Innovation Drive and Technology Expo

Software Requirements Specification

Chris Ciolek

Nicholas Spencer

Maxx Achtman

Apolonio Cazares

# Table of Contents

[**Table of Contents**](#_dqrqas4uzbrj) **1**

[**Introduction (WIP)**](#_a380meivgz77) **3**

[Goals and Objectives](#_n1f88imqgxsc) 3

[Statement of Scope](#_u6d94tsivka2) 4

[Functional Requirements](#_qftmc61dnx8f) 4

[Non-Functional Requirements](#_dcwfq2ewx6bg) 4

[Hardware Requirements](#_nlbsszu6ahxm) 4

[Software Requirements](#_zyer5f5w2l) 4

[Inverse Requirements](#_xbxx9yt09dc) 5

[Deliverables:](#_hc61j4wrzehq) 5

[Inputs](#_uo4q8gjgoh9s) 5

[Processing](#_fou09ez8y8qk) 5

[Output](#_pupz23ctd7t5) 5

[Software Context](#_w6gdv2q6unhp) 5

[Major Constraints](#_pzy30sh2k4fl) 5

[**Usage Scenario**](#_d3bjcclo9mug) **6**

[User Profiles](#_93yn9fl1b84d) 6

[User Story Descriptions](#_4znof4okd8a5) 7

[Register as Evaluator](#_h5laa9f8bddb) 7

[Register as Supplier/Presenter](#_y7kkxpud1nj4) 8

[Modify Registration](#_ayd04bfyxnxj) 8

[Sign Waiver](#_n7h1nh33ls2r) 8

[Fill Registration Form](#_dr30nz5in8fc) 8

[Confirm Registration](#_8oekjbh9wjac) 8

[Log In to Admin Panel](#_8oekjbh9wjac) 9

[Print Attendee Badges](#_8oekjbh9wjac) 9

[Send Email](#_8oekjbh9wjac) 9

[Open/Close Event Registration](#_8oekjbh9wjac) 9

[Add/Remove Event Dates](#_8oekjbh9wjac) 10

[Check in Attendee](#_8oekjbh9wjac) 10

[Add/Remove Technology Category](#_8oekjbh9wjac) 10

[Add/Remove Technology](#_8oekjbh9wjac) 10

[Add/Remove Admin Account](#_8oekjbh9wjac) 11

[Edit Admin Accounts](#_8oekjbh9wjac) 11

[Generate Badge Sheet](#_8oekjbh9wjac) 11

[Generate QR Code](#_8oekjbh9wjac) 11

[Special Usage Considerations](#_tc3dzs48nq1y) 11

[**Data Model and Description**](#_7xrt82cjm9bs) **12**

[Data Objects](#_a6lxrf4j29ia) 12

[Relationships](#_313dtfo1s5au) 14

[Data Model](#_9neftflvu75) 15

[**Functional Model and Description**](#_89ks3lbkgieo) **15**

[Description for Function n (PSPEC)](#_ftq9tvhmhb6x) 15

[Register as Evaluator](#_ym8sddsw88i5) 15

[Register as Supplier/Presenter](#_9msdwa6r3jho) 16

[Modify Registration](#_m22a5znvt7k0) 17

[Sign Waiver](#_1qebr5jnc0pr) 17

[Fill Registration Form](#_gh4km6th73rf) 17

[Confirm Registration](#_9bify5hszdge) 18

[Log In to Admin Panel](#_iwsq3vljezv) 18

[Print Attendee Badges](#_m3u6tdmlzj40) 19

[Send Email](#_2clpcnn5qkva) 19

[Open/Close Event Registration](#_4pyzkv1hk0ma) 19

[Add/Remove Event Dates](#_9gpxhg5nktez) 20

[Check in Attendee](#_sxg4ghw5xrlq) 20

[Add/Remove Technology Category](#_nm7apo3qcymm) 21

[Add/Remove Technology](#_dmiw5ago6faw) 21

[Add/Remove Admin Account](#_t0hdcmt6p0um) 21

[Edit Admin Accounts](#_ymy5qxzcu139) 22

[Generate Badge Sheet](#_kimmns33563) 23

[Generate QR Code](#_9abrp1hibur3) 23

[Software Interface Description](#_y5w93mkg5400) 23

[Activity Diagrams for Core Functionality](#_xafrrv7emel6) 24

