**Academic Calendar Using Voice Assistants**

*By: Zack Tuttle, Andy Duong, Stefen Sharkey, and Evan White*

*Final Revision – May 15, 2020*

# Intro and Objective

Our goal is to create a skill and action for Amazon Alexa and Google Assistant, respectively, which will allow the user to search through their college’s academic events page and retrieve information through verbal requests. With this software, users will be able to easily find these important dates in a matter of seconds so they can focus on other important tasks, such as studying, completing assignments, or preparing for a trip to see a relative or graduate.

# 

# Table of Contents

[**I. Audience: Why and Who?**](#_d9qf5zbqhqzj) **2**

[1. Why build it?](#_kue7yw8dyex7) 2

[2. Who’s it for?](#_ie103744ljg2) 2

[a. Students](#_dk9v0ghctuzo) 2

[b. Faculty](#_r5ev5jqvib64) 2

[c. Student’s Relatives and Friends](#_7j1v237xx3m0) 3

[**II. Requirements**](#_yjugqo61upuk) **4**

[**III. Software Architecture**](#_63ahcu72vcqn) **5**

[1. Events (Table)](#_466yekjivqwu) 5

[2. Parses](#_lcpruawpzbjg) 5

[3. Database](#_yynxypkfw2pe) 5

[4. Voice Utterances](#_yjsmqyunkayd) 5

[5. Voice Assisted Responses](#_y32kegkx94d) 5

[**IV. Use Cases**](#_ba26h8ietec3) **6**

[1. Get Events on Date](#_453bdvrmjvv1) 6

[2. Get Dates of Event](#_oxkuamg6jjya) 7

[3. Get Days Until Event](#_hbujotbc09k6) 8

[4. Get Future Events](#_nw56p4d398pk) 9

[1. Get Events on Date](#_tco6pmy18nab) 10

[2. Get Dates of Event](#_7cizppbfb4y2) 11

[3. Get Days Until Event](#_px4te4pfoob) 12

[**VI. Timeline**](#_wt3qlua1p28x) **13**

# I. Audience: Why and Who?

## 1. Why build it?

1. Voice assistant technology has been growing rapidly in prevalence, and most new devices now come equipped with it in some form or another.
2. Google and Amazon provide powerful, intuitive resources for developers to use to build on their platforms.
3. Obtaining information through conversation is inherent to humanity, and thus requires very little technical knowledge to use, making this product useful to people of all levels of technical literacy.
4. Provides an accessible way for people with various different disabilities to access information.

## 2. Who’s it for?

### a. Students

Students already have enough on their mind. Why not make it a bit easier for them when it comes to planning out their semester? They will be able to easily figure out any important upcoming dates that were posted on their college’s academic calendar.

An example user is:

* “I am a student at SUNY Brockport. I am disorganized in planning my final few weeks of the semester and I have graduation. I need to use the alexa skill to help me organize my events going on later in the semester. ‘Alexa/OK Google, add to my calendar my mid-term dates.’”
* Alexa performs this task and responds to the user with a confirmation.
* “Now, I am able to view my upcoming finals I may have and plan out my study sessions more effectively.”

### b. Faculty

Faculty being able to check the correct dates for events is crucial when sharing event details with their students.

An example user is:

* “I am a professor at SUNY Brockport. I need to check if dates for a guest speaker have changed, I need to check the publicly available event details. ‘Alexa/OK Google, when is the date for John Doe’s speech?’”
* Alexa parses the name and responds with the corresponding date.
* “Now, I am able to confirm the date so I relay correct information and modify my schedule as necessary.”

### c. Student’s Relatives and Friends

Relatives and friends of students often want to attend the graduation events and need to schedule time off appropriately.

An example user is:

* “I have a nephew who is a student at SUNY Brockport and they are graduating this semester. I need to take time off to go to their graduation but I don’t know the date of graduation. ‘Alexa, what is the upcoming date for graduation at SUNY Brockport?’”
* Alexa parses the graduation keyword and responds with the date.
* “Now, I am able to take time off of my job and not miss my nephew’s graduation.”

