



Bilkent University

Department of Computer Engineering

---

# Senior Design Project

*SEPS*

## Analysis Report

Taner Durmaz, Samir Ibrahimzade, Mehmet Erkin Sahsuvaroglu, Alperen Koca, Burak Yeni

**Supervisor:** Shervin Rahimzadeh Arashloo

**Jury Members:** Dr. Can Alkan and Dr. Cigdem Gunduz-Demir

Project Specifications

November 21, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

<b>1. Introduction</b>	<b>3</b>
<b>2. Proposed System</b>	<b>3</b>
2.1 Overview	3
2.2 Functional Requirements	3
2.3 Nonfunctional Requirements	4
2.4 Pseudo Requirements	4
2.5 System Models	4
2.5.1 Scenarios	4
2.5.2 Use Case Model	7
2.5.3 Object and Class Model	8
2.5.4 Dynamic Models	9
2.5.4.1 Activity Diagram	9
2.5.4.2 Sequence Diagrams	10
2.5.5 User Interface - Navigational Paths and Screen Mock-ups E	12
2.5.5.1 Signup Screen	13
2.5.5.2 Login Screen	14
2.5.5.3 Event Creation Screen	15
2.5.5.4 Event Selection Screen	16
2.5.5.5 Event Photos Screen	17
2.5.5.6 Upload Photo Screen	18
2.5.5.7 Profile and Customization Screen	19
<b>3. Other Analysis Elements</b>	<b>20</b>
3.1. Consideration of Various Factors in Engineering Design	20
3.2. Risks and Alternatives	21
3.3. Project Plan	21
3.3.1 Project Goals	21
3.3.2 Milestones	21
3.3.3 Work Packages	22
3.3.3.1 Work Package 1	22
3.3.3.2 Work Package 2	22
3.3.3.3 Work Package 3	22
3.3.3.4 Work Package 4	22
3.3.3.5 Work Package 5	22
3.4. Ensuring Proper Teamwork	23
3.5. Ethics and Professional Responsibilities	23
3.6. Planning for New Knowledge and Learning Strategies	23
<b>4. Glossary</b>	<b>24</b>
<b>5. References</b>	<b>25</b>

# 1. Introduction

In the contemporary world, there exist various events maintained by different communities. As a part of social behaviour, most people take photos in group events and share these photos with each other.

Frequently, it is witnessed that the participants of an event search for their photos taken during the event, from every other participant. Furthermore, there might exist some photographs not taken during an event but had been associated with the event. There is no comfortable way to share and reach photos for certain events, easily identify the photos where you appear and share the memories only with the people who participated.

Our goal is to develop an intelligent system, SEPS (Social event photo sharing), which would enable users to access and share photographs taken during a group event. The users of the application, the attendants, will use the application for an authenticated entry to the event zone. Moreover, users would share and access the photographs taken during the event via the mobile application.

By integrating authenticity and content-sharing environments, SEPS will offer a vast and qualified platform for the users. For the events which are not registered in the app by organizers and not created a photo sharing environment, the SEPS will have reverse image search functionality; Therefore, the application itself will create an event(from the photos shared under unknown or uncreated event tag) and combine the photos according to certain events using machine learning algorithms.

## 2. Proposed System

### 2.1 Overview

The section displays the requirements of the application project. Scenarios, use cases,,dynamic models and UI mockups allow better comprehension and abstraction of the project which fosters both the communication of features to the clients and also the design process. A preliminary object model will be provided in the section, which will be elaborated during the design process, especially when the subsystems will be decomposed and the team expertizes more in the domain.

### 2.2 Functional Requirements

- User can create individualized accounts with their email address.
- User can login with their email or username, and password.
- User can logout.
- User can create events and send invitations to other accounts.
- User can send QR code to event participants of the created event of the user.
- User can display QR code from their screen.
- User can upload photographs to a database.
- User can tag photographs with words and events.

- User can access photographs taken during the event they participated in.
- User can download accessible photographs.
- User can access the information about the event they are participating in.

## 2.3 Nonfunctional Requirements

**Privacy:** The non attendant users will not be able to upload and access to event photos.

**Security:** Users are required to be authorized by QR protocol in order to be tagged to an event tag.

**Reliability:** The untagged photos, which are also associated to a particular event by a machine learning model, will be offered to attendant users of the particular event.

## 2.4 Pseudo Requirements

The application will be available for Android and IOS. For IOS, Swift will be used. For Android, Java based implantation will be used.

The application requires a database, which will be provided by other 3rd party providers. Image optimized structures will foster the efficiency of application overall.

## 2.5 System Models

### 2.5.1 Scenarios

#### Use Case 1:

*Use case: signup*

**Participating Actors:** Event organizers and attendants

#### Flow of events:

1. Actor chooses the "Signup" option from the opening screen.
2. Actor enters the username
3. Actor enters the email
4. Actor enters the password

#### Entry Conditions:

- \* Username must be valid in terms of language rules
- \* Username must not be in database before
- \* Username must not be null
- \* Email must not be in database before
- \* password must contain at least 8 characters

#### Exit Conditions:

- \*User signed up successfully if the user accepts the confirmation in email.

#### Use Case 2:

*Use case: login*

**Participating Actors:** Event organizers and attendants

**Flow of events:**

- 1.Actor chooses the "Login" option the opening screen.
- 2.Actor enters the username
- 3.Actor enters the password

**Entry Conditions:**

- \*username must be in the database
- \*password must be the relative with the username

**Exit Conditions:**

- \*User successfully entered the system.

**Use case 3:**

*Use case: Creating event*

**Participating Actor:** Event organizers

**Flow of events:**

- 1.Organizer can specify name of the event.
- 2.Organizer can specify the date of the event.
- 3.Organizer can specify the place of the event.
- 4.Organizer can update attendants of the event.

