

EventGo!

Event on a Click

Ankit Luthra
Master of Applied Computing
University of Windsor
Windsor, Canada
luthr111@uwindsor.ca

Vipul Hodge
Master of Applied Computing
University of Windsor
Windsor, Canada
hodgev@uwindsor.ca

Khushbu Nakum
Master of Applied Computing
University of Windsor
Windsor, Canada
nakumk@uwindsor.ca

Shubham Mathur
Master of Applied Computing
University of Windsor
Windsor, Canada
mathurs@uwindsor.ca

Dr. Pooya Moradian Zadeh
School of Computer Science
University of Windsor
Windsor, Canada
moradiap@uwindsor.ca

Abstract— In today's world there are many events happening around us. But many times, people are not aware about the happening events and eventually tend to miss it. Events are posted on various social media platforms but there is no single platform where all events can be found. This project will solve that issue and provide the users with a user friendly platform where they can look at all the events around them and register to them as well as host new events. (Abstract)

provide organization with data on emotions amongst their members. Organizations can then use the results obtained to improve themselves and provides services for people who have negative emotional experiences like depression, stress, anxiety.

I. INTRODUCTION AND REQUIREMENT ANALYSIS

Humans are social animals, which means that they can interact with each other. People often like to socialize by attending events, but usually, they don't have any idea about the events taking place nearby. Many events fail to attract the audience as most of the people do not know about the event.

So, we came up with the project idea which helps our users to view, choose and register to the workshops/events of their choice and even organize one of their own. Our potential customers would be anyone and everyone who wants to attend or host events near them.

EventGo is a mobile application that makes the task of viewing the events and hosting it hassle-free. Through this application, people can view the events which interest them and register for the same. This application promotes various events displayed in the application and ensures that no event gets unnoticed.

EventGo fetches and displays events from Google and other social networking sites. EventGo allows the users to share the event with their friends. This application shows all the events from multiple platforms into a single application, which saves time for the users to search for the event. Organization. This will help organizations to understand the emotional and mental status of people. This project aims to

II. RELEVANT LITERATURE REVIEW

A. The Experience Movement

We have studied several papers based on event management. The first paper which we have studied is named "The Experience Movement: How Millennials are Bridging Cultural and Political Divides Offline," which was published by Eventbrite. In this paper, more and more live events are being attended day by day. Three out of four Millennials prefer experiences over things [1].

B. Event Management System (M. Mahalakshmmi, S. Gomathi and S. Krithika)

The second paper named as "Event Management System" which is published by M. Mahalakshmi, S. Gomathi and S. Krithika. The main idea of this paper is used to maintain the College Event information and organize the event and to send the user registration time through SMS with verification code to the user using mobile application. [2].

A. Definition

For this project, we would be following the Iterative process model. Iterative development is a way to break down a broad application's software development into smaller chunks. The function code is designed, developed and tested in repeated cycles in iterative development. Additional features can be designed, developed and tested with each iteration until a fully functional software application is ready for customers to deploy.

The aim of iteratively working is to make changes more flexible. When the conventional approach (sometimes referred to as BDUF or Big Design Up Front) conducts the specifications and design of a large project, there may be unforeseen problems that do not arise until production starts. The project team works iteratively through a cycle in which they evaluate each iteration, and determine what changes are needed to produce a satisfactory end product.

B. Specification

1) Functional Goals

- View all the events location wise in a user-friendly manner in a list as well as map view
- Users can host events and provide the details including the description and location
- Share the events details on multiple platforms like facebook, twitter, etc. through the application
- Invite people for the events if interested
- Users can set reminders for the events for which they have registered

2) Strategic Goals

- Through this application, people can find events related to all the fields (IT, sports, etc.) on a single platform
- Organizers can host the event through this application without much hassle. This application will help them advertise their events easily
- This will also make easy for people to socialize with similar background and mindset

3) Business Goals (time-to-market, cost)

- We will charge a service fee for each ticket sold of an event from the organizer
- Goal is to follow SDLC and complete the application including documentation in 4 months
- Divide the work between 4 resources so that the application can be completed on time and in an efficient manner

4) Technological Goals

- Using a technology ideal for IOS mobile application development using SWIFT
- Further, we can also create an android version for this using Android Studio which requires knowledge of Java, XML, JavaScript and php

5) Quality Goals

- Provide a user-friendly interface to users to view, register and organize events on a single platform.
- Application should always be available and without bugs
- Follow Software development lifecycle to ensure the quality of the project

C. Architecture

The VIP cycle:

View Controller -> Interactor -> Presenter -> View Controller

To help you isolate logic in your program, the Clean Swift architecture uses a VIP loop. The VIP cycle includes a ViewController, an Interactor and a Presenter. For some logic, all classes are responsible. The display logic is the responsibility of the ViewController, the business logic is the responsibility of the Interactor and the presenter is responsible for the logic of the presentation [3].

