CEN 4010 Principles of Software Engineering, Spring 21, 2021

Milestone 1: Team Project Proposal and Description

Alpha Technologies, CalendarConnect; Group 4

Gianluca: [gguagliardo2017@fau.edu](mailto:gguagliardo2017@fau.edu); Leon: [lrice2019@fau.edu](mailto:lrice2019@fau.edu);

Mayte: [mramirezcald2018@fau.edu](mailto:mramirezcald2018@fau.edu); Hendrien: [hpaul2017@fau.edu](mailto:hpaul2017@fau.edu);

Charlton: [cbenting2019@fau.edu](mailto:cbenting2019@fau.edu);

|  |  |  |
| --- | --- | --- |
| Revision History Table: | 02/16/2021 | Waiting In for instructor feedback... |
| Revision History Table: | 04/16/2021 | Revise Data Definition, High Level Requirements, and Non-Functional Requirements.  Add Vertical Prototype |

# **Executive Summary**

Given the current times, we wanted to create an app that will help people feel connected even during these socially distanced times. Calendar Connect is a personalized web app that can be tailored to the needs of each individual user.

The main focus of Calendar Connect is to allow people to stay connected with their friends, family, coworkers, and loved ones. Our app integrates a person’s schedule and allows them to share and connect with others to provide a clear visual of people’s availability. Our user-friendly dashboard can be customized and organized to the user’s needs and includes features such as to-do lists, groups, favorites, upcoming events, messages, alerts, and more! Calendar Connect allows users to create groups to share calendars with so you can maintain connection between family members, friend groups, school peers, or work colleagues. Our integration of pictures throughout your calendar month and year allows the app to become more personalized and for users to have fun and express themselves creatively. When you have an upcoming class, meeting, study session, or need some time alone, Calendar Connect can relay that to others to make sure you are not interrupted. On the other hand, when you are looking for free time within a group to spend some time together, whether virtually or in-person, Calendar Connect does the work of

finding free time for the users so they can focus on planning and enjoying the time together.

Unlike most apps that are tailored specifically for just work, school, or businesses, Calendar Connect offers a large reachability that can be used by everyone for all sorts of needs. Calendar Connects integration allows users to customize their features to fit their needs either for school, work, personal life, or all of the above.

# **Competitive analysis**

|  |  |  |
| --- | --- | --- |
| Google Workspace | Calendar.com | CalendarConnect (Us) |
| Smart Scheduling | Integration of other calendars | Integration with other calendars |
| Instants messaging between team members | Calendar analytics; a breakdown of all type of meetings | Instant messaging between contacts |
| Shareable/team calendars | Artificial intelligence/Machine learning | Group/shareable/team calendars |
| Video conferencing | Smart scheduling (overlay of who what time is available for selected contacts) | Smart scheduling (overlay of who what time is available for selected contacts) |
| Personal member sites | Team Calendars | Automated system to find availability between selected contacts |
| Custom email for your business and includes collaboration tools like Gmail, Calendar, Meet, Chat, Drive, Docs, Sheets, Slides | “Anyone” can book a meeting directly with you | Customizable calendars |

Summary:

Unlike most of our competitors, who are focused on enhancing the scheduling of business operations, our product will be more gauged more towards the general public and working adults.

# **Data definition**

**Database elements:**

* User: id (string), name (string), email (string), phone number (string), dob (date-time), password (string), registration date (date-time)
* Photos (no larger than 16mb): data (Buffer), contentType (string),
* Calendar (json)date

**Maps/Dictionaries (Hashing):** authentication of user information, database security (obfuscation)

**Heap sort (sorting algorithm):** to sort users contacts, events by importance, and messages

**Recursive substring search (searching algorithm):** to find an event, contact, or message/note

**User:** defines the user entity – unique id, name, email, password, phone number, DOB, profile picture

**Event:** defines a calendar event – unique id, event name, time, location, notes, attendees, color

**Messenger:** defines the real-time messaging app users can use to communicate with each other via our site – sending and receiving messages.

**Profile:** defines the user’s profile.

**Calendar Info:** defines an event, as well as what users can access the calendar and defines separate calendars based on color

**Calendar:** defines type of calendar – unique id, calendar name, group calendar (Boolean), viewers, calendar admins, public/private, events, color, calendar view

**Widget**: defines the type of widget available widget the user can add to the dashboard – widget name, location (on the board), delete widget

**Calendar(cannot-be-deleted)/Note/Weather/Messages/Contacts/Favorites/Photo slideshow/time/ widget:** children of the widget class

# **Overview, scenarios and use cases**

During these socially distanced times, many people still continue to balance their work life with school, extracurricular activities, meetings, hobbies, etc. and have found it difficult to still feel close and communicate with their friends, family, and loved ones. With differing schedules, meetings, and a global pandemic, keeping track of one’s free time as well as others can be difficult. With Calendar Connect, users can input their schedules from work, school, and everything in between into a single app and connect with their friends, family, and loved ones to share and view their schedules.

