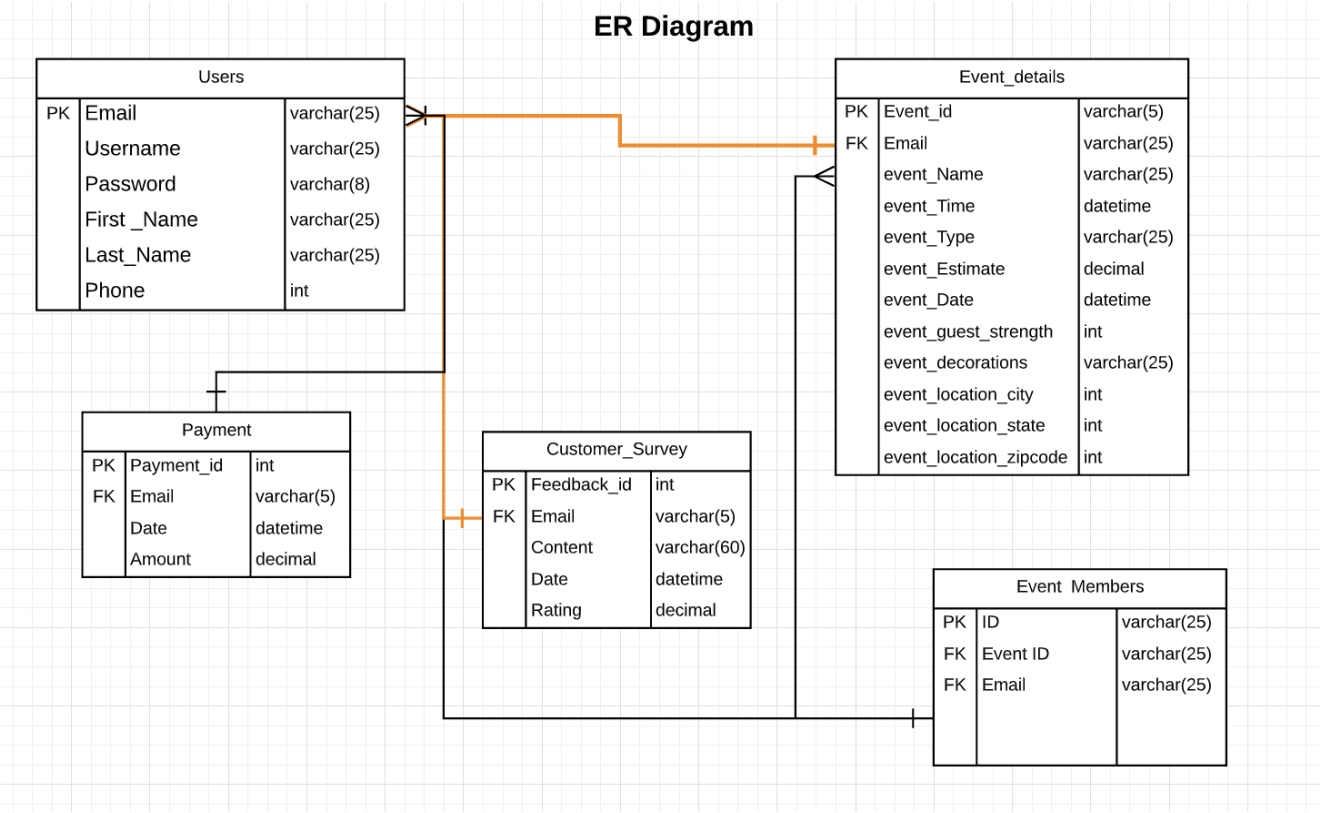
Test input	Email and password with correct format and registered in database table.
Expected Result	Successful! Then redirect to main user page.
Likely problems/Bug Revealed	N/A

Test ID	**2.2**
Purpose of Test	Login with incorrect email and password.
Test Environment	I will be using unit testing JavaScript with Jasmine to test the sign-up page

Test Step	describe("A login is just a function", signup() {
	it("and also is a spec", function() {
	a = true;
	expect(a).toBe(true); });
	});

Test input	Email and password that is not registered in database table
Expected Result	Unsuccessful! Then redirect to login page.
Likely problems/Bug Revealed	N/A

10. Database Specification & Analysis:



In the above diagram PK indicated Primary Key and FK indicates the foreign key.
Our database consists of 3 Tables which are Users, Event_details, Cutomer_Survey.