# 

# II. Requirements

There exist several requirements in order to successfully serve various information about an academic calendar to a user. Some requirements are considered functional—features that are necessary in order to satisfy every major need for an end user. Some requirements are considered non-functional—features that are not necessary, but desirable to the end user, and often enhance a functional requirement. In Table 1, all requirements are categorized into functional vs. non-functional.

|  |  |
| --- | --- |
| **Functional** | **Non-Functional** |
| Parsing the academic calendars of The College at Brockport, | Informing the user of all *events* that occur in the upcoming week. |
| Given a *date*, informing the user of the *events* occuring on that *date*. | Adding *events* from the college calendars to a user’s Google Calendar. |
| Given an *event*, informing the user of the *dates* on which that *event* is occuring. |  |
| Given an *event*, informing the user of how far away that *event* is. |  |

###### Table 1. Functional vs. Non-Functional Requirements

# 

# III. Software Architecture

A software architecture explains, in a high-level overview, the make-up of a software system and all of its components.

## 1. Events (Table)

* Event calendars for most college websites are in the form of an HTML table.
* The key of this table is often the date, where the value of this table is the event itself.

## 2. Parses

* The event table is given a link and information then gets separated into dates and corresponding events to allow for data storage.

## 3. Database

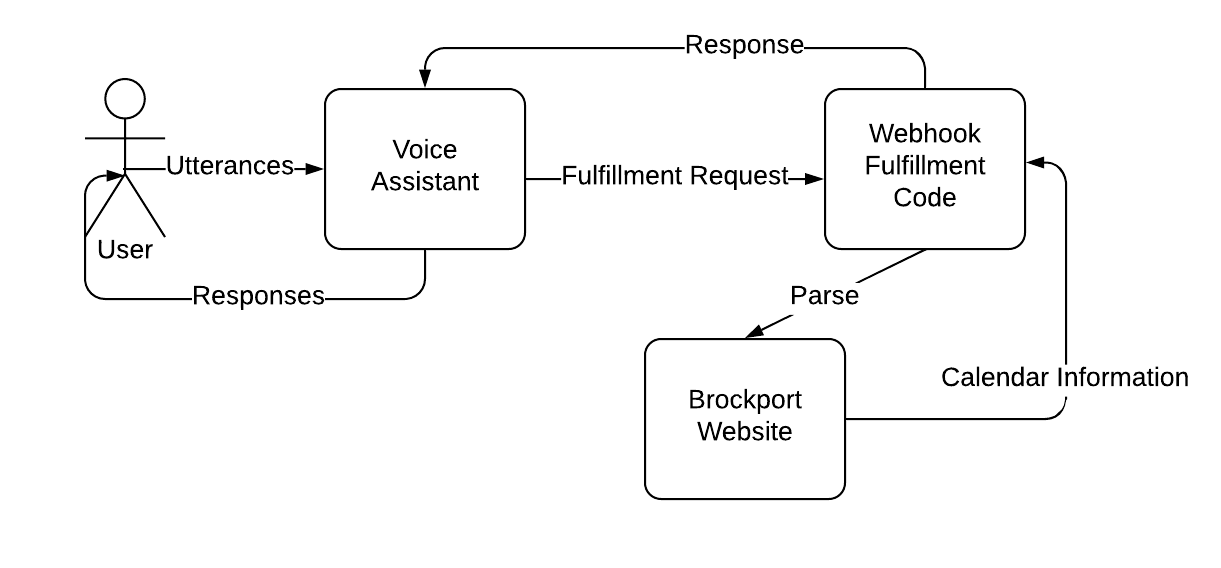
* Will be stored locally.
* Populate the database based on the date and event format that was acquired from the parse.

## 4. Voice Utterances

* Stated in the form of a question/statement, depending on context.
* Amazon or Google libraries will separate the question into text that the database will use in order to retrieve the given response, such as a date or the name of a specific event.

## 5. Voice Assisted Responses

* Amazon or Google libraries will reply with the response tailored to the user’s request.
* Will ask the user if they want to ask about any other dates or events.

****

###### Figure 1. Software Architecture Diagram

# IV. Use Cases

A use case is a moderately lower-level, more technical, explanation of how each end-user action will be performed by the software system.

|  |  |
| --- | --- |
| Use Case Name: | 1. Get Events on Date |
| Description: | |
| The User inquires about the College Events on a certain date and the system provides the requested information. | |
| Preconditions: | |
| 1. The date is within the current academic year. | |
| Workflow: | |
| 1. User approaches the Assistant to begin the interaction and requests the College Events that fall on a specific date. 2. Assistant looks up the College Events folder using the information provided in step 1. 3. Assistant retrieves a collection of all College Event records from the College Events folder using the information provided in step 1. 4. Assistant reads off each event in the collection to the User, specifying the times for the College Events if applicable. | |
| Results: | |
| 1. The User is provided College Event(s) from the specified date. | |
| Alternates: | |
| 1. The date is not within the current academic year. 2. There are no College Events on the requested date. 3. The College Event is not found in the College Event folder. | |
| Entities Involved: | |
| User, Assistant, College Events folder | |

