

# **EVNT: Night Planner**

## **Final Documentation**

Group 10:

Thai An Luong, Yash Gupta, John Krupa,  
Andrew Kwan, William Reil

Group Project - Final Documentation

CS 3307

Prof: M. Katchabaw



# 1. Project Summary

EVNT Party Manager is a product that has been designed to fill an existing need in the social media app marketplace. Over the last couple of years, public opinions and views on Facebook have decreased significantly. Especially among groups 30 and younger. Despite this shift in public perception, Facebook has still remained the primary group activity or event organizing app of choice for both individual hosts and event spaces. Several clubs, bars and restaurants around Western are still using Facebook as their event manager of choice where they upload future events in order to attract patrons. We believe that with our app by combining key aspects of Facebook, BeReal, and Snapchat we can create our own distinct party/event management and social media app which would serve to fill a void in the existing app marketplace. Our app would first and foremost allow users to create a calendar with events and be able to schedule their own events with friends. The Events List feature of the app, also allows the user to browse for parties or events being hosted by their friends.

To create the app itself, we used C++ in combination with the WxWidgets toolkit to allow us to make an interactive and easy-to-use graphical user interface. As per the original project requirements, we also made the project Linux compatible. The app uses a database hosted by Back4App which allows all versions and installations of the app to share a single database to ensure that all users can access all events, and that once a profile has been made, it can be accessed through any device.

As per our original specifications, the users have the following options:

- Create a new account
- Reset password during login (Forgot password)
- View List of events (User can view events being hosted by their friends)
- Search other users (Users can search for other users and choose to follow/unfollow them)
- Change user details (Users can change their own details such as password and date of birth through the app once logged in)
- View user profile (Users can view their own profiles displaying their details as well events they are hosting)
- Create new events

Our original specifications listed a couple more services which we were unable to implement at this time, but, attributed to the modular structure of the project's design, additional features can be designed and implemented with ease.

## **2. Key Accomplishments**

### **Overall Accomplishments:**

Overall, we created a functioning app that allows users to access events, and other user profiles to explore the events they want to go to. Our app has a functioning login system with the ability for new users to create accounts update their information and even change their password through security questions. They can make their own events and everything is shared online across our online database. All of this is done from our easy to navigate UI menu system that allows for modularity and extensibility. Our project was successful in accomplishing its goals and ensure room for future development and expansions.

### **Backend Accomplishments:**

Going into this project, we knew that the direction we wanted to go would require some form of cloud storage or online database, which we were able to successfully use and integrate into our project. With none of us having used cloud databases before, it was a challenge to find a solution, which made finding such a solution a very big accomplishment for us.

We had one group member look into several solutions, who eventually settled on Back4App. After figuring out how to modify and query from the database, we were able to create tables for information we needed to store, then eventually create database handlers which provided a very simple interface for other classes to use for storing information in the database.

After creating working handlers and understanding how to read and write from the database, it made it very easy to add new methods to the database handlers as more functionality was required, and very easy to modify as the project moved along. Having one member responsible for maintaining and modifying the database handlers made it very easy for the rest of the group to integrate it into other parts of the project, not having to worry about the implementation of the database handlers.

With database integration and data sharing being such a vital part of our final application, it made interacting with a database that much bigger of an accomplishment.

### **UI Accomplishments:**

Throughout the software development process, our team made significant progress with the UI. We worked to design an appealing and functional UI that would enhance the user experience. Despite the challenges we faced, such as the lack of documentation for the older UI tool, we were able to overcome these obstacles by using online communities like Stack Overflow for guidance.

Our team's decision to create a modular UI was another critical factor in our success. By designing the UI to be modular, we were able to make changes to individual components without affecting other parts of the software. In the end, our team was proud of what we accomplished with the UI and how it contributed to the overall success of our project.

### 3. Key Problem Areas

#### What Went Wrong:

##### Designing:

During the planning phase, we first had high expectations for the software at the beginning of the design stage, which led to overplanning and creation of more functions without being aware of the tasks' difficulty. Such features include:

- Adding advertisement
- Adding an embedded map
- Push notifications

##### Coding:

We started the coding process before completing the design stage to give ourselves as much time as possible to execute our ambitious goals. The original plan was to make every object a class to store private values and then use object oriented programming concepts to connect classes. This plan changed once our cloud database handlers were created, which drastically changed our plans and ideas.

#### Processes That Did Not Go Well:

##### GUI:

The project GUI was created with the wxWidget libraries, which allowed us to compile/debug on different operating systems. However, the wxWidget default GUI differed depending on the OS being used. This difference made it difficult to perform any kind of design work on different operating systems, forcing everyone to resort to using a virtual machine to ensure we were all on the same page. In addition, wxWidgets was an older library, and had a limited amount of documentation, making solving problems associated with the wxWidgets a very length and difficult process.

##### UML Diagram:

We had to change/remove many of the plans and functions in the UML diagram through phases due to the non-use of classes and the addition/removal of other features. This forced us to have to refactor the UML diagram and code simultaneously because the plan before the final stage was no longer effective, leading to a lot of work near the end of the final stage.