Users Table: Users table is used for storing all the sensitive information regarding the user. Our primary key in the table is “Email”, rest of the variables are shown in the above diagram.

Event Details: This table is used for storing all the information regarding the events created by the users. Primary Key in this table is “Event_Id” which would be unique and created by us during the event submission. The Foreign key would be the “Email” of the User by which event is created.

Customer Survey: This table is used for storing all the information regarding the survey or feedback submitted by the clients. Primary Key in this table is “Feedback_Id” which would be unique and created by us during the survey submission. The Foreign key would be the “Email” of the User who submits the feedback and it’s optional. So, users can submit anonymous feedback to our events.

Database Specification: We are MONGODB which is a non-relational database as our backend database for storage of information regarding the users, events in a secure way.

11. Implementation:

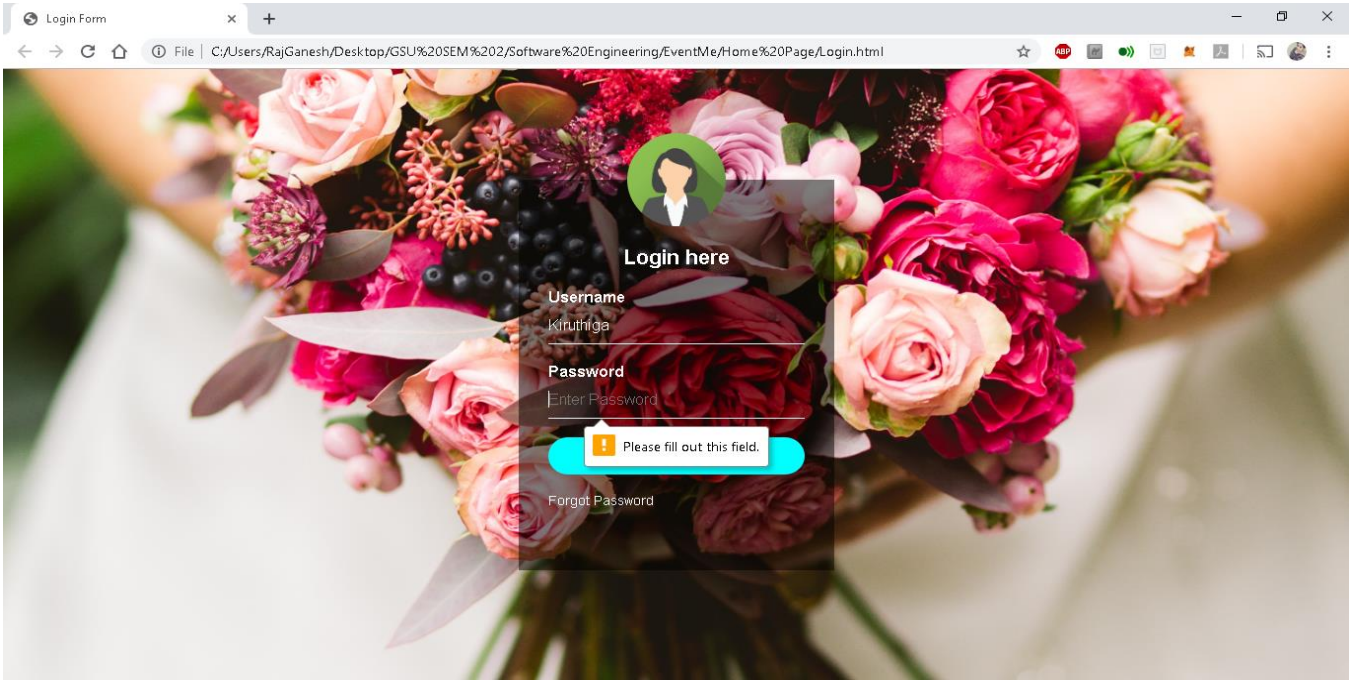
Database:

DATABASE SIZE: 0B INDEX SIZE: 16KB TOTAL COLLECTIONS: 4

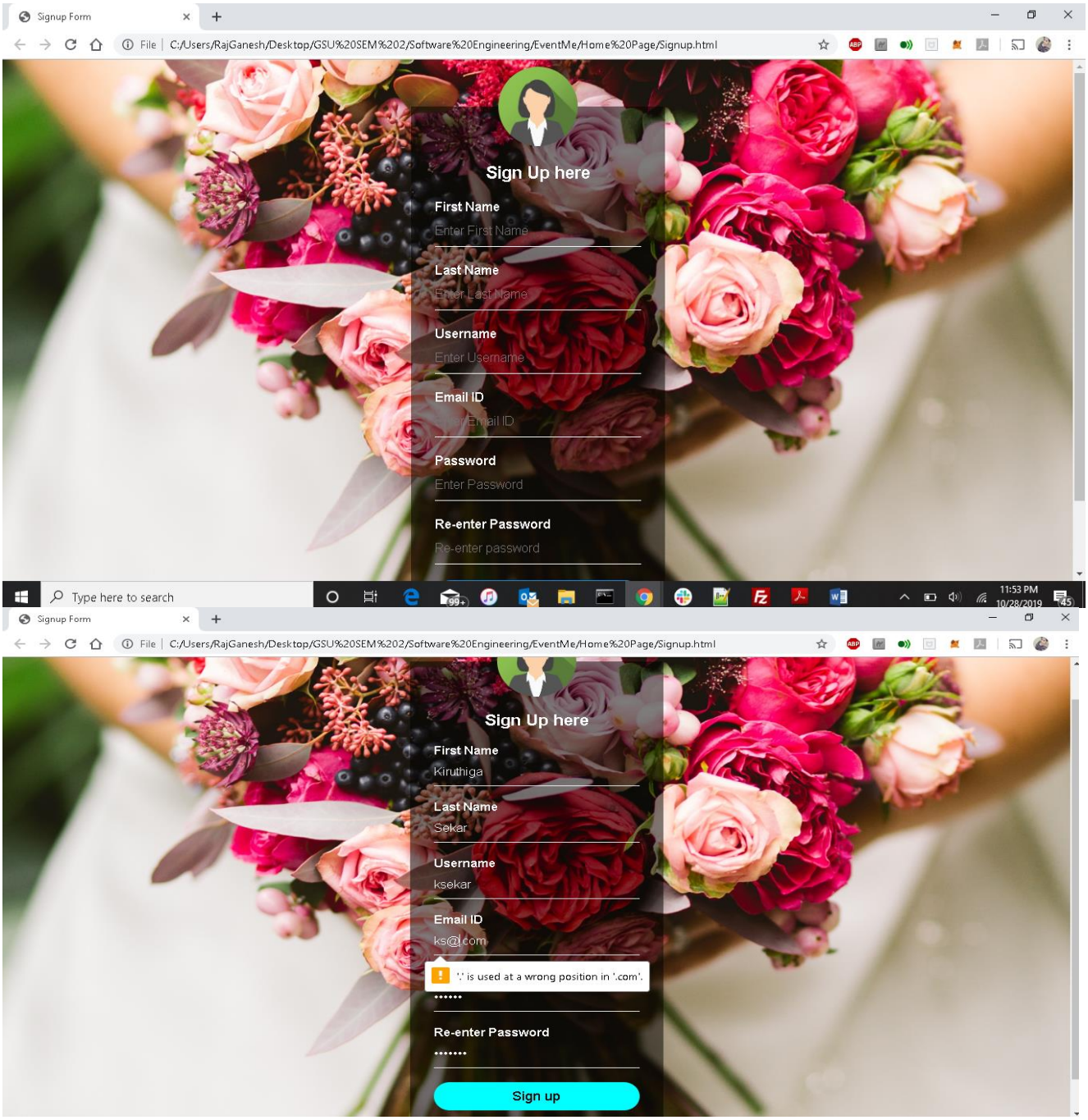
CREATE COLLECTION

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
customer_survey	0	0B	0B	1	4KB	4KB
event_details	0	0B	0B	1	4KB	4KB
payment	0	0B	0B	1	4KB	4KB
users	0	0B	0B	1	4KB	4KB