[Attendee Registration](#_i0ke2h1tq0nf) 24

[Modify Registration](#_ko1zksthahxc) 25

[Checking in Attendee](#_8o4hvbh2l9fz) 25

[Add Event Date](#_r2kz1olt1jzb) 26

[Print Attendee Badges](#_r2kz1olt1jzb) 27

[**Behavioral Model and Description**](#_xjodq9dbog7m) **27**

[Description for Software Behavior](#_glnxtne9b6cf) 27

[Events](#_v9nmmgdcs0uo) 27

[States](#_o07k6rxzjfft) 28

[State Transition Diagrams](#_xafrrv7emel6) 28

[Control Flow Description](#_awvgmxoe0m91) 30

[**Restrictions, Limitations, and Constraints**](#_4q4yw6qr0rqt) **32**

[**Validation Criteria**](#_5vkgz5i69129) **32**

[Classes of Tests](#_xafrrv7emel6) 32

[Expected Software Response](#_xafrrv7emel6) 33

[Performance Bounds](#_jsrqxs95fwn2) 34

[**Appendices**](#_svgw5hyw0iu4) **34**

[System Traceability Matrix](#_lym04rg9zz33) 34

[Product Strategies](#_5ak048m3sj5c) 35

[Analysis Metrics to be used](#_l43uamgu1v1h) 35

# 

# 

# 

# Introduction (WIP)

## Goals and Objectives

The goal for this software is to allow for users to select dates and register to be a participant in Fords IDTE event, and receive confirmation emails regarding the event. Additionally, each user will have a generated QR code on their ticket that will let them check in when they arrive at the IDTE.

## Statement of Scope

### Functional Requirements

* + - 1. User can register as a supplier, presenter, or evaluator
      2. A waiver needs to be signed before a user can sign up to attend the event
      3. Attendees for the event can modify their registration at any point before an event
      4. Administration confirms a registration from a requested attendee
      5. Administration can view how many attendees there would be on a given day for the event
      6. A superadmin would be created that can have access to administrative accounts
      7. Administration can change dates of events
      8. Administration can open or close the registration dates for events
      9. Email that is sent to the attendee contains a QR code with their information provided during registration
      10. Badges would be printed on the day of the event by administration

### Non-Functional Requirements

* + - 1. Application can run on mobile
      2. Application can run on any web browser
      3. Database is secure and cannot be accessed outside of select individuals
      4. User information for attendees can only be viewed by their own person
      5. Information stored in database is only for IDTE

### Hardware Requirements

* + - 1. Windows OS
      2. Mac OS
      3. Web camera
      4. Mobile Phone

### Software Requirements

* + - 1. Node.Js
      2. HTML
      3. CSS
      4. JavaScript
      5. Java
      6. Sqlite

### Inverse Requirements

### Deliverables:

#### Inputs

* + - * 1. User credentials
        2. First Name
        3. Last Name
        4. Email Address
        5. Association with event
        6. Dates to attend event

