

## Description

The event planning application is an organized and dynamic method for starting your own events of which the event planner can add friends for collaboration.

## Team Members

- Alexis Leroux
- Mariia Korolenko
- Jennifer Wong
- Bibek Man Shrestha
- Imzan Khan

## Targeted Audience

Millennials as they are comfortable using technology. We have three user roles:

1. Admin (us)
2. Members
  - a. Host
  - b. Guests
3. Non-members (potential clients)

## Objective

- As a host, party planning is stressful, and we want to make the process easier by:
  - Providing resources (pre-made music playlists, theme ideas, to-do lists)
  - Food and drink recommendations/recipes
  - Collecting party-funds from guests
  - Guests will RSVP
- As a guest, parties can also be a source of stress and we would make their experience easier by:
  - Providing a map of the party location
  - Organizing carpools ahead of time
  - Help people connect with one another before/after the party

## Project Scope

- Technical Considerations:
  - Languages: HTML, CSS, JavaScript, PHP
- APIs:
  - Spotify API
  - LCBO API
  - Google Maps
  - PayPal API
  - Uber/Lyft API
  - Food2Fork
- Database:
  - MySQL
- Libraries:
  - jQuery
  - Pure CSS

## Deliverables/Features

### Must haves:

1. Users can create events/parties. This automatically makes them the “host” of the party.
2. Users can invite collaborators to have access to the same privileges as them (i.e. a party could have two or more hosts).
3. To create an event/party, users will fill out a form. Example fields in the forms will include:
  - a. Name:
  - b. Number of Guests
  - c. Theme
  - d. Date
  - e. Guests' emails
4. Users can invite people via email to a link to the web app. This invite will be in the form of a virtual invitation.
5. When users click the link in the email, they will be redirected to the web app. The page they will be redirected to will be a form where the user will decide if they would like to attend the party or not (RSVP: YES OR NO):
  - a. If yes, they will be redirected to the registration page
  - b. If no, there will be a message to say that their attendance has been noted and they will not be granted access to the event web page.
6. The registration page:

- a. Ask users to create a username and password
  - b. This will be their login every time they use the web application
7. The web application will have:
- a. A chat system
  - b. A list to see who RSVP'd and who didn't
  - c. To do lists (i.e. the host can assign people to specific tasks)
    - i. Users can add new items
    - ii. Users can cross off items
    - iii. Users can edit items
    - iv. Users can delete items
  - d. Polls (i.e. users can vote on a specific theme)

## Responsibilities and Roles

Team Member: (role)	Responsibilities
Alexis Leroux (Full Stack Web Developer)	<ol style="list-style-type: none"> <li>1. Event Location - Using Google Maps API, users will be able to see where the event is happening ahead of time.</li> <li>2. Poll - Users will be able to create polls based on different questions.</li> <li>3. Photo Sharing - Users will be able to share photos after the party and tag their friends</li> </ol>
Mariia Korolenko (UI/UX Designer)	<ol style="list-style-type: none"> <li>1. Drinks/Recipes Recommendations – The application have a page where drinks are recommended randomly, and users can click and view the drinks or recipes.</li> <li>2. RSVP status – Hosts can see lists of guests based on whether they have RSVP'd to their invitation or not</li> <li>3. Email RSVP – When the host sends an invitation to a guest, they receive an email which allows them to see the details of the event and asks them if they will be attending or not.</li> </ol>
Jennifer Wong (UI/UX Designer)	<ol style="list-style-type: none"> <li>1. Invitations to friends - Based on the event, the application will create, and send invitations to friend's email</li> <li>2. Drink/Food list - host and guests can check what drinks are being brought and add what they will be bringing</li> </ol>

	<ol style="list-style-type: none"> <li>3. Summarize of the event - Guests can check the music list, menu, drinks, cost from the event even the event is over</li> </ol>
Bibek Shrestha (Database Architect)	<ol style="list-style-type: none"> <li>1. Event Creation - Hosts will create their event, specifying details, such as, locations, how many, type of event, etc.</li> <li>2. To-Do List - Host can create an ongoing to-do list within the web application to help further organize their party</li> <li>3. Carpool - Guests can request if they need a ride, and organize with other Guests and decide whether they would like to carpool together</li> </ol>
Imzan Khan (Project Manager/Team Lead)	<ol style="list-style-type: none"> <li>1. Dynamic Music Playlist - Using the Spotify API have a "Event Playlist" which guests in the event group can add to. Also, have preset Playlists for different types of events.</li> <li>2. Event updates - Using an RSS feed, notify the guests whenever changes are made to the event, i.e. time, location, etc.</li> <li>3. Money Collection - For events which require some form of payment pool, i.e. for sharing the costs of food, decorations etc. or for events that require an entry fee.</li> </ol>