The image below shows the flow of the VIP cycle.

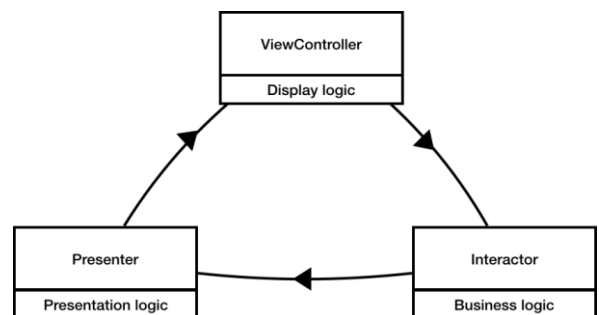


Figure 1. VIP CYCLE

Using the structure of Clean Swift, you know exactly which logic in which class should be placed. It makes the software more maintainable because you know exactly where the improvement should be made in your code when you need to fix a problem or want to add more features[4].

D. Platform

We have followed a client-server architecture for our project. Apple iOS is a proprietary mobile operating system running on the Touch iPhone, iPad and iPod. Apple iOS is based on the desktop and laptop computers Mac OS X operating system. The iOS developer kit offers tools for the creation of iOS apps.

E. Design

USE CASE DIAGRAM:

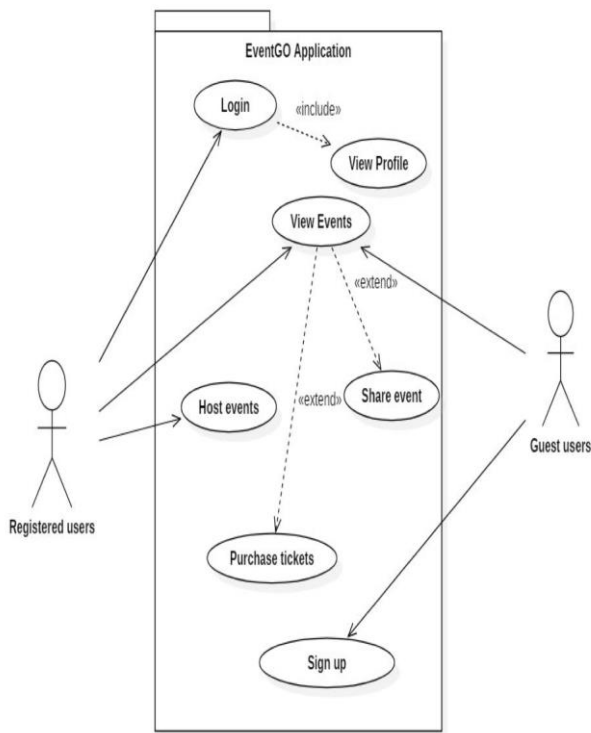


Figure 2. Use Case Diagram

DFD DIAGRAM:

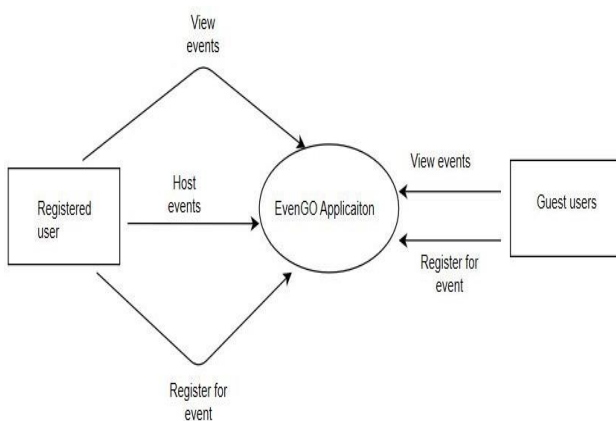


Figure 3. DFD 0

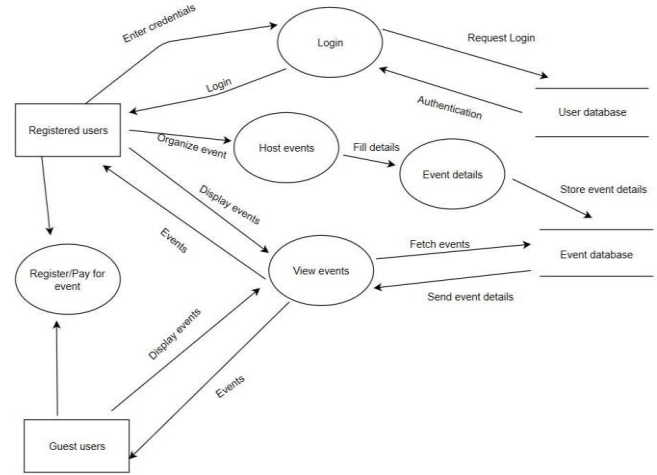


Figure 4. DFD 1

F. Implementation Details

- 1) **Xcode:**
EventGo application is developed using Xcode which is an IDE, an interactive development environment, developed by Apple for macOS, iOS, watchOS, and tvOS software development. It is the only officially supported tool to create and publish apps in Apple's app store, and is designed to be used by beginners and experienced developers [5].
- 2) **SQLite:**
For the backend system, we used SQLite which is a popular choice for local / client storage in application software such as web browsers as embedded database software. It is arguably the most widely used database engine, as it is used today, among other things, by several widespread browsers, operating systems, and embedded systems (such as mobile phones) [6].
- 3) **Cocoa Pods:**
Cocoa Pods is an application level dependency manager that makes it much easier to handle your design. We used Cocoa Pods to use various libraries in our project implementation. It can save you a lot of effort and time when dealing with your project's dependencies, as it makes it much easier to add, remove and update libraries [7].
- 4) **Google Maps:**
We used Google Maps API for Swift to show the nearby events on the maps. Google Maps is web mapping service developed by Google. Some of the main features of Google maps are Satellite imagery, aerial photography, street maps, 360° panoramic views of streets, real-time traffic conditions, and route planning for traveling by foot, car, bicycle and air, or public transportation [8].

IV. TESTING

S. No.	TEST CASE	OUTCOME
1	Error message is displayed after entering wrong input values. For creating the user screen and creating the event screen.	Pass
2	After creating the user, details should be added in the user table in the database.	Pass
3	User profile is displayed for the user who has logged in successfully.	Pass
4	User can see the events nearby on the Google maps fetched from Google Calendar API.	Pass
5	The details of the event are added successfully to the event table in the database.	Pass
6	Search bar is aligned properly and is able to search different locations.	Pass
7	Events can be shared on different platforms like WhatsApp, Facebook by clicking on the respective share button.	Pass
8	Users can see the created event fetched from database on the Google Maps.	Pass
9	All the buttons, textboxes and other widgets are aligned properly with a uniform layout.	Pass
10	User is able to receive notifications about an event when disabled.	Fail

TABLE I: TEST CASES

Software Testing is referred to as an activity to ensure that actual results match the expected results and the product is defect free. We have done the following testing in our project lifecycle:

- 1) Unit Testing:
We tested the individual models separately. For example, first we checked the Home screen of the application and tested whether all the text fields and buttons are working correctly or not. After that we checked all other modules separately.
- 2) Integration Testing:
We tested the application to check whether all the modules are linked to each other correctly or not. For example, whether the screen flow is correct after clicking the login button.
- 3) Functional Testing:
We tested the application to check whether its performing its functionality properly or not. In this testing we checked whether output is according to the requirements or not. It is a type of Blackbox testing in which we did not require the code to test the application.
- 4) Negative Testing:
We tested the application by providing incorrect input and observing the Behaviour of the application. For example, If we want to search for an event, we

deliberately provided wrong event name and observed the behavior of the application.

5) Branch Testing:

We tested every branch in the code and checked whether all the branches are giving the desired output or not. It is a type of White box testing.

V. CHALLENGES

During development cycle of this project we faced few issues. Some of those challenges are:

- We faced resource issues. We decided to make a native IOS application, but we realized that not all the members have access to MAC OS which is suitable for IOS application development. So, managing and organizing tasks was challenging.
- Everyone was not familiar with the technologies used. So, learning it and implementing it in time was a challenge.
- Everybody has a different level of experience so at times everyone had different ideas. So, finding a common ground was a bit challenging.