**Entry Conditions:**

- \*User needs to login system successfully

**Exit Conditions:**

- \*User press the "submit event" button and turn back to main menu screen

**Use case 4:**

*Use case: Entering Event*

**Participating Actor:** Attendants and SEPS

**Flow of events:**

- 1.Attendant selects the one of the desired events he or she applied for.
- 2.Attendant filter the photos in the event which he or she exist in them.
- 3.SEPS gives the information whether the attendant wears a face mask or not.
- 4.Attendant can update new photos to the selected photo sharing event.
- 5.SEPS give the notification if attendant did not upload any photos

**Entry Conditions:**

- \*User scan the qr code of the event. If the user is valid then he or she enters the event.

**Exit Conditions:**

- \*User selects the "close event" button in order to return to the main screen.

**Use case 5:**

*Use case: Upload Photo*

**Participating Actor:** Attendant and Organizer

**Flow of events:**

- 1.User uploads the photo
- 2.Organizer verifies the photo if it is taken in the event.

**Entry Conditions:**

\*If the user is in the selected event.

**Exit Conditions:**

\*User clicks the "submit photo" button.

**Use case 6:**

*Use case: Profile Customization*

**Participating Actor:** Attendant and Organizer

**Flow of events:**

1. User can specify the job title
2. User can upload profile picture
3. User can specify the age
4. User can specify the gender

**Entry Conditions:**

\*User clicks the "customization" button

**Exit Conditions:**

\*User closes the customization page

**Use Case 7:**

*Use case: Monitoring Public Events*

**Participating Actors:** Attendant and Organizer

**Flow of events:**

1. User enters the public events section
2. User chooses one of the public events.

**Entry Conditions:**

\*User logs in to the SEPS app.

**Exit Conditions:**

\*User closes the public events section.

**Use Case 8:**

*Use Case: Individual Photo Upload*

**Participating Actors:** Attendants

**Flow of Events:**

1. Users upload photos whatever they want
2. User selects the type of the upload which is public or private

**Entry Conditions:**

\*User login to the system.

**Exit Conditions:**

\*User submits the photo successfully.

**Use Case 9:**

*Use Case: Classification the individual photo*

**Participating Actors:** SEPS

**Flow of Events:**

- 1.From the individually uploaded photos which are in public format, SEPS test them and classify their tag.
- 2.SEPS creates an event with related photos.
- 3.SEPS updates the attendants
- 4.SEPS sends the notification about new photo sharing events to the attendants.
- 5.SEPS removes the selected photos from the test data input.

**Entry Conditions:**

\*If there are an adequate number of individually uploaded photos in the photo database.

