Gainesville To-Do

David Espantoso
despantoso@ufl.edu

Marco Cardodo
mcardosog@icloud.com

Roberto Garcia
r.garcia1@ufl.edu

Roberto Pantoja
rbrtopntja@gmail.com

INTRODUCTION

The city of Gainesville, home to the University of Florida, houses thousands of students, faculty members, and local families. Local residents and tourists often tend to find themselves in repetitive plans due to the lack of knowledge of other activities. Gainesville-ToDo, an interactive web application, aims to provide a resource for these individuals that informs them of events and sights all around Gainesville. Our goal is to notify the people of Gainesville of all the amazing activities that there are to do in Gainesville, ranging from great night life to having a nature walk at Sweetwater Wetlands Park, breaking the endless repetition of the same events week in, week out.

USER REQUIREMENTS

Based on the results of our online questionnaire, the user requirements for the Gainesville-ToDo app are mentioned below. However, before requirements we must first establish the users of the application.

User Types

1. **College students**: these users are in their late teens to early twenties, have vast knowledge of basic technology, and have interest in all fields (interest depends on individuals rather than the group).
2. **Local residents**: these have a wide range of ages [children to adults], may know some technology or none at all, and have interest varying fields depending on age.
3. **Gainesville visitors**: part of the Gainesville community for a short burst of time, mainly come to visit family members and are interested in short term events.

Based on the 4 kinds of user types, the app will have two different modes: seeker and provider.

- The seeker's main goals are to find events catering to their interest and add them to their profile to see at a later time. As all interests are not the same, these users will have the opportunity to pick events based on their desired hobbies and save them on to their own personal profile.
- The provider will provide information on specific events going on around the Gainesville community or at a specific organization.

Functional Requirements

As a seeker, I want to create a personal account so that I am able to display my preferred events.

- This requirement requires both an account creation and login process with stored data.

As a seeker, I want to see all current event details so that I am able to gage my interests regarding specific events.
As a seeker, I want to be able to favorite preferred events so that I am able to see only those events that interest me.
As a seeker, I want to be able to remove events from my profile so that I may have an updated list of events.

As a provider, I want to create a personal account so that I am able to provide information regarding my events.

- This requirement requires both an account creation and login process with stored data.

As a provider, I want to be able to add new events so that seekers may be up to date with events around the community.
As a provider, I want to be able to delete events so that cancelled or past events may not show up on the seekers end.

Data Requirements

For this application, the data will be stored in a database called Firebase. The data is to be stored in two separate collections: seekers and provider. Inside these collections, the data for the individual users is stored based on user ID and will contain the following information:

- For seekers: account username, account password, name, preferred list of events.
- For providers: account username, account password, company/organization name, events.

With in the events field, providers will be able to insert: event location, event start time, event duration, event description, and a picture for the event.

Context of Use Requirements

In order to be able to use the application, the user must have a smart cellphone or computer (laptop or desktop) with internet connection. Given that it is a web application, the app may be used anywhere with internet connection. However, as the app targets the Gainesville community, the events on the app will solely be based around Gainesville.

System Experience Requirements

In order to accomodate for all technological backgrounds, the landing page will provide users with all required steps needed to proceed through the application and may be revisited at any time.

INITIAL DESIGN

The main goal when creating the low-fidelity prototype was to make a design that was simple and streamlined. Our low-fidelity prototype would serve as the blueprint for the high-fidelity prototype. When creating the high-fidelity prototype, the team decided that flooding the user with a million buttons and actions would lead to an unpleasant experience, therefore, we strived to keep the prototype minimalistic. In both our low-fidelity and high-fidelity designs, most pages only have a couple of elements so that the page never feels overwhelming or confusing. As for peer feedback, we did not get much negative criticism which made it difficult to change much about the final design, however, it did assure us that we had created a good overall design. Most reviewers liked that our design was kept simplistic and highly functional. Taking this into consideration, we noticed that in the "New Event" page we had a lot of unused space, so we split the page into two vertical sections that each have important inputs. We felt this made the page feel less cluttered, which creates a more positive viewing experience for the user. There was one comment which we felt we could not make a change on, not because of the content of the feedback, but because of unclear wording. The reviewer stated, "I really do feel that the log should be its own page," but we were unsure what element of our project would be considered the "log" and therefore could not change our site to incorporate their feedback. The team ensured that the design fulfilled the user requirement by ensuring it meets the needs of the seekers, who want to find events, and the providers, who create events. The seekers can easily find events by searching by category or by event name. We also provide the ability to say that they are attending the event if they create an account and log in, although this is optional. Providers can create events by logging in and navigating to the new event page. This setup also allows the providers to search through events to find one to attend. Conversely, a seeker can easily become a provider as long as they have an account or make one.

IMPLEMENTATION

Our application was developed using NodeJS for the backend and React for the front end. Regarding the database implementation, we decided to use Firebase from Google. The reason for the selection of these technologies was the flexibility that these platforms offer rapid development and good responsiveness for mobile devices. Among the features that we planned to implement in our project were:

- Register users
- Users can create events adding information like date, time, description, category, and images.
- Display events organized by different categories and most recently created.
- Users can mark if they will attend the events.
- Filter events by category, keywords or popularity.

The back-end implementation was designed using React controllers to maintain a better organization, allow easier modification of the code and increase code reusability. The controllers created were: account, firebase, new event, event page, home, landing, password management, session, sign in and sign out.

The core of the back-end was developed in the Firebase implementation due to the heavy traffic that was required in order to feed all the data to the interface. To improve stability and error handling we created multiple functions to handle the queries. The functions allowed the front-end functionalities like getting user information, filter events by creators, category or the top most popular events.