#### Processing

* + - * 1. Data from inputs to database
        2. Generate QR code

#### Output

* + - * 1. Email sent to user with QR code

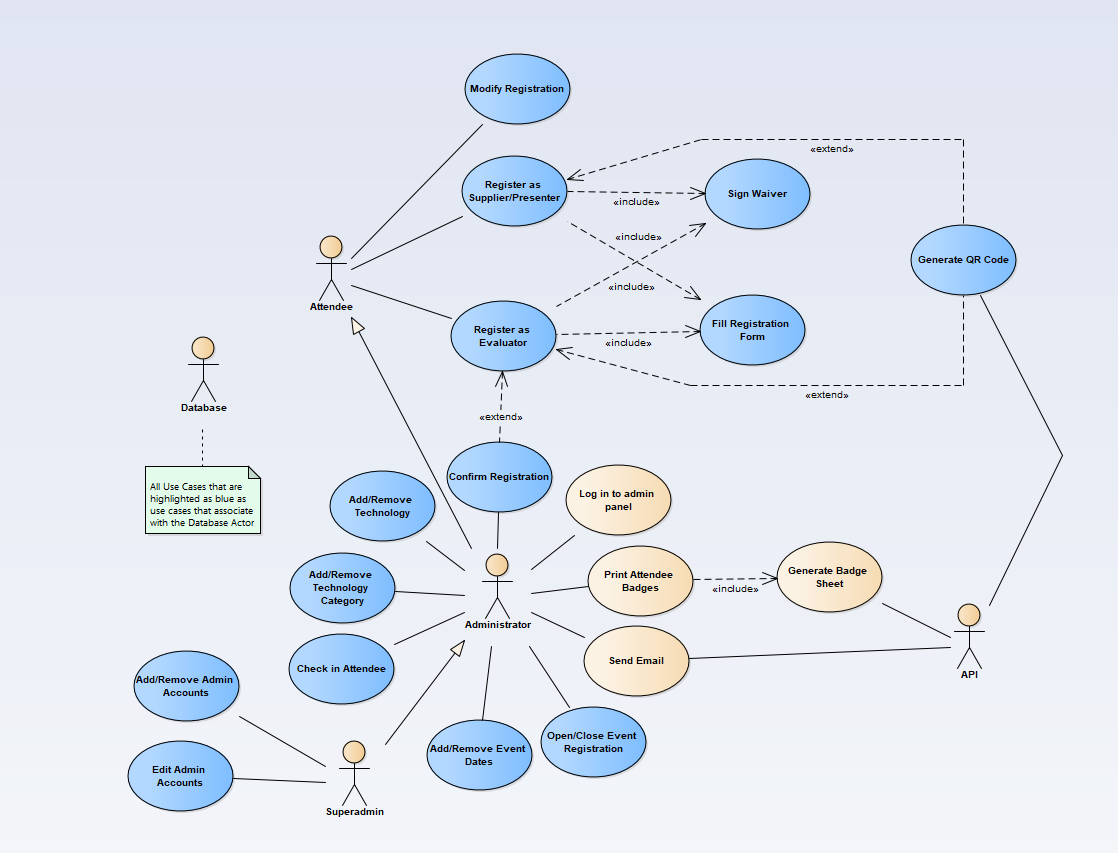
## Software Context

This web based application would allow users to register for the IDTE event hosted by Ford. A user would have to first access the website and sign a waiver. Once that is completed, the user would then register for the event by providing the necessary information as requested for the database system. Upon completion, the user would be sent an email with a generated QR code that would be presented upon the day of the event.

## Major Constraints

Time frame for release of product.

# Usage Scenario



## User Profiles

* + 1. Attendees
       1. Attendees should be able to register for the event and to modify their registration information
    2. Administrators
       1. Administrators are responsible for handling the list of registered attendees, managing the technologies and associated categories, checking in attendees and printing badges, managing event dates, managing event registration availability, sending emails, and logging into the administrative panels
    3. Superadmins
       1. Superadmins can do everything administrators can do, but superadmins can also manage administrative accounts
    4. API
       1. The API handles all database connections, as well as some backend processing tasks such as generating QR codes, generating the printable badge sheets, and sending out emails as the administrators writes them
    5. Database for Registration
       1. The database is responsible for storing data to be used in any of the other use cases. Talks to database for technologies when user is registering for event.
    6. Database for Technologies
       1. The database is responsible for storing data to be used for technologies that would be read by managers for approval prior to user registration

## User Story Descriptions

* + 1. Register a Technology
       1. Primary Actor:
       2. Preconditions
       3. Description
       4. Acceptance Criteria:
    2. Pull list of Technologies
       1. Primary Actor:
       2. Preconditions:
       3. Descriptions:

## Register as Evaluator

* + - 1. **Primary Actor**: Attendee / User.
      2. **Preconditions**: User is on website interface.
      3. **Description**: User chooses to register as an Evaluator on the Main menu, then proceeds to fill out the registration form. Before registration is complete the User must receive confirmation from an admin.
      4. **Acceptance Criteria**: User is successfully registered and confirmed as an evaluator.

## Register as Supplier/Presenter

* + - 1. **Primary Actor**: Attendee / User.
      2. **Preconditions**: User is on website interface.
      3. **Description**: User chooses to register as a supplier or presenter on the Main menu, then proceeds to fill out the registration form.
      4. **Acceptance Criteria**: User is successfully registered as a supplier or presenter.

## Modify Registration

* + - 1. **Primary Actor**: Attende / Admin / User.
      2. **Preconditions**: User is on website interface.
      3. **Description**: User chooses to modify an existing registration. User searches for themself and then proceeds to make changes to their own registration.
      4. **Acceptance Criteria**: User modifications are completed and successfully stored.

## Sign Waiver

* + - 1. **Primary Actor**: Attendee / User.
      2. **Preconditions**: User is on website and started the registration process.
      3. **Description**: User starts registration and before they can fill out the form they are required to read and sign the provided waiver.
      4. **Acceptance Criteria**: User has read and signed the waiver.