### Example Utterances

1. “What is happening on April 8th?”
2. “What is scheduled for tomorrow?”
3. “Is there anything happening on May 5th?”
4. “What is today?”
5. “What is on the calendar on March 20th?”
6. “Is anything scheduled for February 24th?”

|  |  |
| --- | --- |
| Use Case Name: | 2. Get Dates of Event |
| Description: | |
| The User inquires about the date from a certain College Event and the assistant provides the requested information. | |
| Preconditions: | |
| 1. The Event exists within the College Event calendar. | |
| Workflow: | |
| 1. User approaches the Assistant to begin the interaction and requests the date on which a specific College Event falls. 2. Assistant looks up the College Events folder using the information provided in step 1. 3. Assistant retrieves a collection of College Events records from the College Events folder in step 2 which occured on the given date provided in step 1. 4. For every College Event record in the College Event collection retrieved in step 3, the Assistant:    1. Retrieves the date of the College Event.    2. Provides the date retrieved in step 4a to the User. | |
| Results: | |
| 1. The User is provided a Date from the specified College Event. | |
| Alternates: | |
| 1. The College Event is not within the academic calendar. 2. There is no date from the requested College Event. 3. The College Event is not found in the College Event folder. | |
| Entities Involved: | |
| User, Assistant, College Events folder | |

### Example Utterances

1. “When are midterms?”
2. “When is Scholar’s Day?”
3. “When does Spring break begin?”
4. “On what date is there a reading day?”
5. “What day does the add period end?”
6. “On which day do classes resume?”

|  |  |
| --- | --- |
| Use Case Name: | 3. Get Days Until Event |
| Description: | |
| The User inquires about how many days are remaining until a College Event occurs and the system provides the requested information. | |
| Preconditions: | |
| 1. The date is within the current academic year. 2. The College Event has not yet occurred. | |
| Workflow: | |
| 1. User approaches the Assistant and requests the number of days until a certain College Event occurs. 2. Assistant looks up the College Events folder using the information provided in step 1. 3. Assistant retrieves a collection of all College Events records from the College Events folder with a start date after the current date and matches the given information in step 1 and presents them to the User to choose the correct one. 4. User selects the desired College Events record from the given College Events collection provided by the Assistant in step 3. 5. Assistant retrieves the desired College Events record selected in step 4. 6. Assistant determines the number of days until the College Event retrieved in step 5 occurs. 7. Assistant provides the information from step 6 to the User. | |
| Results: | |
| 1. The User is provided the number of days until a certain College Event occurs. | |
| Alternates: | |
| 1. The date is not within the current academic year. 2. There are no requested College Events that have not already occurred. 3. The College Event is not found in the College Event folder. | |
| Entities Involved: | |
| User, Assistant, College Events folder | |

### Example Utterances

1. “How many days until Scholar’s day?”
2. “How far away is Spring break?”
3. “How long until undergraduate registration begins?”
4. “How soon is the Honors and Awards ceremony?”
5. “How much time until final grades are due?”