**Exit Conditions:**

\*If new event created successfully

## 2.5.2 Use Case Model

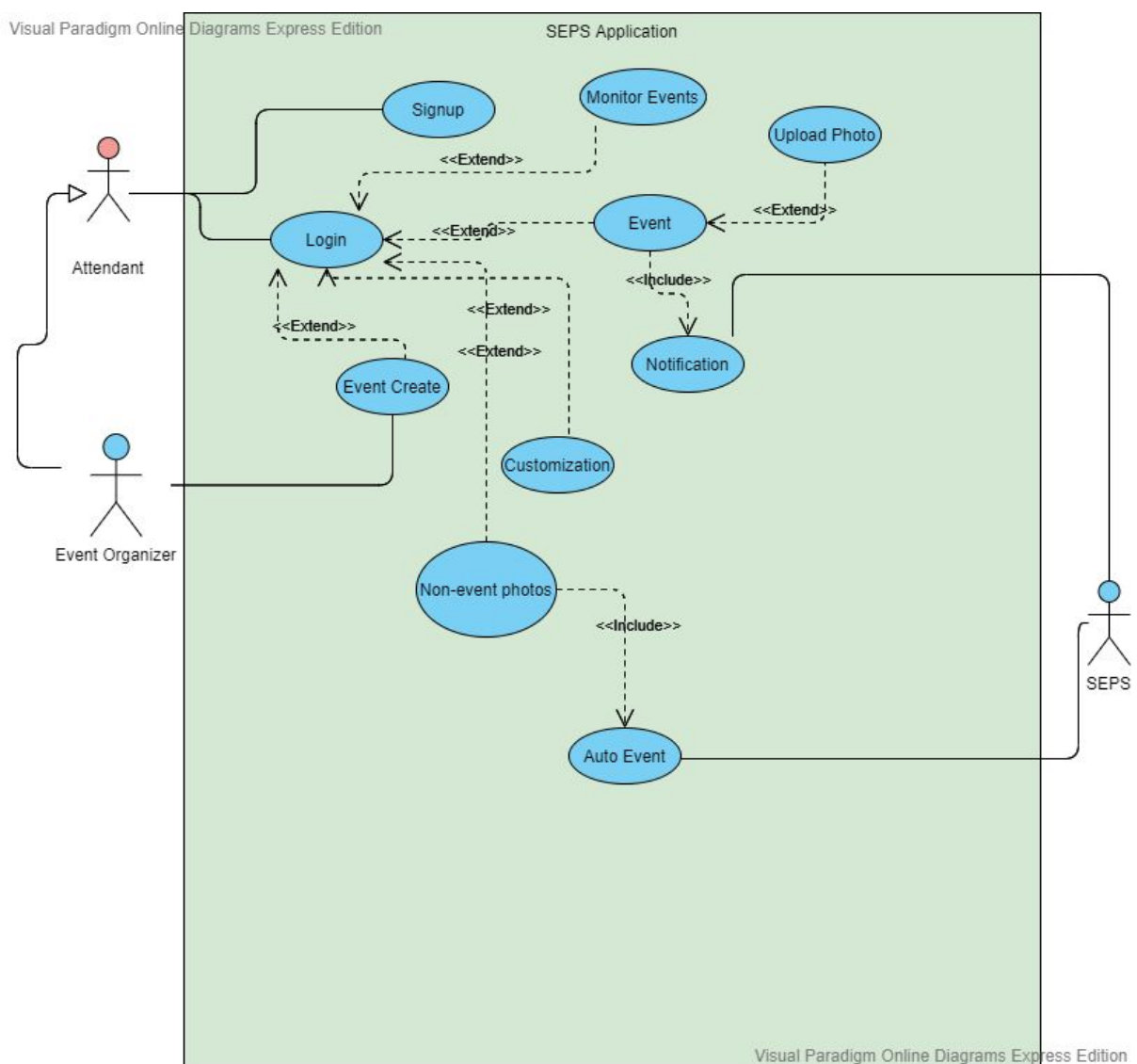


Figure 1. Use Case Diagram

## 2.5.3 Object and Class Model

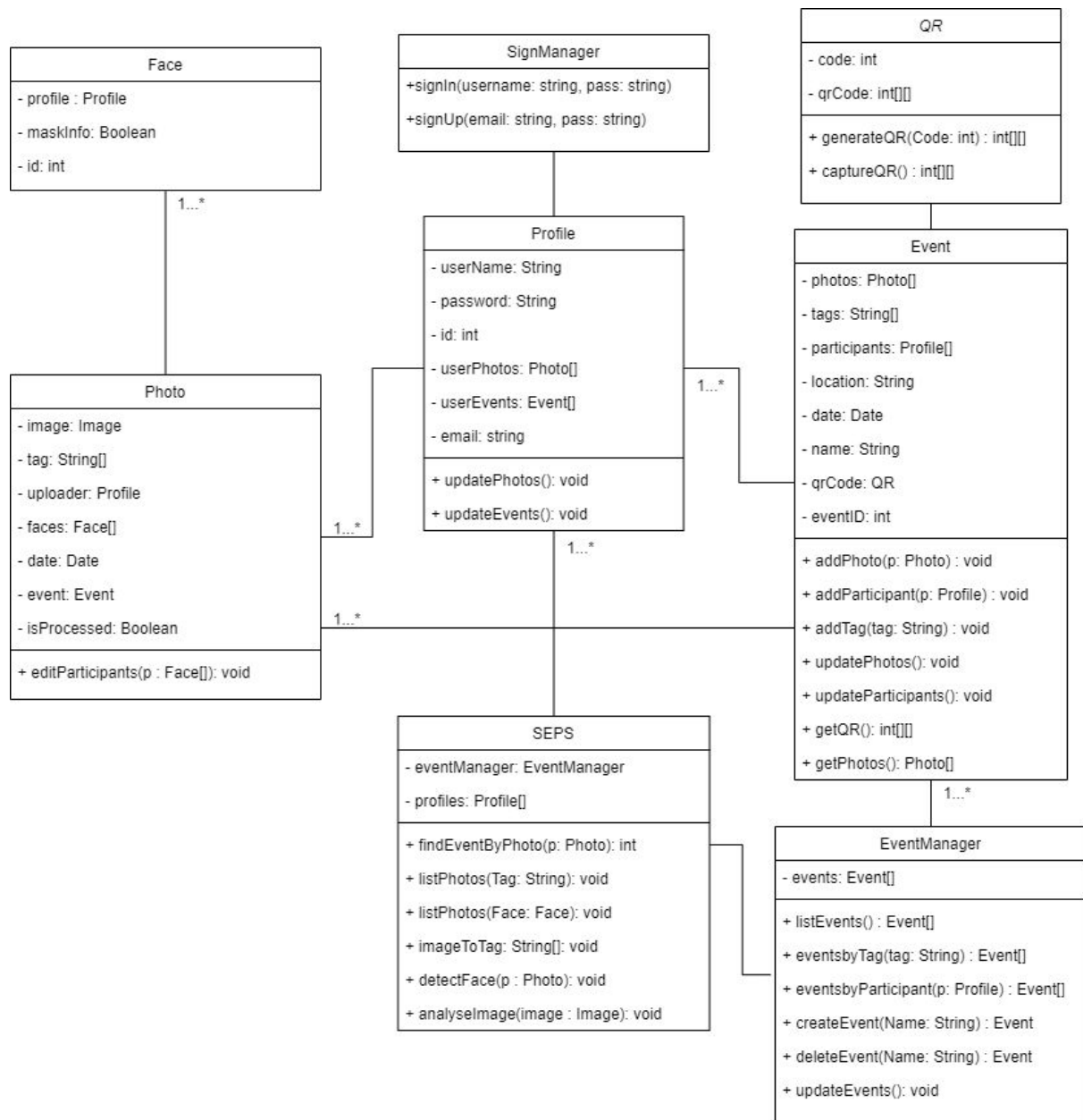


Figure 2. Object and Class Diagram

Full resolution image at the following link: <https://ibb.co/Gv1BzBq>

## 2.5.4 Dynamic Models

### 2.5.4.1 Activity Diagram

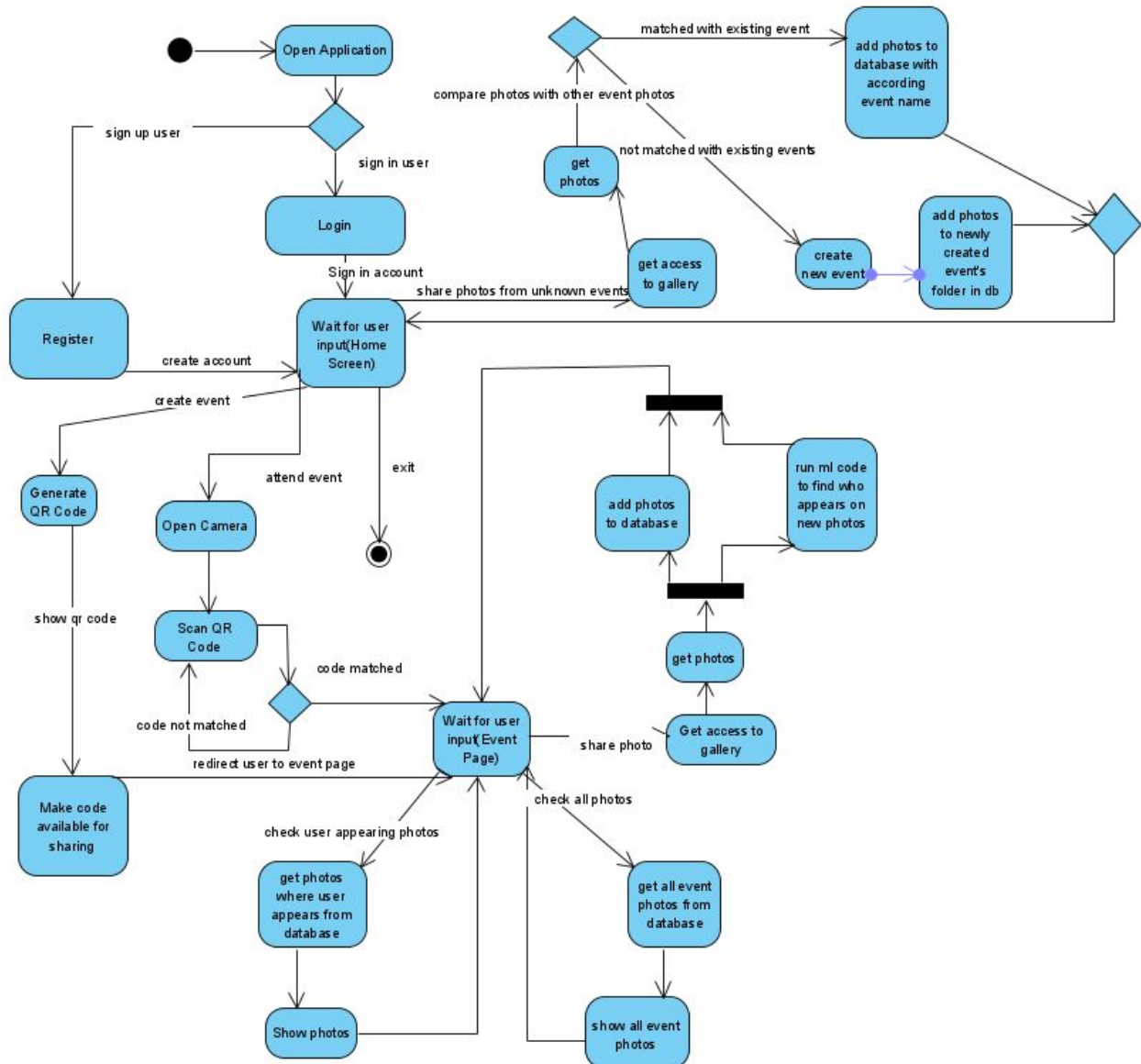


Figure 3. Activity Diagram

After the user opens the application, he/she either signs in – if already registered – or signs up and is redirected to the home screen by app.

The application waits for the user to select one of the options on the home screen: create event, attend event, share photos from unknown events. If a user chooses the create event option, then QR code is generated, made available for event organizer to share with attendees and organizer is directed to the Event Page. In case user clicks attend an event, a camera is opened for the user to scan the QR code created by the event owner. Consequently, the app tries to match the scanned code with available events and redirects user to the Event