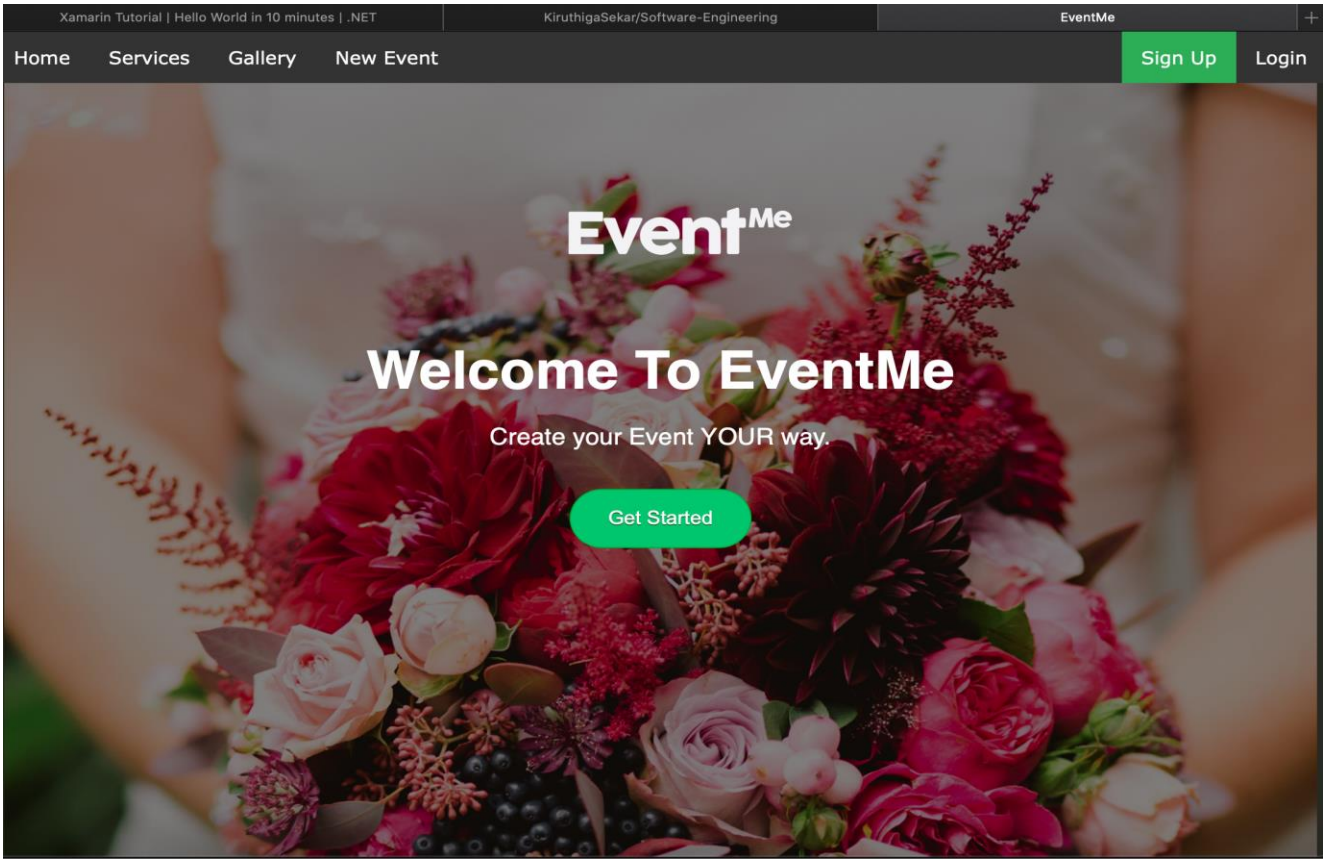
Login:



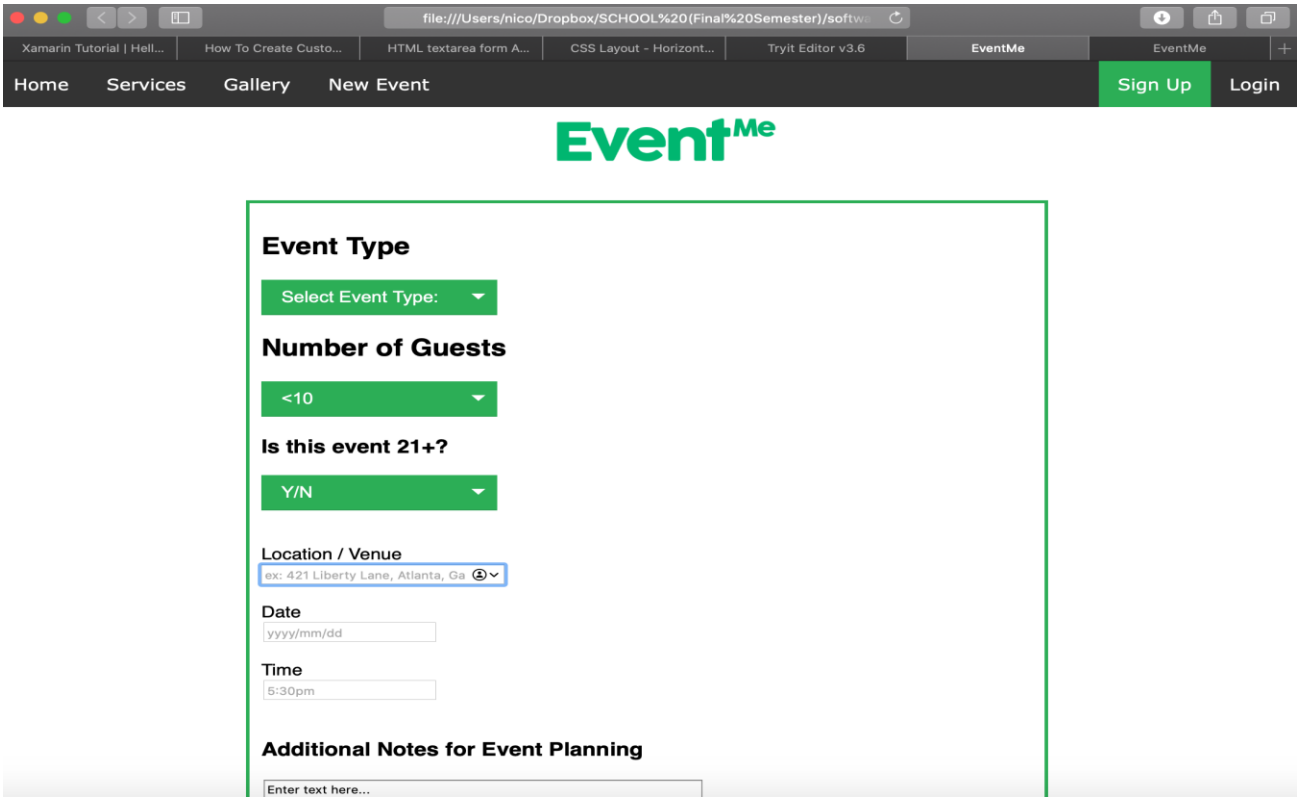
SignUp:


Signup Form
File | C:/Users/RajGanesh/Desktop/GSU%20SEM%202/Software%20Engineering/EventMe/Home%20Page/Signup.html
Sign Up here
First Name
Enter First Name
Last Name
Enter Last Name
Username
Enter Username
Email ID
Enter Email ID
Password
Enter Password
Re-enter Password
Re-enter password
Type here to search
11:53 PM
10/28/2019
Signup Form
File | C:/Users/RajGanesh/Desktop/GSU%20SEM%202/Software%20Engineering/EventMe/Home%20Page/Signup.html
Sign Up here
First Name
Kiruthiga
Last Name
Sekar
Username
ksekar
Email ID
ks@i.com
'.' is used at a wrong position in '.com'.
Re-enter Password
Sign up