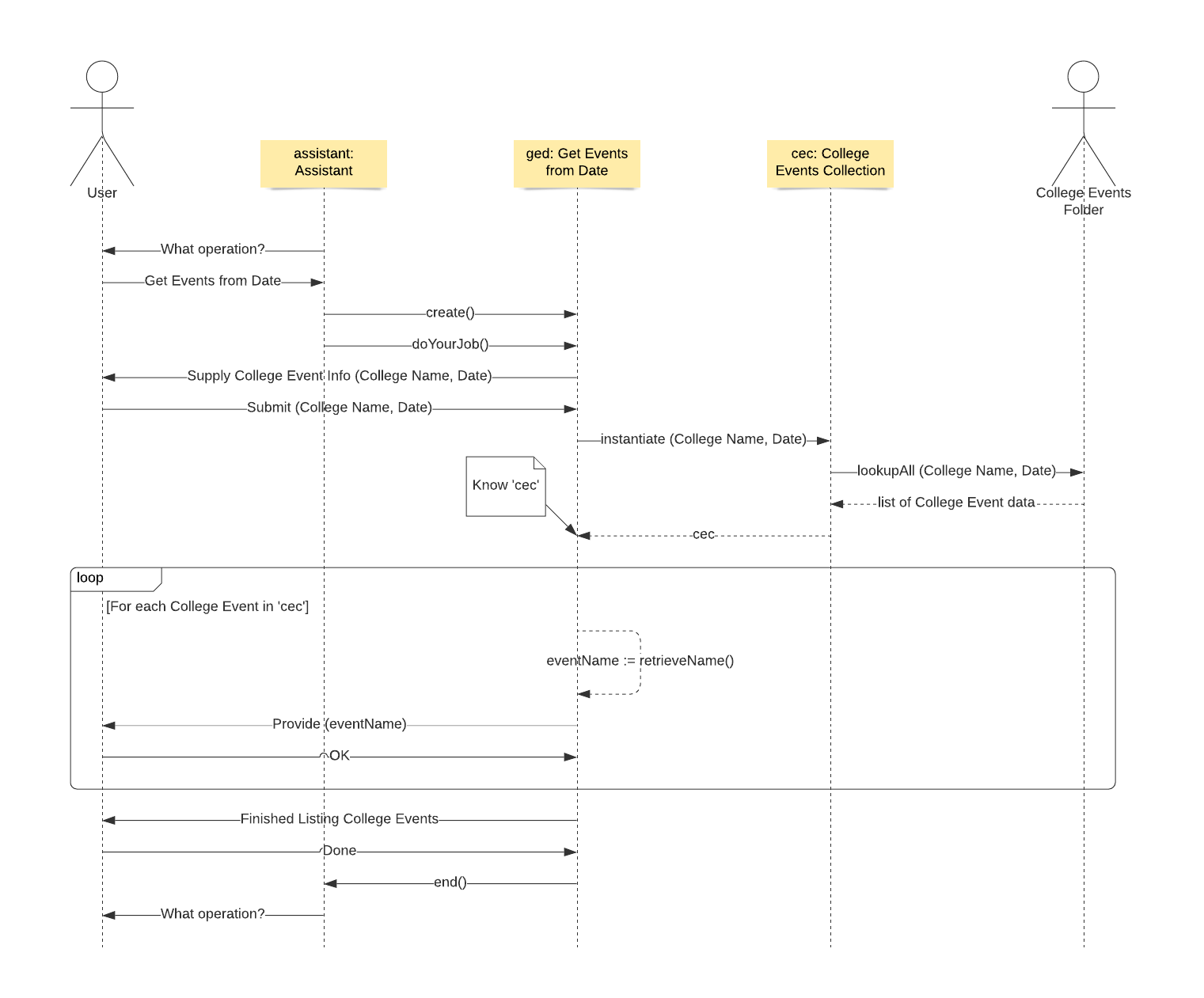
|  |  |
| --- | --- |
| Use Case Name: | 4. Get Future Events |
| Description: | |
| The User inquires about upcoming events in the next N days | |
| Preconditions: | |
| 1. N > 0 2. N <= 50 | |
| Workflow: | |
| 1. User approaches the Assistant and requests to know upcoming events. 2. Assistant asks how many days into the future it should look. 3. User provides the number of days. 4. Assistant retrieves a collection of all College Events that are in the next provided amount of days from the College Events Folder. 5. Assistant puts these retrieved College Event Records in chronological order. 6. Assistant reads off the retrieved collection to the user. | |
| Results: | |
| 1. The User is provided the events in the upcoming N days. | |
| Alternates: | |
| 1. There are no events in the next N days 2. N < 1 3. N > 50 | |
| Entities Involved: | |
| User, Assistant, College Events folder | |

### 

V. Sequence Diagrams

A sequence diagram goes into much more technical detail of how each specific use case will be performed by the system, exposing interactions between the various software architecture components.