Page – if QR code matched – or back to scanning – if code not matched. Another option for user in Home Screen is to share photos from unknown events. After getting the photos from the user, in this case, the app compares them with already available event photos and adds them to the database if matching is successful, on the other hand it creates a new event with the newly added photos.

After the user reaches the Event Page, the system waits for input from the user. The user can share photos, check all photos shared by other participants or check specific photos where he/she appears. In the first case, after getting photos the application synchronously adds photos to the database and analyzes to detect the users who appear in them. In the second and third cases respectively, the system gets all photos taken in the selected event and the photos where the user is visible.

#### 2.5.4.2 Sequence Diagrams

The below diagrams show the behaviour of several use cases distributed among objects.

##### First Diagram: User creates new event and gets qr code to share with attendees

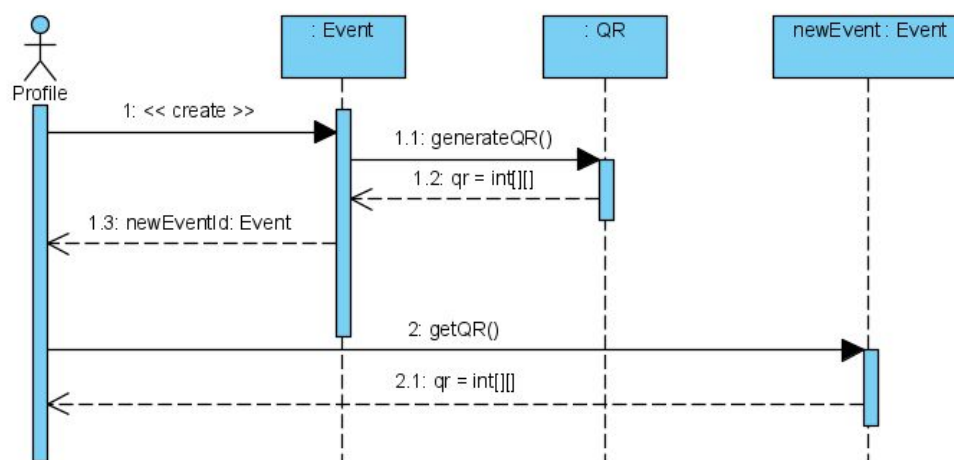


Figure 4.