#### Technical Challenges:

##### Preparing for Development:

Installing all the libraries and setting up Linux virtual machines stalled the group's process. The issue with VMs was the limited processing speed, so installing essential tools and performing regular coding tasks made the development process a lot more lengthy.

##### Project Setup:

The limited documentation of wxWidgets made setup very difficult. We searched through many different sources, all of which gave different solutions with varying levels of

success. Since we were not able to start development until setup was complete, finding a setup solution was a very lengthy and frustrating process that bottlenecked our development.

### **Decisions That Led to Difficulty:**

#### Adding Images (Coding):

The project's base idea was social media, in which pictures and photos were very desirable for the software. The wxWidget library required some basic knowledge of using sizers and bitmaps to add images which our group had to spend a lot of time learning, limiting the number of image features we were able to include.

#### Image (Database):

Storing images on a cloud database required compression, and the average size of an image itself was 10MB (JPEG) and 5MB (PNG). The cloud database for this project had a size limitation for each read and write, along with a limited amount of file storage, so pulling a large number of bits was not possible.

### **Effects of Problem Areas:**

#### Coding:

A lot of the project relied on our group knowing the wxWidgets library well, which took a very long time for all of us to get accustomed to. The amount of time we had to spend debugging simple issues really limited our productivity and often took our focus away from the big picture. This was made even more difficult with the amount of overhaul we had to do from the first stage in order to make our GUI more modular and easier to work on.

### **Corrective Actions:**

#### Considering Technical Limitations:

Having not considered our GUI tool of choice until we began the coding process, we really did not understand what features we would be able to implement into our final product within our limited time frame, so it would have been more beneficial to do more research into GUI tools before we started our project so that we could have planned around that, and ensure our final product would be more like our original plan.

## 4. Lessons Learned

The project provided our team with a valuable learning experience that yielded several key lessons and takeaways. Throughout the semester as we learned more about the project and each other and as we worked on the project we became more cooperative and thus more successful. One of the most significant lessons and also earliest lessons for us was the importance of communication and coordinating our technology. Selecting the right technology for the project was a process that took longer than anticipated due to poor in-group communication about requirements and needs. Because of this we started on work on one UI library but ended up needing to swap part way into making our project to another front end framework. Swapping quickly to the new UI also provided difficulties too, we found that working with an older UI tool like wxWidgets meant we had limited documentation that created challenges and delays in getting minor graphical bugs fixed. Going forward, we would prioritize researching and selecting all of our group managed technologies well ahead of time. This would give us more time to familiarize yourself with the technology. Furthermore if we had researched the technology more we could have found a more modern alternative with ample documentation and community support to help mitigate potential issues.

Another area where we encountered difficulty as a group was with our Project's initial structure, initially our code was almost entirely in our mainframe class. However especially when working on git and transferring files between each other we found a lot of difficulty when all working on one file. We learned the importance of modularity in the design process and eventually separated our project into different sections. This facilitates us each working individually on parts of the assignment, and then bringing those windows together for the final design. By creating a modular UI design, it allowed us to make changes to the UI without affecting other parts of the software. This also further proved useful when we needed to make changes to the project scope, which we had to do to address the technical challenges we encountered.

When working through our project to try and meet all of our goals we quickly found that we needed to start decreasing our original scope for the project. There were several features which in hindsight should've only ever been on the 'Wishlist' for completion but we initially thought they were feasible but later realized we would not be capable of. If we had better researched and planned out our technologies we would have had a better idea of the scope we could design for going in.

As the project progressed, our team recognized that effective communication and meeting scheduling were critical components of a successful project. We realized that getting everyone together for a meeting was not only challenging, but it was also counterproductive if we were not well-prepared. Therefore, we created a best practice to keep us on track by deciding the next meeting time at the end of each meeting and posting it on our group chat. This approach helped us stay organized and on track. By deciding on the next meeting time at the end of each meeting, we were able to avoid the confusion and ambiguity that often arise when scheduling meetings. We also found that this approach helped us stay accountable to each other, and it helped us prioritize our work more effectively. Moreover, by planning ahead for our meetings, we were able to set specific goals and objectives, and had a clearer idea of what needed to be accomplished. We found that this approach helped us make the most of our meeting time, and allowed us to focus on the most critical tasks at hand.

In conclusion, the project provided our team with a rich learning experience that helped us cultivate important skills in software development. We learned how communication and coordination are vital to the success of a project, and that selecting the right tools for the project is crucial. We also learned the importance of modularity in the design process, which allowed us to work more efficiently and make changes without affecting other parts of the software, while keeping the door open for future improvements. Moreover, we learned the value of planning and scoping, as it can help avoid overcommitting and under-delivering. Overall, the project gave us an insight into real-world problem solving and allowed us to come up with solutions that actually solve something and are applicable. The project gave us the experience of the ups and downs of working on software development as a team and left us wanting to work on similar projects in the future.