# GamePlan

**Matthew Yin**
Georgia State University

**Soojie Chin**
Georgia State University

**Amber Pradhan**
Georgia State University

**Raghuveer Gummadi**
Georgia State University

## Abstract

In this present era, social media attracts people to present their views [1] has a lot of impact in our lives. We all are well aware of the fact that social media has a lot of influence over people's mental Health like depression, dementia, schizophrenia etc. Although the usage of social media platform like Facebook, Instagram , twitter and software engineering together is not well understood, these mechanisms influence the software development practices. Software developers use and integrate into a wide range of tools ranging from code editing web-based portals . In our research project we would like to discuss about software engineering practices implemented in our project "GamePlan" . We used resourced data, "Bootstrap" and "Django" in our project. Using this data gives the way into utilizing the machine learning models and can be extended to deep learning methods such as CNN, AE [2] and in real live scenarios such as twitter analysis. We used the most compatible architectural model, which is "repository architecture" for our project.

# 1 Introduction

GamePlan is a website that allows coordinators to create and host events; either in the real-world, within a video game, or within a virtual meeting room. Our website allows for gamers to gather in one collected place to plan events and meetings.

Some of the competitors in this field are EventBrite and Meetup.com, but they are not focused solely on gaming events. EventBrite focuses on artistic and musical galleries whereas Meetup.com focuses on social gatherings.

Our system is made possible through the use of Bootstrap in the front-end markup development for HTML templates. The database language used for the back-end is Django, as well as the front-end functionalities. Django provides the website it's functionalities to receive and process information from the client server into the GamePlan database.

## 1.1 Figures

The figures shown in this paper represent the class diagram and the sequence diagrams for this project.

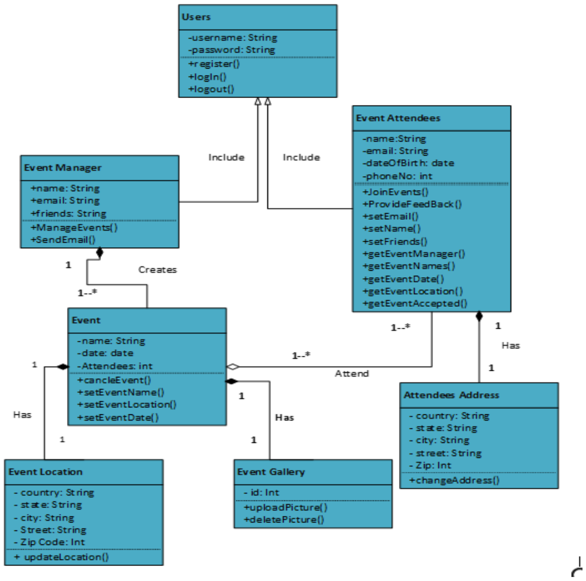

Figure 1: Class Diagram

The Class Diagram above (Figure 1) illustrates how the classes are structured and their relationships among other classes. This diagram shows the association of classes between Event Manager and event Attendees through the Event class.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

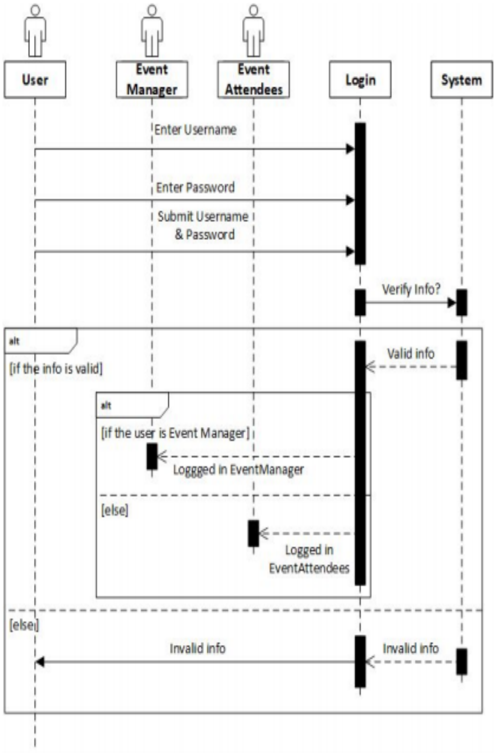

Figure 2: Sequence Diagram for Login() function

The Sequence Diagram (Figure 2) below illustrates the login function where the User must provide valid information in order to access their account. If the user does not provide valid information to login they will receive an error message on their display.