## Second Diagram: User joins event and gets photos

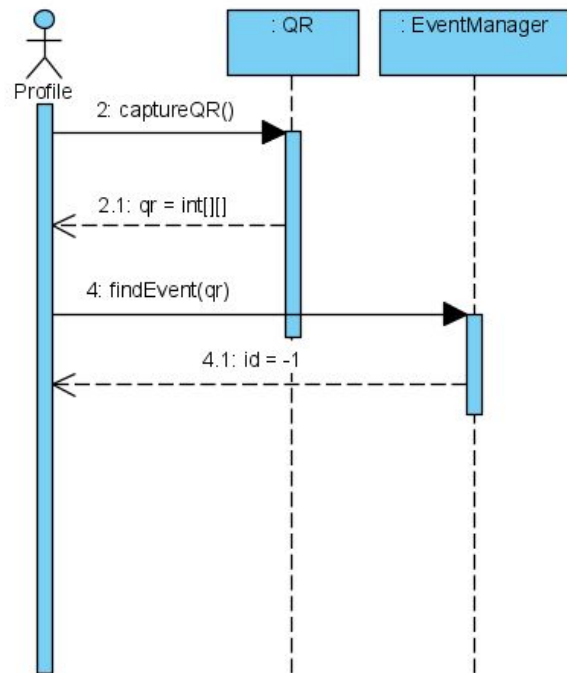


Figure 5.

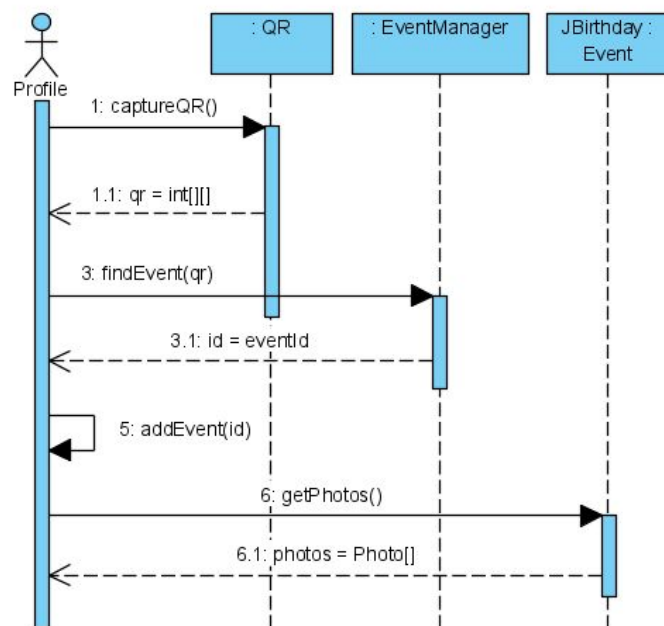


Figure 6.

Figure 4 describes the scenario where the event manager cannot find the event with scanned qr and returns -1 as id. Consequently, Figure 5 depicts the use case in which a user attends an event successfully with QR code and gets photos of that event.

### Third Diagram: User adds photos from unknown events

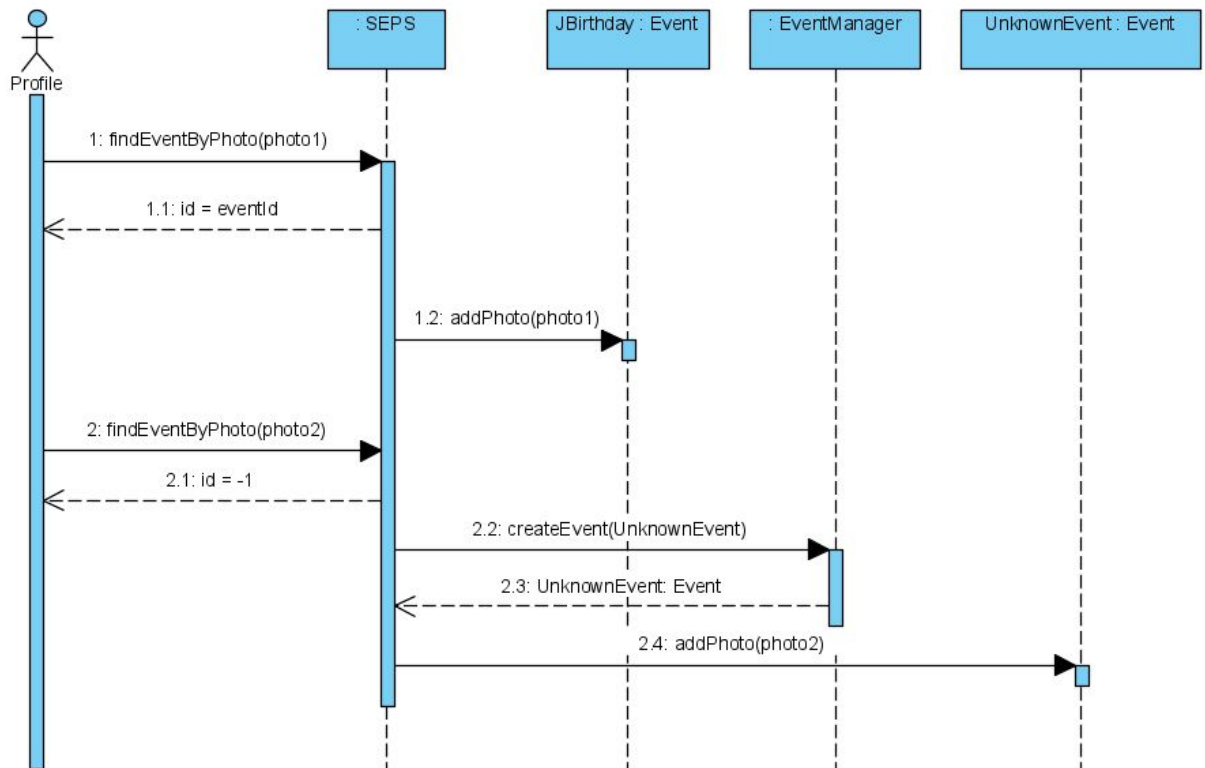


Figure 7.

The above diagram describes the scenario in which the user uploads photo1 and photo2 consequently. After photo1 is uploaded, the system tries to match the photo with already available events and finds that the photo belongs to JBirthday event; therefore the photo is added to the corresponding event. However, the system cannot find the event that matches with the photo2 and as a result, a new event (UnknownEvent) is created and photo2 is added to the event.