## ~~Fill Registration Form~~

* + - 1. **~~Primary Actor~~**~~: Attendee / User.~~
      2. **~~Preconditions~~**~~: User is on website and started the registration process. User has Signed the waiver.~~
      3. **~~Description~~**~~: User signed waiver and continued to the registration form. User fills out this form correctly and submits.~~
      4. **~~Acceptance Criteria~~**~~: Registration is correctly entered and successfully submitted.~~

## Confirm Registration

* + - 1. **Primary Actor**: Administrator / Admin.
      2. **Preconditions**: A user has attempted to register as an evaluator and the Admin has logged in.
      3. **Description**: Administrator looks over evaluator registration form and verifies the information.
      4. **Acceptance Criteria**: Evaluator registration form is reviewed and a decision is made whether the registration is valid or not.

## ~~Log In to Admin Panel~~

* + - 1. **~~Primary Actor~~**~~: Administrator / User.~~
      2. **~~Preconditions~~**~~: User has navigated to website home.~~
      3. **~~Description~~**~~: User enters their credentials into the login fields. Credentials are verified.~~
      4. **~~Acceptance Criteria~~**~~: User is logged into the website as an admin and now has access to admin functionality.~~

## Print Attendee Badges

* + - 1. **Primary Actor**: Administrator / Admin.
      2. **Preconditions**: Administrator is logged in.
      3. **Description**: Administrator chooses which group or individual to print a badge for. Badges are printed.
      4. **Acceptance Criteria**: Badges for selected group are printed successfully.

## Send Email

* + - 1. **Primary Actor**: Administrator / Admin.
      2. **Preconditions**: Admin wants to send an email to a selection of attendees. Admin is logged in.
      3. **Description**: Administrator selects option to send email and then selects which group/individual to send an email to. Admin then sends the email.
      4. **Acceptance Criteria**: Email is received by the target audience.

## Open/Close Event Registration

* + - 1. **Primary Actor**: Administrator / Admin.
      2. **Preconditions**: Admin wants to stop or start allowing users/attendees to register for the event. Admin is logged in as admin.
      3. **Description**: Administrator selects option to open/close the registration. Then admin chooses the option they like
      4. **Acceptance Criteria**: Email is received by the target audience.

## Add/Remove Event Dates

* + - 1. **Primary Actor**: Administrator / Admin.
      2. **Preconditions**: Admin wants to create/remove events that are available to sign up for. Admin is logged in as admin.
      3. **Description**: Administrator selects option to add/remove the registration. Then admin chooses the option they like and fills out required information.
      4. **Acceptance Criteria**: Event is successfully created and users can register for it, or event is successfully removed.

## Check in Attendee

* + - 1. **Primary Actor**: Administrator / Admin.
      2. **Preconditions**: Attendee has arrived for event. Admin is logged in.
      3. **Description**: Administrator selects option to check in an attendee. Then the admin opens the web interface and checks in the Attendee via using a camera for the QR code, or searching the attendee by name.
      4. **Acceptance Criteria**: Attendee QRcode/Registration info is identified and the Attendee is able to go to the event.

## Add/Remove Technology Category

* + - 1. **Primary Actor**: Administrator / Admin.
      2. **Preconditions**: New technology needs to be added or removed. Admin is logged in.
      3. **Description**: Administrator selects option to add/remove a technology category. Technology category information is filled in. New technology category added or old technology category is deleted.
      4. **Acceptance Criteria**: Technology category is now part of list or not on list anymore.

## Add/Remove Technology

* + - 1. **Primary Actor**: Administrator / Admin.
      2. **Preconditions**: New technology category needs to be added or removed. Admin is logged in.
      3. **Description**: Administrator selects option to add/remove a technology. Technology information is filled in. New technology added or old technology is deleted.
      4. **Acceptance Criteria**: Technology is now part of list or not on list anymore.

## Add/Remove Admin Account

* + - 1. **Primary Actor**: Superadmin.
      2. **Preconditions**: Superadmin is logged in.
      3. **Description**: Superadmin chooses to add or remove admin accounts. Account information is filled out. Account gets added or removed.
      4. **Acceptance Criteria**: New admin account exists or old admin account does not exist.

## Edit Admin Accounts

* + - 1. **Primary Actor**: Superadmin.
      2. **Preconditions**: Superadmin is logged in. Admin accounts exist to be edited.
      3. **Description**: Super admin chooses to edit admin accounts. Account id/name is entered. New account form is filled out.
      4. Acceptance Criteria: Modifications are successfully saved on the account.