Our user-friendly dashboard is suitable for people of all ages and expertise levels given that our default template includes the key features most important to the functionality of the app. Our dashboard includes a monthly calendar showing all your events and tasks, a notes section, a to-do list, a list of your favorite connections, a reminder of upcoming events, any pending requests, and the ability to add events and contacts. Our spatially organized and labeled dashboard allows users with minimum technological experience to still enjoy and use the key features of the app. For those who wish to personalize their dashboard, Calendar Connect allows users to customize and add widgets to their dashboard based on their preferences or needs. These include but are not limited to a weather section, time zones, external links to specific zoom meetings, google hangouts, etc. and the ability to create groups and switch between shared group calendars.

A beneficial and key feature Calendar Connect offers is the ability to create groups with people you connect with to share calendars within friends, families, peers, or colleagues. This allows you to keep track of your personal and professional life separately all in one place. Within these groups, Calendar Connect will find free time within all corresponding members in case they want to meet virtually or in person, work on a project, or simply have time to catch up. Messages can also be created within groups to make communication for planning events easier. On the other hand, when a member of a group is extremely busy or simply wants some alone time during a mutual free day, they will be able to edit that in the app so that no one attempts to plan time with them.

For users who enjoy expressing their creativity and personalizing their calendars, Calendar Connect allows users to input pictures as memories for certain events or days to allow users to free up space on their personal devices while being able to look back at their week, month, or even year in pictures. Because Calendar Connect offers the ability to create groups, users can have memories on their personal calendar while also including separate memories in their group calendars with friends, families, peers, or coworkers.

# **High-level functional requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Functional Requirements | Must/Want | Comments |
| FR01 | The website shall have a home page that lists the purpose of the organization. | Must | Top left will say “about the project” |
| FR02 | The website shall have a hub for all users to access their calendars and meetings | Must | This will be the user’s dashboard |
| FR03 | The website shall have a link to each of our members where visitors can review our names, project priority and short description | Must | Each link will shortcut to each member where their information will be available to the visitor, name, contact info, short description |
| FR04 | The website shall have the ability to sign in with Google and Facebook, including our personal login as well. | Must | Most users prefer to log in with Google and Facebook due to time saving, accessibility, and security, if this does not matter then the user can log in through us. |
| FR05 | The website shall have the ability for users to add new friends who also have an account, this interaction will be customized depending on the privacy restrictions that each user permits, and denies. | Must | Users might want to add all their friends but not disclose all their information without consent, users will be able to customize privacy settings for each friend. |
| FR06 | The website shall have the ability to create calendars and have the option to set, private, public, and share to friends. Private is shared only with yourself and selected users via the share option, public includes all friends, and the share option is to specifically target friends who the user would want to share with. | Must | Users have full privacy and will decide for themselves how they want to display their information |
| FR07 | The website shall have the ability to customize the dashboard to fit the aesthetic the user is looking for. This includes accessibility for color blindness. | Want | Users should be happy logging into the website and seeing their personalized dashboard. |
| FR08 | The website should include a widget for quick access to whatever the user customized the widget to take them to, could be friends, messages, and specific calendars. | Want | This is to facilitate user fluidity. |
| FR09 | The website shall have a section for private messaging between users. This will give the ability to share your calendar, sort of how Apple iMessage allows games locally. | Must | This will make it much easier to coordinate and send each other calendars. |
| FR10 | The website shall have the user dashboard have a bell that will pop up if the user receives a notification while logged onto the website, this can be turned off in the settings. | Must | Notifications are important, but the user should have the option to remove them |
| FR11 | The website dashboard shall have the ability to add widgets, apart from calendars, for example, a weather widget, or a to-do list widget | Must | Keeping things organized and being up to date on the weather is essential in an app that revolves around calendars |
| FR12 | Users shall be able to customize their dashboard and move around their widgets, to fit the look they want. This includes Calendars, Weather, and the to-do list | Must | Being able to customize the look and feel helps reinforce our idea of full user customization, this also refers to FR07 |
| FR13 | The website shall have a “contact us” page, that will include a business email, telephone, and FAQ | Must | Not every user has a seamless experience and might need extra help, or might have ran into a bug |