VI. CONCLUSION AND FUTURE WORK

This report summarizes the concepts we used in development cycle of this project. It consists of the concepts taught in class and how we used them to implement a real time software. Some of those concepts include:

- 1) UML Diagram
- 2) Software Testing
- 3) Data flow
- 4) Following Software Development Life Cycle (SDLC) properly.

Also, this report includes a literature review we have done for this project. We have implemented most of the essential features in this application which includes viewing the events and hosting it hassle free. But there is scope for further improvement in this application. We fetched and displayed events from Google. But, to increase the scope of the application we can integrate the application with Facebook, Twitter etc. so that we can fetch events from various other social networking sites as well. Also, we can use QR codes on the tickets which will make it easier for the organizers to scan the ticket and confirm its validity on the event day. For this, we will have to make another small application for the organizers which will be used just to scan the QR codes on the tickets.

VII. SCREENSHOTS



Figure 5. Cover Screen

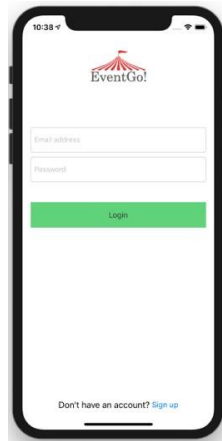


Figure 6. Login Screen

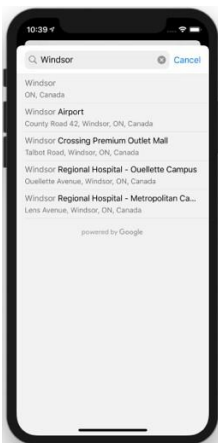


Figure 7. Search Bar

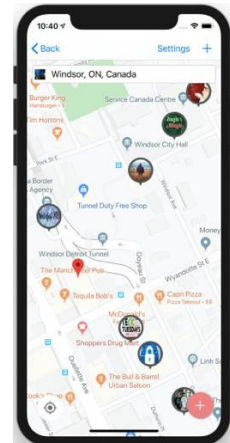


Figure 8. Events on Map



Figure 9. Event Selection

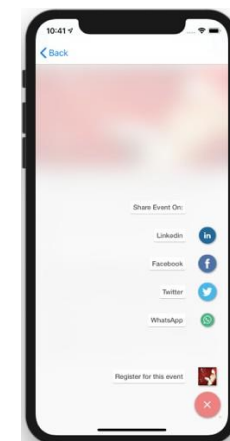


Figure 10. Event Details



Figure 11. Share Options Menu



Figure 12. Create Event Menu

VII. REFERENCES

- [1] (2019) Retrieved 27 November 2019, from <https://www.ijraset.com/files/serve.php?FID=12457>
- [2] "Eventbrite Research: Millennials Fuel the Experience Economy Amidst Political Uncertainty," Eventbrite US Blog. [Online]. Available: <https://www.eventbrite.com/blog/press/press-releases/eventbrite-research-millennials-fuel-the-experience-economy-amidst-political-uncertainty/>. [Accessed: 28-Nov-2019].
- [3] "Clean Swift, the architecture explained using diagrams," Jeroen Zonneveld, 20-Oct-2019. [Online]. Available: <https://zonneveld.dev/the-clean-swift-architecture-explained/> [Accessed: 28-Nov-2019].
- [4] Zullo, C., & Zullo, C. (2019). Clean iOS Architecture pt.7: VIP (Clean Swift) – Design Pattern or Architecture? – Essential Developer. Retrieved 26 November 2019, from <https://www.essentialdeveloper.com/articles/clean-ios-architecture-part-7-vip-clean-swift-design-pattern-or-architecture>
- [5] "Xcode - Apple Developer", Apple Developer, 2019. [Online]. Available: <https://developer.apple.com/xcode/>. [Accessed: 29- Nov- 2019].
- [6] "What is SQLite? Top SQLite Features You Should Know," SQLite Tutorial. [Online]. Available: <https://www.sqlitetutorial.net/what-is-sqlite/>. [Accessed: 29-Nov-2019].
- [7] C. P. D. Team, "CocoaPods.org," CocoaPods.org. [Online]. Available: <https://cocoapods.org/>. [Accessed: 29-Nov-2019].
- [8] "Google APIs Explorer | Google Developers", Google Developers, 2019. [Online]. Available: <https://developers.google.com/apis-explorer>. [Accessed: 25-Nov- 2019].