## Generate Badge Sheet

* + - 1. **Primary Actor**: Administrator / API.
      2. **Preconditions**: Accounts exist in order to print badges. Admin is logged in.
      3. **Description**: Administrator selects option to print attendee badges. Upon printing, the API generates a Badge sheet of the selected group.
      4. **Acceptance Criteria**: Badge sheet is generated correctly.

## Generate QR Code

* + - 1. **Primary Actor**: API.
      2. **Preconditions**: Attendee has registered.
      3. **Description**: Attendee registers. API generates a unique QR code for the new attendee and stores it.
      4. **Acceptance Criteria**: QR code is unique and correct.

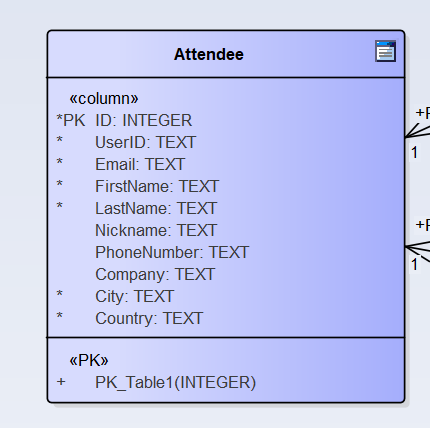
## Special Usage Considerations

* + 1. Work within the confines of Ford’s internal security system

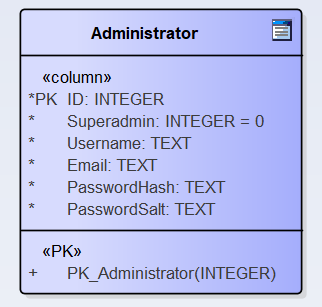
# Data Model and Description

## Data Objects

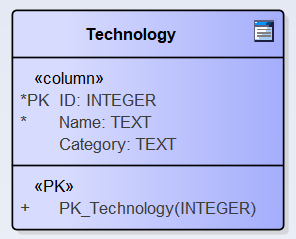
* + 1. Attendees
       1. Stores basic information about the attendee, like email, first name, last name…



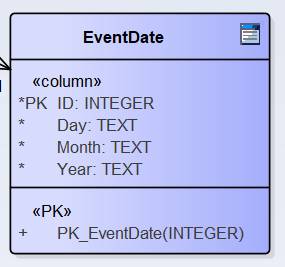
* + 1. Administrators
       1. Stores the administrative username, email for that administrator, the password hash and salt, and a boolean stating whether the admin account is a super admin



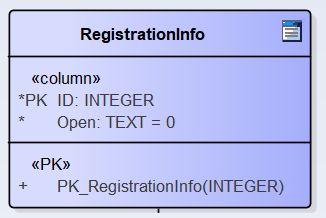
* + 1. Technologies
       1. Technologies are stored with a name describing them or ID, and a category to which they should fall under



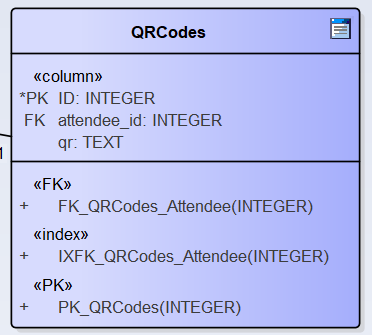
* + 1. Event Dates
       1. Event dates store the day, month, and year of the event



* + 1. Registration Info
       1. This table will act as a single row table in order to store and access global information. At this point the only column is used to keep track of when registration is open or closed

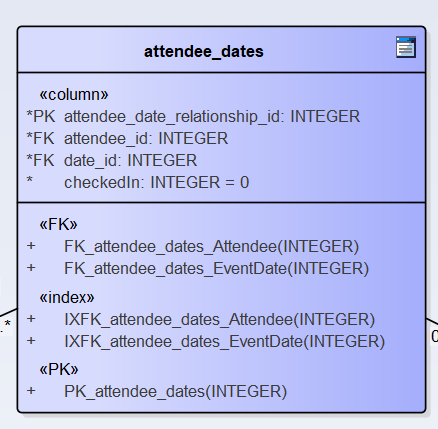


* + 1. QR Codes
       1. The QR Code table will store the information for the QR codes, and how they are related to the attendees they associate with