# **Non-functional requirements**

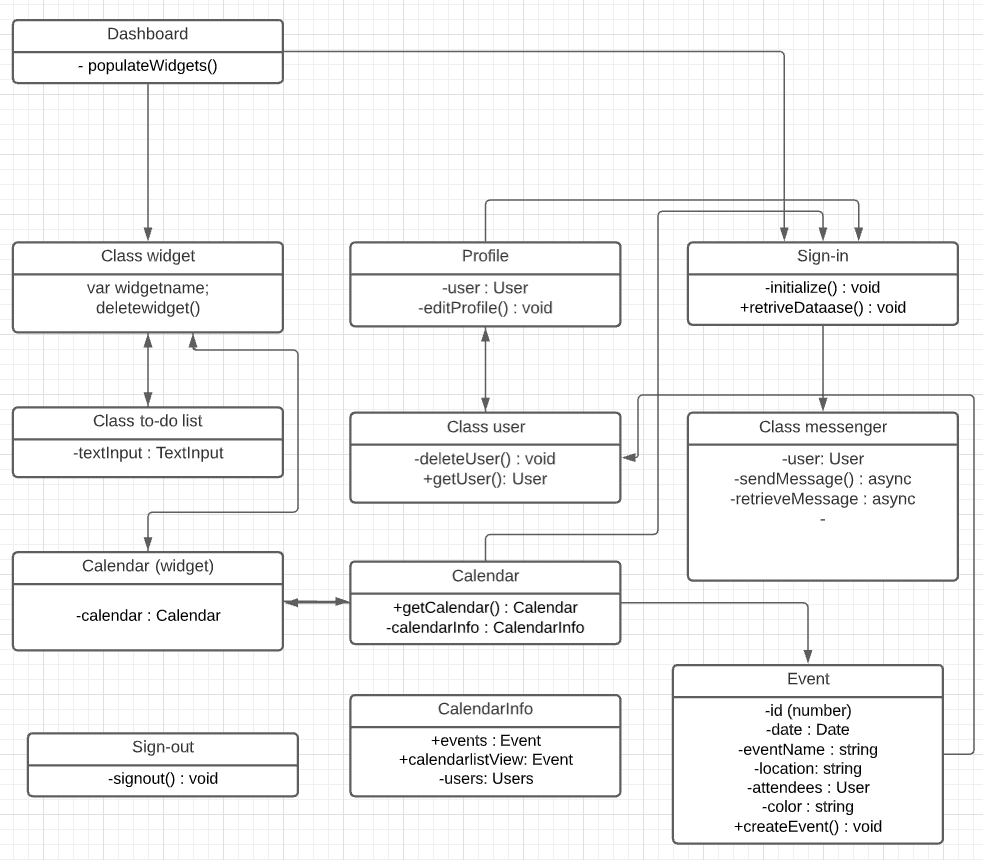
|  |  |
| --- | --- |
| **No.** | **Non-Functional Requirement Description** |
| FR01 | **Usability**   * Must have color blind mode, and dark mode for customizability and user necessity. (colors customizable) * Buttons must be large enough for users of all ages. (customizable) * Calendars will also be customizable for user comfort. (size, color, location) * Text will be customizable. |
| FR02 | **Reliability / Availability**:   * Stable website, no bugs, or crashes * Website cannot crash from too many users online at once(sometimes it's unavoidable) * In case of outage there will be back-up generators to give time to inform the users of an outage and estimated server downtime. * Servers will be owned and accessible by CalendarConnect to ensure upkeep and uptime. |
| FR03 | **Scalability**:   * The system will be upgraded according to the number of users * Server will be able to hold a greater number of users than the expected amount for lower latency, faster response times, and server overcrowding. * Physical server space will be sufficient for future upgrades. |
| FR04 | **Performance**   * Quick response time, low MS latency on clicks. * Smooth transitions(low MS and reactive movements) * When pinging a user, the ping should be near instant.(notification, calendar requests, etc.) * Functional buttons, users should not have to click more than once to access anything on CalendarConnect. (hitboxes) |
| FR05 | **Supportability**   * Will have support for android and IOS (down the line) * CalendarConnect will have an email that will allow customers to send their perspectives to the company. This will allow for the Devs to make changes accordingly. * CalendarConnect will have a short informational video explaining how to use the website/app and its features. |
| FR06 | **Security**:   * Highly maintained security measures, especially since CalendarConnect deals with personal information such as locations and schedules. * Cloud and physical user information must also be protected at all times. Security measures involving these will move quickly and secretly. |

# **High-level system architecture**

* Database and Media Storage: The database is broken up into several sections the first is the user database which hold the basic user information such as the unique id, user’s name, email, phone, date of birth, date of registration, user profile picture. The next database is the users calendar database which holds the unique id and the calendar object in a json format. The third database is to hold user posts.
* Search/Filter Architecture: Within our application the user is able to search for other users, contacts, events, and messages. We will implement this by utilizing a custom recursive substring searching algorithm.
* API: There are no current plans to develop and API for our project
* When searching the search output displays the favorited contacts, events, and messages with a priority to flagged objects. This is done by utilizing custom heap sorting algorithm that prioritizes flagged objects.
* **Languages** : CSS, Html, JavaScript, ajax, tailwindcss, nodejs, typescript
* **API**: Open Weather API, Google Calendar API, Outlook API, iCloud API, Nylas API (potentially)
* **Browsers**: Chrome, Edge, Opera, Safari, Mozilla Firefox, and Android “Internet”.
* **Frameworks**: React.js, MongoDB, express, Signal E2E, FullCalendar.js (potentially)