### 2.5.5 User Interface - Navigational Paths and Screen Mock-ups E

This part of the report includes the current state of the interfaces of the SEPS. Mock-ups are used to show how SEPS will look and feel when completed. It is important that new interfaces might be added. And mock-ups do not represent the complete and final interfaces of SEPS.

### 2.5.5.1 Signup Screen

9:05

## Sign Up

**Username**  
Your name

**Email**  
Your email address

**Password**  
Your password

☒ I agree to the **Terms of Services** and **Privacy Policy**.

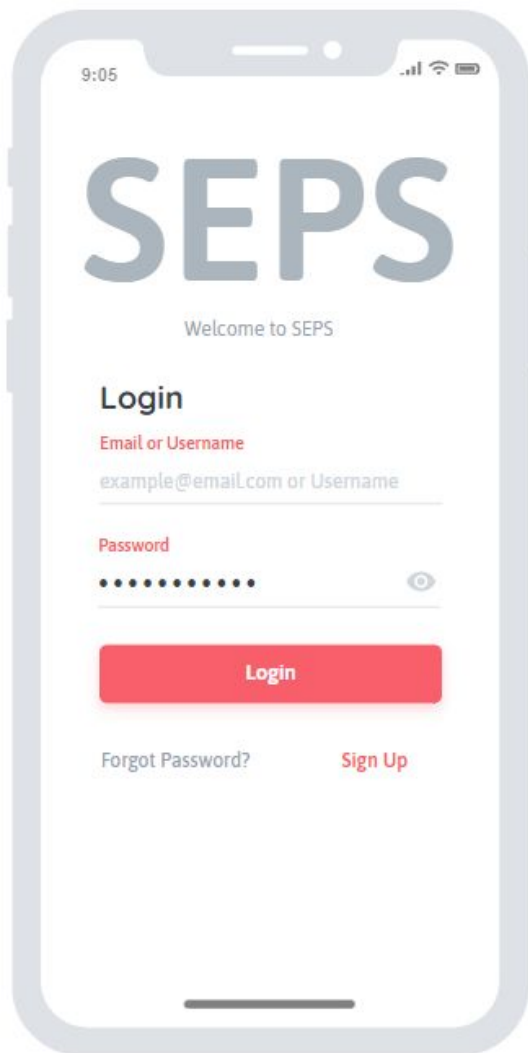
**Continue**

Have an Account? **Sign In**

*Figure 8. Sign Up Screen*

Figure 8 shows the sign up screen. Users can sign up for SEPS or go to sign in page if he/she does not have an account. If the user successfully signed up, the user is navigated to the event selection screen.

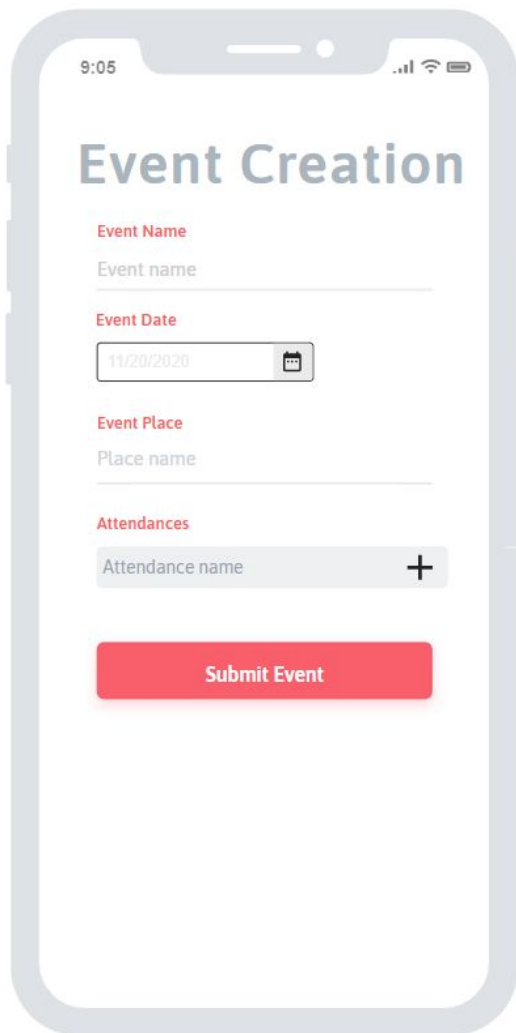
### 2.5.5.2 Login Screen



*Figure 9. Login Screen*

Figure 9 shows the login screen. Users can login into SEPS or go to sign up page if he/she forgot their password. If a user successfully login, he/she is navigated to the event selection screen.