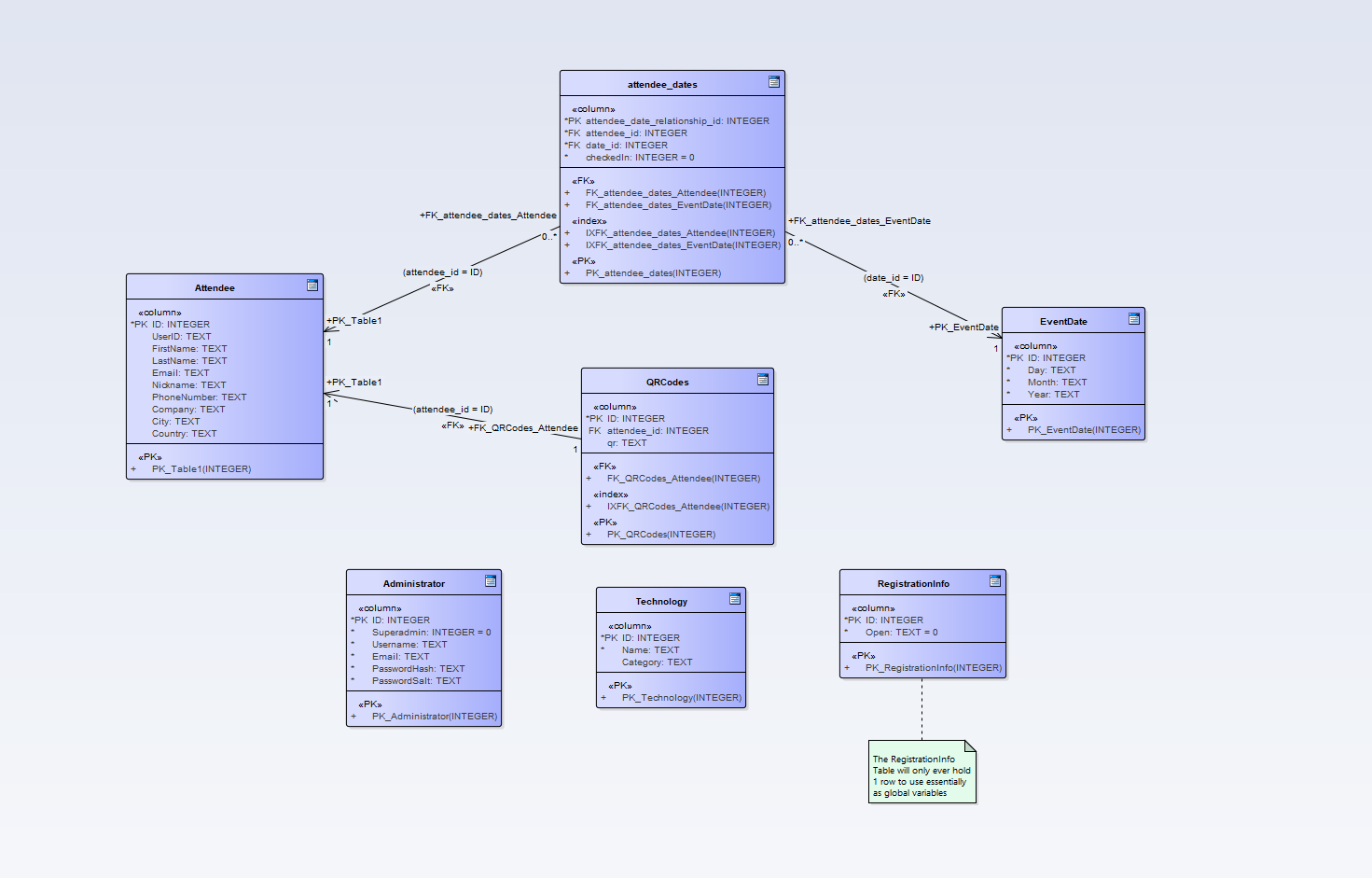
## Relationships

* + 1. Attendee Dates
       1. An attendee can attend any number of event dates, and any event date can have any number of attendees
       2. The relationship between dates and attendees also keeps track of the attendee has attended the event on that date



* + 1. QRCode Attendee
       1. Every QR Code is associated with exactly one attendee. This relationship will also keep track of the QR code information.