## Description of Roles:

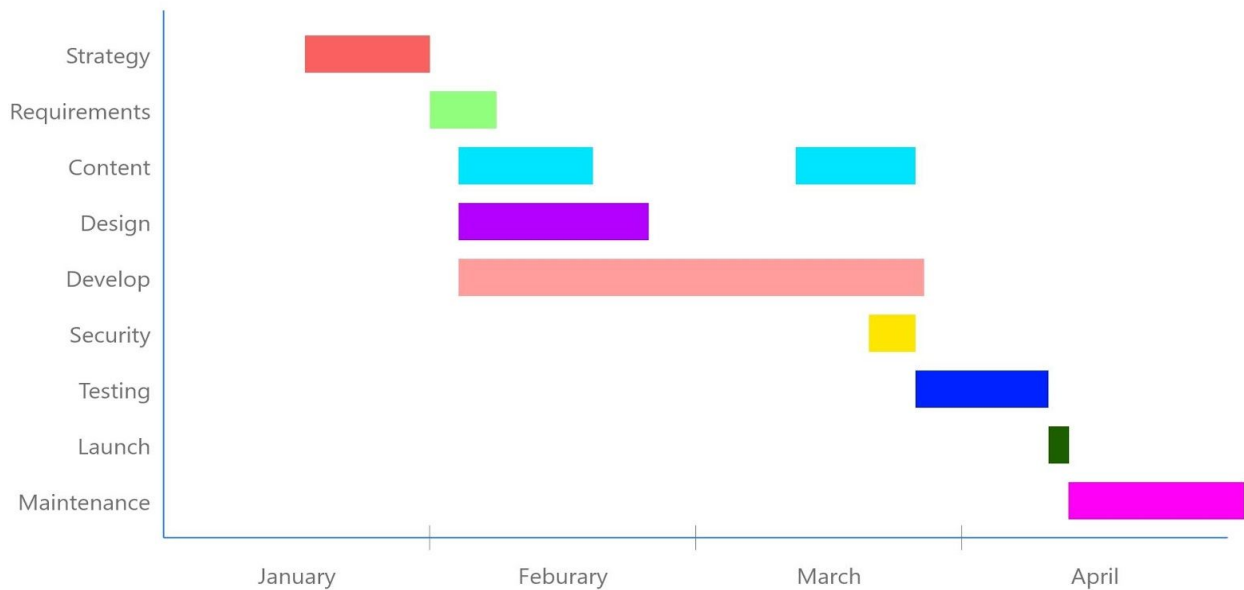
**UI/UX Designer:** In charge of design and usability for the web application, these designers will ensure that a consistent structure is used throughout the project.

**Database Architect:** This member is in charge and of designing and developing a database to store the web applications information.

**Full-Stack Developer:** The full-stack developer will be involved in both the design and development on the front and back end, as they will provide help where necessary.

**Project Manager/Team Lead:** The team lead will ensure all tasks are met as per the required criteria. They will make sure all tasks are completed timely, and within the set structure.

## Timeline



Section	Time Period	Activities
Backend	Jan 31 - Feb 7, 2019	Requirements
	Feb 4 - Mar 26, 2019	Development
Interface	Feb 4 - Feb 25, 2019	Design UI and UX
Content	Feb 4 - Feb 18, 2019 & Mar 11 - Mar 25, 2019	Writing the subject matter/content of the website
Security	Mar 19 - Mar 26, 2019	Make the website secure
Strategy	Jan 17 - Jan 31, 2019	Planning the development of the website from the beginning till the end
	Mar 26 - April 17, 2019	Testing the functionality and bugs in the website
	April 18, 2019	Launch the website
Maintenance	After website launch	Maintain the website by

		updating the code making the website secure
--	--	---

## Project Lifecycle Overview

1. Planning: 3 weeks
2. Design: 4 weeks
3. Development: 7 weeks
4. Deployment: 2 weeks

All the sections can be categorized within the following headings: planning, design, development, and deployment

Planning for the project will be broken down into 3 weeks, within this time we will find the main problems which our application will solve and how we will solve them. Our solutions to these issues will be solved via the features we develop. These features will be determined and distributed to the group members for design and development.

In the Design phase of the project, wireframes and mockups will be developed for the web application and used as the basis for development. The designs will be created with the intention of having a good user experience, while having an accessible, responsive, and clean user interface.

During the Development portion of the project is where we will begin to create our features, these features will be the solutions the Event Planner application will solve and makes up majority of the timeline.

Lastly, the Deployment phase will occur after the 3 previous stages, and will involve launching out Events application online, which will require hosting (which has been purchased by our team). The launched application will be tested for functionality, as well as security purposes to ensure a successful application launch.

# Communication Plan

**Development Server:** Localhost for development and paid hosting for testing

**Documentation:** Google Docs

**Version Control:** Git and GitHub