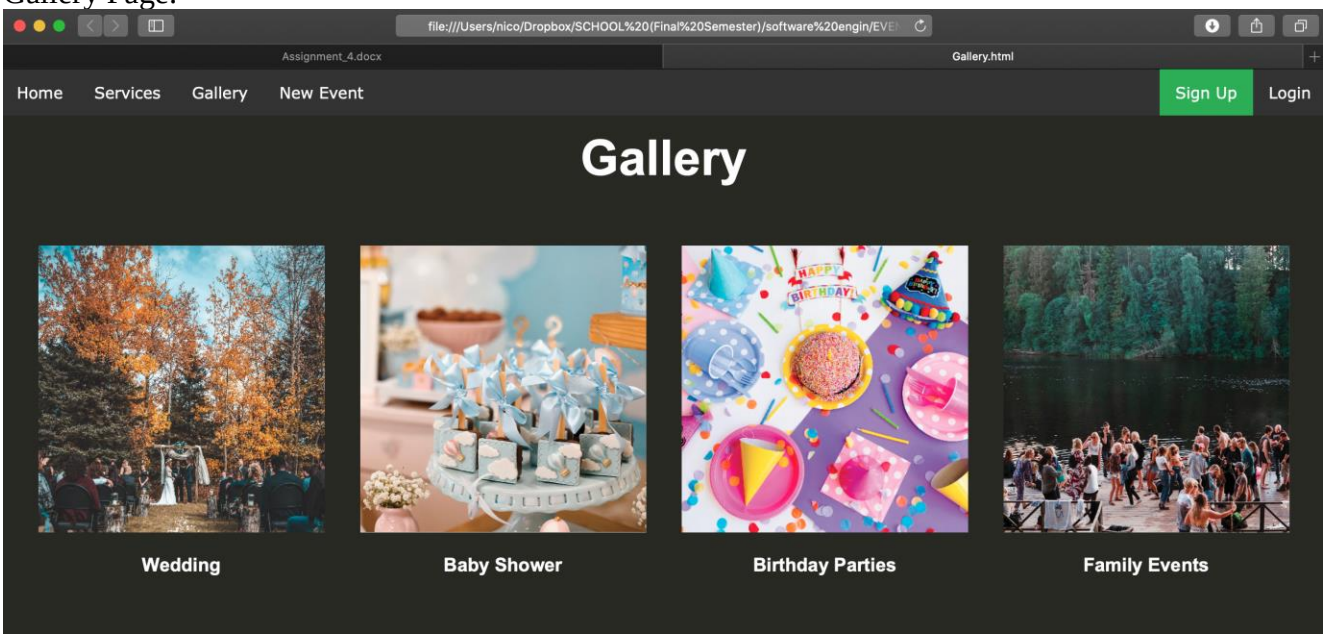
HomePage:



Create New Event

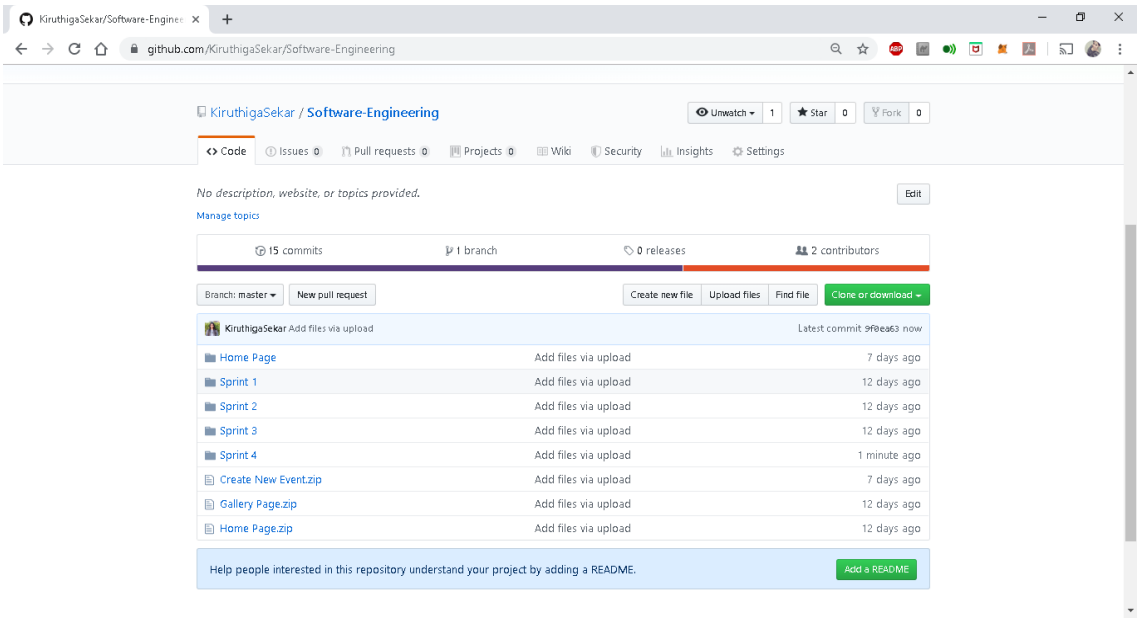


Gallery Page:



APPENDIX

1. GITHUB



2. Trello Channel for TeamWork Distribution and Architecture