## Data Model



# Functional Model and Description

## Description for Function n (PSPEC)

Refer to Section 2 Use Case Diagram for use case relationships

## Register as Evaluator

* + - 1. **Actors:** Attendee, API, Database, Administrator
      2. **Pre-Conditions:** User is on website homepage
      3. **Triggers:** User clicks on “Register as Evaluator” button
      4. **Scenario Description:** After clicking on the register button, the user will be prompted with a waiver that they will need to agree to and sign.   
           
         After signing, the user will be prompted to enter the following information: email address\*, first name\*, last name\*, preferred nickname, phone number, company, city\*, and country\*  
           
         \* Denoting required field
      5. **Post-Conditions:** After the information is submitted, an administrator will be required to confirm the registration for evaluators. Upon this confirmation, the system will generate a QR code for the user and store all the information in the database. A confirmation email will be sent with the QR code and user ID
      6. **Exceptions:** If the administrator denies the registration, the information will not be committed to the database. If there are invalid formats in any of the form fields, upon submission of the form a message will pop up prompting the user to fix the error.

## Register as Supplier/Presenter

* + - 1. **Actors:** Attendee, API, Database
      2. **Pre-Conditions:** User is on website homepage
      3. **Triggers:** User clicks on “Register as Supplier” button or “Register as Presenter” button
      4. **Scenario Description:** After clicking on the register button, the user will be prompted with a waiver that they will need to agree to and sign.   
           
         After signing, the user will be prompted to enter the following information: email address\*, first name\*, last name\*, preferred nickname, phone number, company, city\*, and country\*  
           
         \* Denoting required field
      5. **Post-Conditions:** After the information is submitted, the system will generate a QR code for the user and store all the information in the database. A confirmation email will be sent with the QR code and user ID
      6. **Exceptions:** If there are invalid formats in any of the form fields, upon submission of the form a message will pop up prompting the user to fix the error.

## Modify Registration

* + - 1. **Actors:** Attendee, Database
      2. **Pre-Conditions:** User is on website homepage
      3. **Triggers:** User clicks “Modify Registration” button
      4. **Scenario Description:** After clicking the button, the user will be brought to a screen where they can search for their registration information by either entering their name or their user ID.  
           
         After the user finds their registration, they can modify it by going through the registration form again.
      5. **Post-Conditions:** Upon submission, the database will be updated and a new QR code will be generated. A confirmation email will be sent with the QR code and user ID
      6. **Exceptions:** If there are invalid formats in any of the form fields, upon submission of the form a message will pop up prompting the user to fix the error.

## Sign Waiver

* + - 1. **Actors:** User
      2. **Pre-Conditions:** User is registering for the event
      3. **Triggers:** User has clicked on the registration button
      4. **Scenario Description:** Upon clicking the button to register, the first menu that will come up will be the waiver. The user will need to read through the agreement, check off a checkbox indicating the user agrees, sign the form, and submit.
      5. **Post-Conditions:** When the waiver is submitted successfully, the user will be brought to the registration form.
      6. **Exceptions:** If the user does not agree to the waiver, they will not be able to continue registration until they agree.

## Fill Registration Form

* + - 1. **Actors:** User
      2. **Pre-Conditions:** User is registering for the event
      3. **Triggers:** User has signed off for the waiver
      4. **Scenario Description:** When the user has signed the waiver, they will be asked to enter: email address\*, first name\*, last name\*, preferred nickname, phone number, company, city\*, and country\*
      5. **Post-Conditions:** The user will then submit the information which will either be forwarded to an admin for confirmation or directly committed to the registration database
      6. **Exceptions:** If there are invalid formats in any of the form fields, upon submission of the form a message will pop up prompting the user to fix the error.

## Confirm Registration

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** User has submitted registration for event
      3. **Triggers:** Administrator navigates to registration confirmation dialog
      4. **Scenario Description:** An administrator will need to confirm the registration on an individual basis for each submitted evaluator registration.
      5. **Post-Conditions:** When an administrator confirms the registration, the registration information will be committed to the database
      6. **Exceptions:** None

## Log In to Admin Panel

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** Admin is on home page
      3. **Triggers:** Admin navigates to admin page and is prompted to log in
      4. **Scenario Description:** When an administrator tries to access the administration page without being logged in, they will be prompted to do so. They will have to enter their administrative username and password, and the database will confirm the credentials.
      5. **Post-Conditions:** After submitting the login request, the administrator will be able to access administrative functions
      6. **Exceptions:** If the credentials entered are not valid, the user will not be able to access administrative functionality

## Print Attendee Badges

* + - 1. **Actors:** Administrator, Database, API
      2. **Pre-Conditions:** User is on admin page
      3. **Triggers:** User clicks on print badges button
      4. **Scenario Description:** The administrator will be brought to a page where they will be prompted to either print all the badges, all the badges for a type of attendee, or a single badge for an individual attendee. The admin will be able to search for the attendee by their user ID or their name.  
           