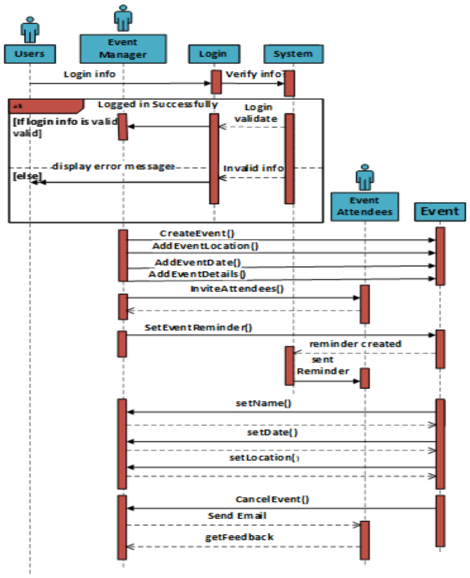

Figure 3: Sequence Diagram for createEvent() function

The Sequence Diagram (Figure 3) represents an Event Manager creating an event. The Event Manager has the option to create multiple events and choose to invite a limited number of attendees based on his requirements.

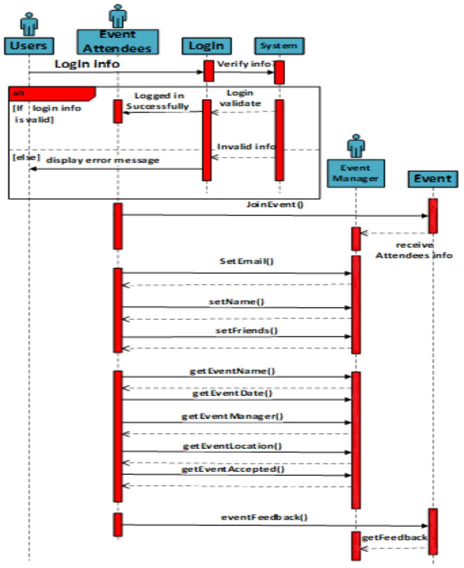

Figure 4: Sequence diagram for joinEvent() function

The Sequence Diagram (Figure 4) shows the necessary steps for a user to join an event. Any user has the ability to use this functionality to join the event and receive all of the event information.

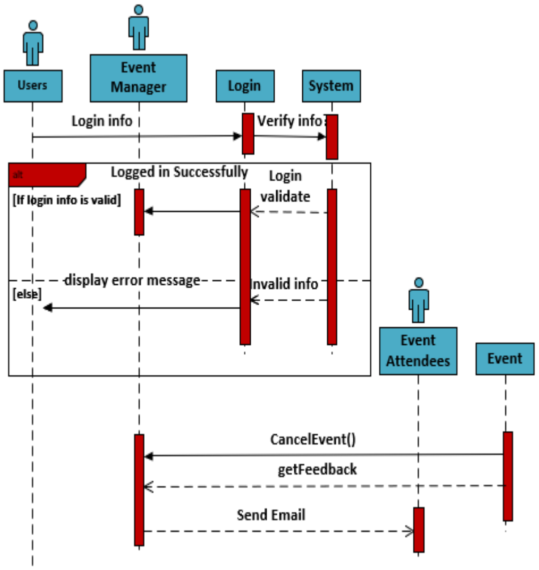

Figure 5: Sequence diagram of cancelEvent() function

The Sequence Diagram shows the necessary steps an Event Manager will need to do in order to cancel an event. Only the Event Manager has the ability to execute this function.