### 2.5.5.3 Event Creation Screen

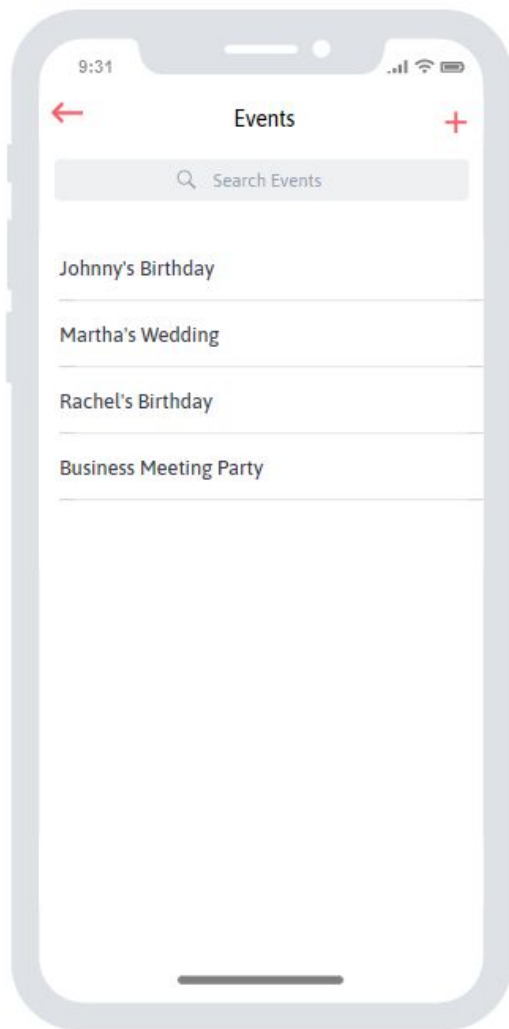


A mobile app mockup of an event creation screen. The screen is titled "Event Creation" in a large, bold, dark blue font. Below the title, there are four sections, each with a red header label: "Event Name", "Event Date", "Event Place", and "Attendances". The "Event Name" section has a text input field with the placeholder "Event name". The "Event Date" section has a date input field showing "11/20/2020" and a calendar icon. The "Event Place" section has a text input field with the placeholder "Place name". The "Attendances" section has a text input field with the placeholder "Attendance name" and a plus sign icon to its right. At the bottom of the screen is a red button with the text "Submit Event". The entire screen is framed by a light gray border representing a smartphone.

*Figure 10. Event creation screen*

Figure 10 shows the event creation screen. Organizer can create an event with name, date and place, and add attendance into the event and submit it.

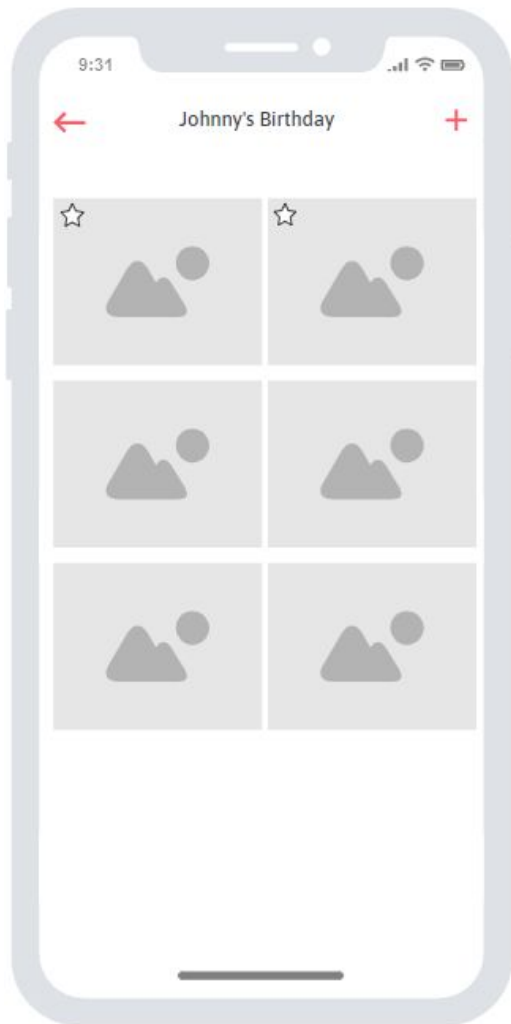
#### 2.5.5.4 Event Selection Screen



*Figure 11. Event selection screen*

Figure 11 shows the selecting event screen. This is the main screen for now. Users can search or select events to view photos of him/her. Users can also add new events by touching the “+” button and scan QR code. Users can also go to their profile page by touching the back symbol.

### 2.5.5.5 Event Photos Screen



*Figure 12. Photos in the event screen.*

Figure 12 shows photos captured in the event. Users can select the photos which he/she exists in, and also add new photos into the event by touching plus symbol. Users can return the event's list by touching the back symbol.

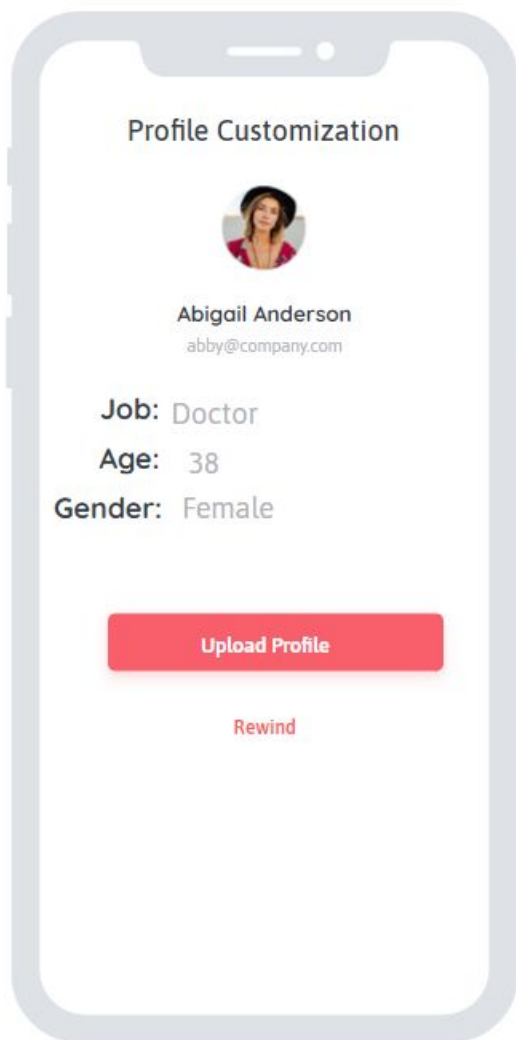
#### 2.5.5.6 Upload Photo Screen



*Figure 13. Photo upload screen*

Figure 13 shows photos captured. Users delete it and take a new one or upload it into the event. If the user selects the “upload photo” option, the pop-up screen appears upon upload photo button including options of private and public and the user selects one of them. If public selected, photo can be viewed by all participants in the event; if private is selected, then photo can be viewed by only the user who takes the photo.