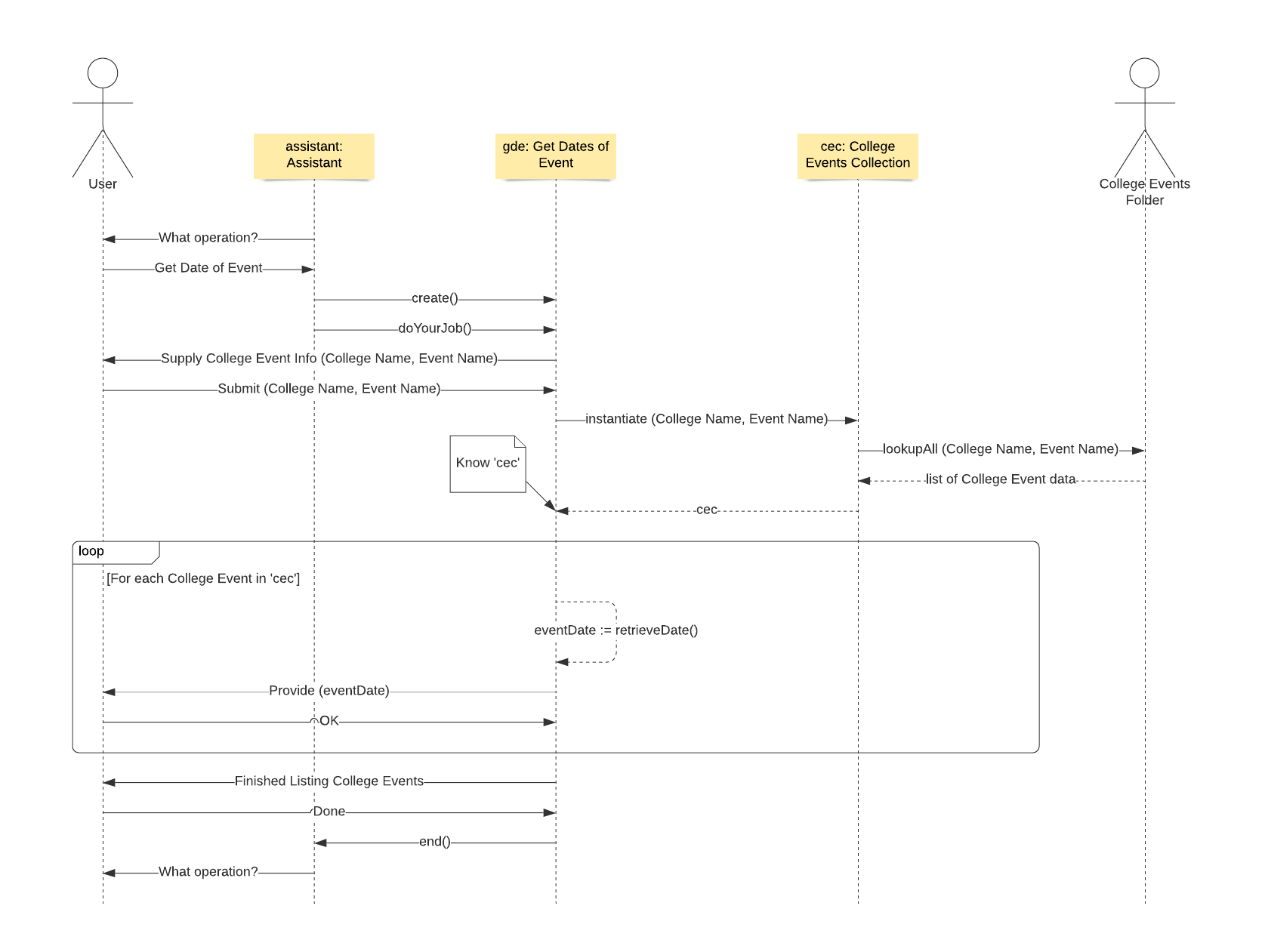
## 1. Get Events on Date



###### Figure 2. Get Events on Date Sequence Diagram

## 

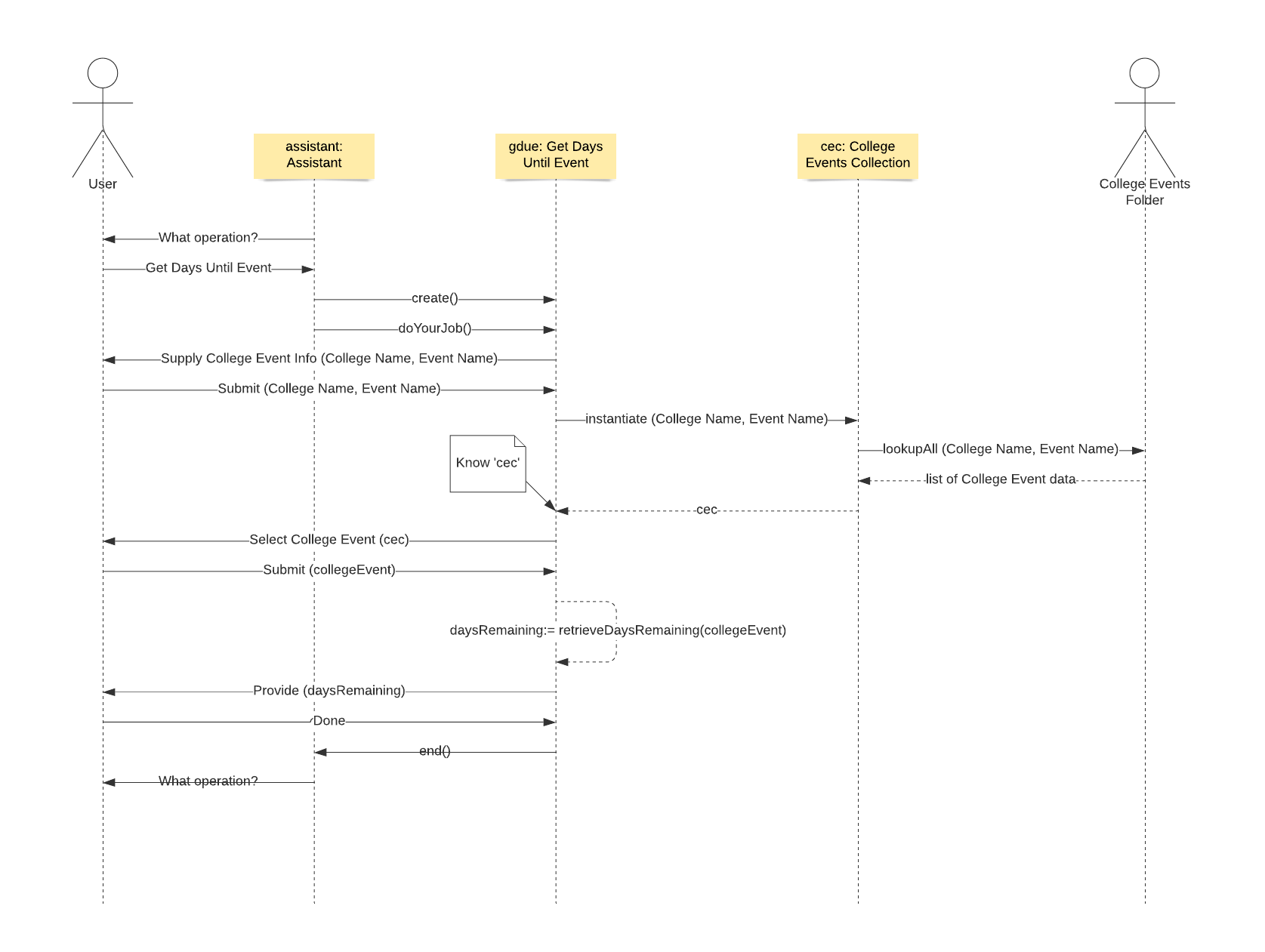
## 2. Get Dates of Event



###### Figure 3. Get Dates of Event Sequence Diagram

## 

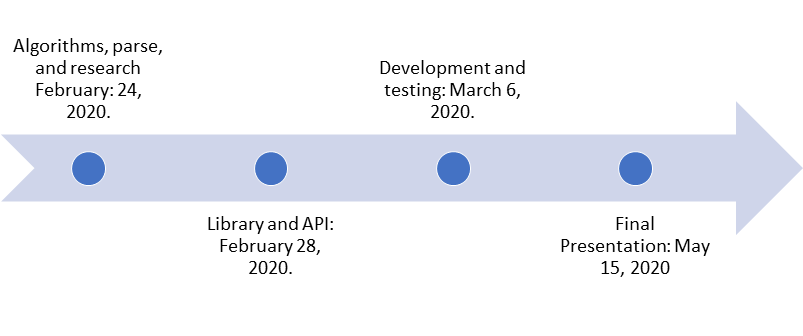
## 3. Get Days Until Event



###### Figure 4. Get Days Until Event Sequence Diagram

# 

# VI. Timeline



|  |  |  |  |
| --- | --- | --- | --- |
| **Event** | **Description** | **Due By** | **Checked** |
| Algorithms, Parse, and Research | Research into the various algorithms, including parsing algorithms, necessary to obtain information about events within an academic calendar. | February 24, 2020 | February 24, 2020 |
| Library and API | Research into the various libraries and APIs necessary to perform any and all algorithms and parsing. | February 28, 2020 | February 28, 2020 |
| Development and Testing | Completion of implementation of all research of all algorithms, parsing, and APIs necessary to fulfill all functional and any non-functional requirements. | March 6, 2020 | March 6, 2020 |
| Final Presentation | Presentation of final product requirements document, describing the various algorithms and APIs selected for the implementation. | May 15, 2020 | May 15, 2020 |