# 2 Architectural Model

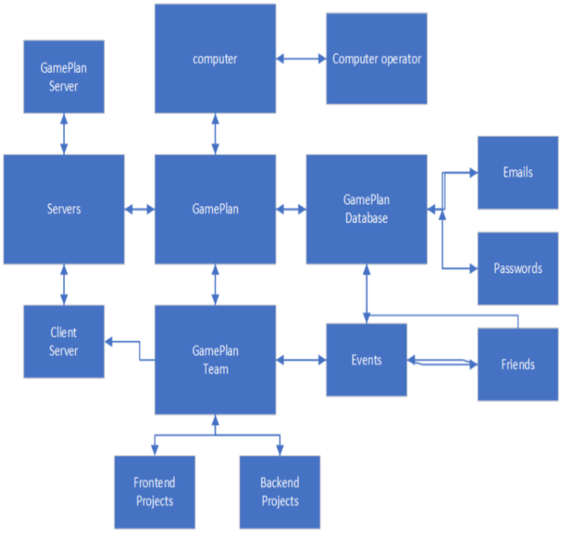

Figure 6: Architectural Model

The Architectural Model used for GamePlan is using the repository architecture. The model allows for developers to understand the desired model that was developed and how each component inside of GamePlan correlates with all other components. The main components as shown in the figure are the servers, database, the development team, and the use of a computer.

## 2.1 Logical View

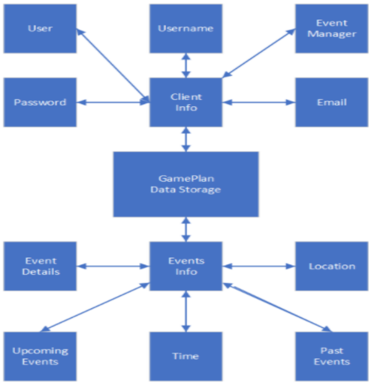

Figure 7: Logical View

The logical architectural view of Gameplan consists of vital data to the program. The information consists of user information and event information necessary to access all of the features of the website. The user information contains vital information such as name, username, password, and email when creating an account. The event information consists of information for the event including the details of what the event includes, location, and time of the event.

### 2.1.1 Physical View

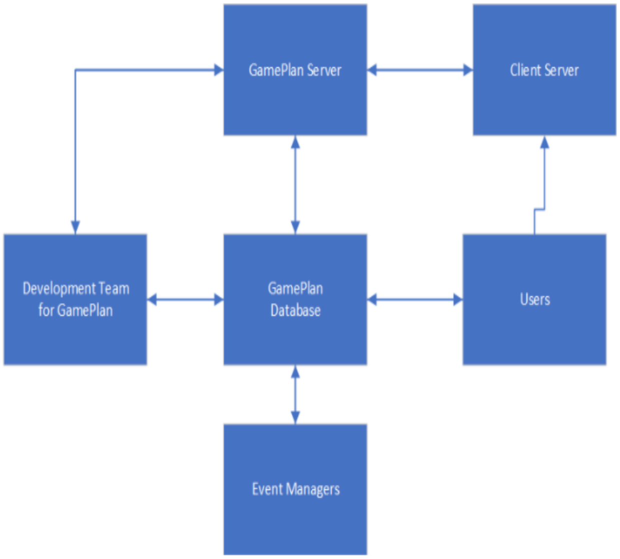

Figure 8: Physical View

The physical architectural view focuses on the database and servers for GamePlan. The information provided inside of the client server will be transferred to GamePlan servers and stored into the GamePlan database. The development team will be able to access both the GamePlan server as well as the database to make any necessary changes.

### 2.1.2 Development View

The development architectural view consists of the two central components to GamePlan. The front-end which contains the HTML templates of which a given user will view. The back-end contains the database in order for the website to have functionality. Both of these are crucial parts for the website to function properly for each user.