### 2.5.5.7 Profile and Customization Screen



*Figure 14. Profile and Customization Screen*

Figure 14 shows a profile and customization screen. Users can view and customize their profiles. If they customize, they can save new one by the “upload profile” button. If a user does not want to upload a new profile (reverse the changes), he/she can touch the rewind button.

## 3. Other Analysis Elements

### 3.1. Consideration of Various Factors in Engineering Design

#### **Public Health**

The application's face recognition model should take COVID-19 mask regulations into consideration.

#### **Public safety**

The application should guarantee security of the event zone by the usage of QR code.

#### **Global Factors**

The application's implementation will take global trends on software solutions into account. The most trending photograph sharing applications will be analyzed in order to ensure the quality desired by users.

#### **Cultural Factors**

The application will provide users with the opportunity to easily find photographs that they do not even know but they are in. Moreover, the users will not search for their photos from other attendants, they will be able to access them directly via the application.

#### **Social Factors**

The application should use ethical constraints for determining uploadable photographs. Any abusive usage should be reported and deleted from the database.

#### **Environmental Factors**

The application reduces the amount of consumed paper due to press of invitation letters.

#### **Economic Factors**

The application does not have any economic factors dependency.

<b>Factors</b>	<b>Risk Points</b>
<b>Public Health</b>	4
<b>Public Safety</b>	5
<b>Global Factors</b>	7
<b>Cultural Factors</b>	10
<b>Social Factors</b>	5
<b>Environmental Factors</b>	0
<b>Economic Factors</b>	0

## 3.2. Risks and Alternatives

<b>Risk</b>	<b>Plan B</b>
We witnessed that the users of the application violate COVID-19 mask regulations in order to have their faces recognized.	The face recognition algorithm will be compatible with the mask regulations.
We witnessed that QR protocol is violated by sharage of the QR image.	The users violating rules will be detected.

## 3.3. Project Plan

### 3.3.1 Project Goals

- **Functionality:** The project would be evaluated as successful if the application satisfies the requirements described in functional requirements.
- **Usability:** The application will be simple so that users' experience will be enhanced.

### 3.3.2 Milestones

<b>Milestones</b>	<b>Deadline</b>
Project Specification Report	Oct 12, 2020
Analysis Report	Nov 21, 2020
Front-End Implementation	<i>TBD</i>
High-Level Design Report	Dec 21, 2020
Back-End Implementation	<i>TBD</i>
Low-Level Design Report	Feb 8, 2021
Database & Server Installation	<i>TBD</i>
Testing	<i>TBD</i>
Final Reporting	April 30, 2021
Presentations & Demo	May 2021

TBD: To be declined

### 3.3.3 Work Packages

A brief distribution of workloads and packages is given below. All the team members are expected to collaborate on every work package.

#### 3.3.3.1 Work Package 1

Leader: Taner Durmaz

Major milestones and deliverables: Project Specifications Analysis Report

Start Date: September, 21, 2020

End Date: November 21, 2020

#### 3.3.3.2 Work package 2

Leader: Samir Ibrahimzade

Major milestones and deliverables: Front-End Implementation, High-Level Design Report

Start Date: November, 21, 2020

End Date: Dec 21, 2020

#### 3.3.3.3 Work package 3

Leader: Alperen Koca

Major milestones and deliverables: Back-End Implementation, Low-Level Design Report

Start Date: December, 21, 2020

End Date: Feb 8, 2021

#### 3.3.3.4 Work package 4

Leader: Mehmet Erkin Şahsuvaroğlu

Major milestones and deliverables: Database & Server Installation, Testing

Start Date: Feb 9, 2021

End Date: *TBD*

#### 3.3.3.5 Work package 5

Leader: Burak Yeni

Major milestones and deliverables: Final Reporting, Presentation & Demo

Start Date: March 15, 2021

End Date: April 30, 2021

## 3.4. Ensuring Proper Teamwork

We are using several tools for proper communication among us, the number of such communication tools might be increased as time goes on throughout the semester. We are using Whatsapp messages, e-mail messages, Zoom calling and phone calling effectively now. However, since file sharing and editing is too hard for teamwork use on these communication tools, we are using shared Google Drive folders and Google Docs for this purpose. We also divide projects into parts and distribute them among us and set due dates in the future so

everyone in the group has certain work and time to do their work. For the later parts of the project, mainly implementation parts, we will use GitHub to share and synchronise project's separate code segments.

### 3.5. Ethics and Professional Responsibilities

- In the ethical scope, our system respects the privacy of users, so our system must keep and protect confidential and private information of users. Since our program collects the photos which are the most valued information of the users, the program must store these photos privately, and must not make these photos available publicly without permission of users who are in photos.[1]
- Users are stakeholders in computing ethics, so the stake of every user must be kept and protected by our program. In some cases, users may accidentally log in the wrong event and then he/she realises that situation and log out from the event; if this kind of situation occurred, the system would delete this user's information from the event. By doing that, our system protects the private information of the users.[1]
- In ethical view, the users must be kept distant from harm caused by computer professionals. In some cases, if any threat or action happens during the event that may cause harm to the user's information, users would inform the system; according to this information, the system will inform other user's in this event. [1]
- In order to respect the privacy of users, users won't be asked to get irrelevant information. Only the necessary information will be asked to users in our program.[1]

### 3.6. Planning for New Knowledge and Learning Strategies

We are working on machine learning and how to use it for recognition faces and researching more of the principles of already used face recognition systems. Moreover, we have some difficulties with face recognition of the users who are wearing masks, we are working hard to solve this problem. Meanwhile, we are learning the principles and properties of coding on mobile phone's operating systems such as iOS and Android. As time goes, we are gaining experience on mobile devices because we know that each new information we learned and each problem we faced will contribute to us and late periods of the project will ease.

## 4. Glossary

## 5. References

[1] The Code affirms an obligation of computing professionals to use their skills for the benefit of society. (n.d.). Retrieved November 1, 2020, from <https://www.acm.org/code-of-ethics>.