All external code you plan to use must be listed along with their license. (GOES HERE)

# **High-Level UML diagram**



# **Risk Identification**

**Skills risks:**

The project certainly has an issue when it comes to the skills of our members as a whole, most of the team is inexperienced with JavaScript, CSS and HTML, our back end is doing the best they can to learn and adapt to the situation to try and apply their overall programming knowledge, into creating a functional website. The front-end team has not seen much progress in the addition of CSS skills, we are currently working on having a functional website first and then using our time to perfect the aesthetic of it.

**Schedule risks:**

Most of our members have very tight schedules, with work and university and it has been progressively harder to get everyone to meet and start working on the project, that being said when we do meet, we are friendly and able to get work done, the group has their concerns about being able to deliver all the promised features by the end of the class but will do our best to deliver as much functionality as we can, we plan on seeing through this assignment if it is not completed fully by the due date.

**Technical risks:**

The back end has a good idea of how to implement everything, currently we are having an issue fully merging all the code together on to the lamp server, this should not be a big deal and will be done by the beta and final beta. Our front-end knowledge is creating code that will not break making changes is an important factor we need to take into consideration given that we want our application to look aesthetically pleasing. For this, we used a bootstrap template as our foundation for our front-end design which allows for easier design and manageability especially since it includes customizations depending on the screen size.

**Teamwork risks:**

Our teamwork in terms of how the group gets along is not an issue, we do, however, lack the initiative to meet and discuss progress on the project, we are not seeing as much effort as we would like, or communication, but will do our best to resolve these internal issues to give the user a good experience.

**Legal/content risks:**

Currently, we don’t have any issues with API, and framework, licensing. The APIs and frameworks used are covered under the MIT public license and are free to use.

The way we plan on resolving these risks, would be to make sure everyone is equally involved and does their part in trying to address these issues and finding a solution for each risk that benefits the entire team.

Addressing each issue individually, the skills risk is not a big issue, we all have been through coding courses and currently some of the members have studied a lot of web development code in another course, this helps the team learn where to find information and how to answer any concerns and doubts. We will continue to teach ourselves and trial and error code, to find the best solution to fix any problems.

The Schedule risk is a difficult one to fix, as mentioned above most members have a tight schedule with a priority of work, we would have to find a time that's best for everyone but even then, that would be a difficult task as it is a group of 5, this is exactly the reason we want to create CalendarConnect; with a website like we could solve scheduling problem similar to ours. In the meantime, we send updates to the chat and try to assign tasks to work remotely.

The team is currently having an issue merging the code from each back-end member for full functionality. Lack of communication has also been an issue, as described above, we have seen both back-end, and front-end, work getting completed twice. Once our communication issue is resolved, we should no longer have this issue.

The Teamwork risk is probably the risk most affecting the group right now, we have not had the most involved past couple of weeks and are struggling to maintain a consistent schedule, this risk is to be addressed after this milestone as we do not have much time to get everyone on task. The way we are going to resolve this risk is by having a pretty serious conversation with the group about their commitment to the success of our project and getting to the root of the problem.

Currently we don't see any legal risk in the near future.

# **Team**

**Group Name:** Alpha Technologies

**Scrum Master:** Gianluca Guagliardo

**Product Owner:** Leon Rice III

**Initial Roles:**

Charlton Benting (Dev Team/Backend)

Gianluca Guagliardo (Dev Team/Backend)

Hendrein Paul (Dev Team/Frontend)

Mayte Ramirez-Calderon (Dev Team/Frontend Lead)

Leon Rice III (Dev Team/Backend Lead)

# **Checklist**

For each item below you must answer with only one of the following: DONE, ON TRACK (meaning it will be done on time, and no issues perceived) or ISSUE (you have some problems, and then define what is the problem with 1-3 lines). Reflect these items in your Jira project space:

1. Team decided on basic means of communications: **DONE**
2. Team found a time slot to meet outside of the class: **DONE**
3. Front and back-end team leads chosen:
   * Frontend: Mayte Ramirez-Calderon
   * Backend: Leon Rice III
4. GitHub master chosen: Gianluca Guagliardo
5. Team ready and able to use the chosen back and front-end frameworks: **DONE**
6. Skills of each team member defined and known to all: **DONE**
7. Team lead ensured that all team members read the final M3 and agree/understand it before submission: **DONE**