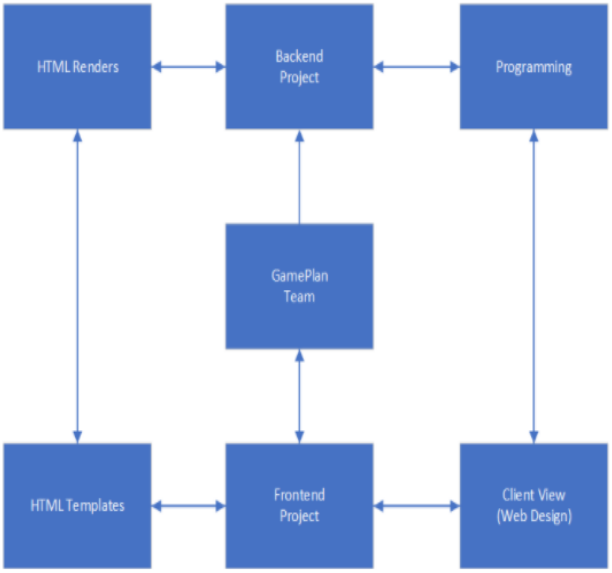

Figure 9: Development View

### 2.1.3 Implementation

A version of the GamePlan website can be accessed at https://se-gameplan.herokuapp.com/. GamePlan's source code can be seen in the following git repository: https://github.com/myin3/Software-Engineering.

### 2.1.4 Process View

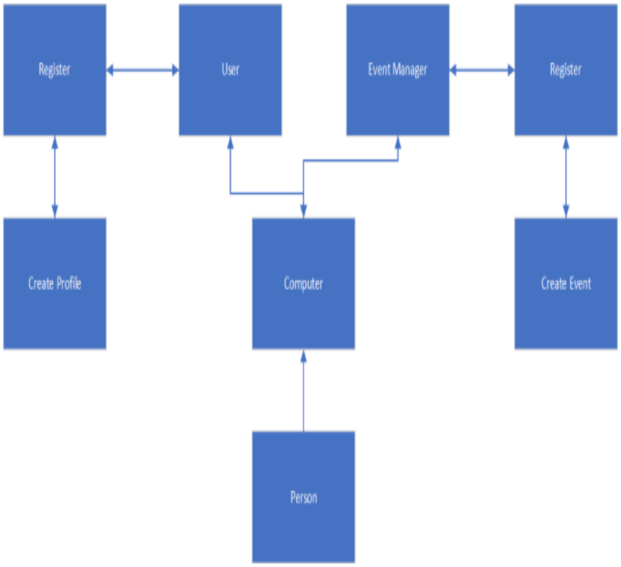

Figure 10: Process View

The process architectural view shows how a user will be able to create a profile inside GamePlan's database. The process to creating a User profile only consists of using a computer and providing the necessary information that is asked of in the register page. Event Managers will follow the same process of creating a profile, but where they are different is the fact that event managers have the intent to create an event, therefore they will need to create a event.

**Understanding** Throughout the testing and process of GamePlan, we came to an understanding that tests should be written concurrently in order to prevent delay. Functionality and test cases heavily go hand in hand, so the writer of the functions should also write test cases or the team should work simultaneously to prevent confusion. In order to prevent any confusion, clear and frequent communication should be practiced within the group. Apart from unforeseen complications within group members, the group noticed scalable cloud platforms allow convenience and aid visualization of the progress of the project.

### Conclusion

Conclusively, with further resources, GamePlan could potentially become more interactive on a large scale base. The team worked successfully as a group, maintained clear communication, and gave their best to meet personal deadlines. It allowed for a minimal amount of complications and quick feedback and alterations before the official deadline. A special consideration of Raghuveer Gummadi for using his prior experience to assisting with a substantial portion of the programming on the back-end.

### References

[1] S. T. Sadasivuni and Y. Zhang, "Using gradient methods to predict twitter users' mental health with both covid-19 growth patterns and tweets," *second IEEE International Conference on Humanized Computing and Communication with Artificial Intelligence (HCCAI 2020) September 21-23, 2020 Irvine, CA, USA*, 2020.

[2] J. K. Mandivarapu, B. Camp, and R. J. Estrada, "Self-net: Lifelong learning via continual self-modeling," *Frontiers in Artificial Intelligence*, vol. 3, p. 19, 2020.