The successfully implemented features were registered users, create events, filter events by category, display and organize events. The function that was not implemented was to filter events by popularity. With the functions created in the backend the information required to implement this task in the future.

The front-end styling was done through a front-end framework called React Bootstrap.With Bootstrap different components like buttons, jumbotrons, and forms were imported and then styled to match our high-fidelity prototype. Furthermore, in order to implement our carousel's where events would be displayed, we imported another styling aid called react-slick. Using react slick allows us to display a continuous array of events based on those in our database.

For most of the controllers, a front-end page was created that illustrated the data stored in the back end. Those components that dealt with the manipulation of the database itself (firebase and session)

- The **landing front-end page**, illustrated the first view the users would have upon entering the site. This view was made up of a jumbotron with an image of Gainesville on the back, the title of the website, a sign-in button, a register button, and two sliders. This page will allow the users to browse the events without having to create an account. However, if they'd like to save their events they would need to create an account.
- The **home front-end page** illustrated an identical view of that seen in landing but with the user registered and signed in. Rather than just viewing the events, those users who create an account and sign-in, will be able to add events to their profile and keep track of their favorite events.
- The **sign in front-end page** presented the user with form boxes to enter their account information. In the first box, the user will insert the email address they used when creating their account. In

the second box, the user will insert their password for the account. Moreover, once the user inputs their information, they may click the login button and go into their account.

- The **sign out front-end page** is utilized in the home page and allows the user to sign out of their account once done. This component is implemented as a button inside the jumbotron for the home page.
- The **account front-end page** displays the users account information along with their stored events. On the top left of the screen, the user's name as inserted during registration will appear. On the top right of the screen, a button utilized for creating new events is displayed to the user and upon click will redirect them to the new event component. Below the top, the two sliders (one with created/attending events and one with suggested events) appear for the user to view. If the user clicks on one of those events, they will be redirected to the event page for that event.
- The **new event front-end page** displays the different form boxes the user may use when creating their new event. These form boxes include the name of the event, time for the event, place of the event, date of the event, categories for the event, description for the event and a picture for the event. Once the user has completed all the information, they may click the create button and their event will be added to the database.
- The **event front-end page** displays all information of the event based on data from the database. This information includes the following: title, location, date, time, categories, description and picture. Furthermore, inside this page the user may select whether they will be attending the event to save it to their profile page.

Although most of the features have been implemented on the front-end pages, there were some features which the team was not able to comple. Below we list those features not yet implemented.

- Home Page and Landing Page search bar: although we have the skeleton in place for the search bar in each of the two pages, yet, the connection between the skeleton and the back-end function has not been implemented.
- The attending button in the event view page: the button utilized to add the events to the user profile has yet to be implemented. Thus, neither has the functionality of adding this event onto the user page.
- Categories front-end page: as a group we were not able to complete the individual front-end pages for each of the categories. However, the event do have a categories feature that is utilized during their creation.

EVALUATION PLAN

For the evaluation plan, our team decided to evaluate the system using a formative evaluation. Due to the challenges and hardships brought by the Covid-19 pandemic, the system is still not fully completed, henceforth, our decision to use a formative evaluation on the design. The specific goals of the system is to create satisfaction and fewer problems when scheduling what events they will attend in Gainesville. Questions that we would like to be answered rely on the efficiency that our application affords them and if the current application serves as a blueprint for future online planners. The team decided to conduct a quick and dirty evaluation technique, where we can gauge the system's attractiveness, as well as if it satisfies the user's needs. The reason for this decision was due to the fact that the technique provides low cost, qualitative data. Evaluations will be run by intimate family members of each group member in an authentic work location. The system allows for this evaluation technique to be performed as it actually mirrors how our system will be used in the real world, because of the nature of the system. The system is for residents of Gainesville to plan and attend different events that occur in their city.

For the hypothesis, the team is trying to identify what level of satisfaction the application serves users and how a planner leads to fewer conflicts faced with scheduling events.

Primary hypothesis: By reducing the information displayed to users and feeding them with information they want to see, user satisfaction will increase. By allowing users to pick what events they will attend, users will be faced with less problems in event scheduling.
Secondary hypothesis: If I display the most recent events by date, users will be able to schedule events faster. If users have the option to view what events they are attending, users will be able to attend more events.

The method of choosing was quick and dirty. By using this method, we will inform users of our evaluation of the idea behind Gainesville To-Do and allow them to use the application. The objective of using this method is to receive quick feedback that will create valuable viewpoints of our application.
Metrics:

User satisfaction of application
Value created by the application for users
Understandability of the application
User's engagement with application

User’s retention rate to continue using the application

COMPLETION

We were able to complete a lot of the back-end functionality of our project. There are functions for everything needed to manage user accounts like creating a user, signing him in, and others. We also have methods for uploading and retrieving images. Our last main back-end functions are for getting the most recent events and for creating new events. In the front end, we have created the landing page, which holds cards for each category and some other cards for recently made events. When these events are clicked on, they redirect the user to the event page. On the landing page, we also have two buttons that allow the user to either sign up or sign in. We also have a search bar, but we have not implemented its functionality just yet. We also have a page to create events. Finally, we have the home page, which is where an user is redirected to after they sign in. It is similar to the landing page, but instead of the two sign in and register buttons, it only has one sign out button. We still must implement the pages for each category, which allow the user to search for events within a certain category. We also have to create the user page and implement user-specific features, like allowing them to say that they’re going to certain events. Overall, we are more than halfway done with the project and I would say we are about 80% done with the project. We should be able to finish this within the next couple of days.

VIDEO LINK

Link to system video: https://www.youtube.com/watch?v=xYvlc1djFAY&feature=youtu.be