         After the admin has selected in which manner to print the badges, the API will generate the badge sheet and send it back to the admin to be printed.
      5. **Post-Conditions:** The badges will be printed onto 8.5x11 printer paper with 4 badges to a page.
      6. **Exceptions:** None

## Send Email

* + - 1. **Actors:** Administrator, API
      2. **Pre-Conditions:** User is on admin page
      3. **Triggers:** User clicks on the Send Email button
      4. **Scenario Description:** When the button is clicked, a dialog will be brought up prompting for a Subject and Body. These will be the fields for the email sent out. The user can choose to send the email to every attendee, to attendees in a specific group, or to a specific attendee.
      5. **Post-Conditions:** When the form is submitted, emails will be sent to the appropriate attendees by the API.
      6. **Exceptions:** None

## Open/Close Event Registration

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** User is on admin page
      3. **Triggers:** User clicks on the Open/Close Registration button
      4. **Scenario Description:** When the button is clicked, the user will be brought to a page with 2 buttons: Open Registration and Close Registration. Clicking on the open button will allow attendees to start registering while clicking on the close button will make registration unavailable for the attendees.
      5. **Post-Conditions:** Registration will be enabled or disabled based on the button pressed.
      6. **Exceptions:** None

## Add/Remove Event Dates

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** User is on admin page
      3. **Triggers:** User clicks on the Add/Remove Event Dates button
      4. **Scenario Description:** When the button is clicked, the user will be brought to a page where they can add a new event date to the list, or select a date from a drop down menu to be edited/deleted.
      5. **Post-Conditions:** When the form is submitted, the appropriate event will either be added to the list or deleted from the list
      6. **Exceptions:** None

## Check in Attendee

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** User is on admin page
      3. **Triggers:** User clicks on the Check in Attendee button
      4. **Scenario Description:** When the button is clicked, the user will be prompted to either check in attendees using their QR code, or by searching their name.
      5. **Post-Conditions:** When the form is submitted, the relevant attendee will be marked as having attended that day of the event
      6. **Exceptions:** If there user is not registered for that event day, an error message will be displayed. An error message will also be displayed if the QR code could not be read successfully, or if the QR code is not for the event.

## Add/Remove Technology Category

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** User is on admin page
      3. **Triggers:** User clicks on the Add/Remove Technology button
      4. **Scenario Description:** The user will be brought to the add/remove technology page. From here they will be able to add a technology category, or to remove a technology category.
      5. **Post-Conditions:** If the user chose to add a category, the category will be added to the database, and if the user chose to delete a technology category, that category will be deleted from the database.
      6. **Exceptions:** None

## Add/Remove Technology

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** User is on admin page
      3. **Triggers:** User clicks on the Add/Remove Technology button
      4. **Scenario Description:** The user will be brought to the add/remove technology page. From here they will be able to add a technology to a category, or to remove a technology.
      5. **Post-Conditions:** If the user chose to add a technology, the technology will be added to the database with a given category, and if the user chose to delete a technology, that technology will be deleted from the database.
      6. **Exceptions:** None

## Add/Remove Admin Account

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** A superadmin is on admin page
      3. **Triggers:** User clicks on the Manage Admin Accounts button
      4. **Scenario Description:** The user will be brought to a page where they will be able to add, edit, or remove administrative accounts.   
           
         If they choose to add an account, the user will be prompted to enter a username, email address, and password. The password will be secured by hashing and salting. The user can also specify if the new account should be a superadmin.  
           
         If they choose to remove an account, the user will be prompted with a list of administrative accounts or a search bar to search by admin username. From here the user will be able to select an account for deletion, and will be prompted for confirmation.
      5. **Post-Conditions:** If a new admin account is created, it will be added to the database. If an account is marked for deletion, it will be removed from the database.
      6. **Exceptions:** Two accounts cannot share email addresses or usernames, so an error will be displayed if an account is created to have a duplicate name or email address.

## Edit Admin Accounts

* + - 1. **Actors:** Administrator, Database
      2. **Pre-Conditions:** User is on admin page
      3. **Triggers:** User clicks on the Manage Admin Accounts button
      4. **Scenario Description:** The user will be brought to a page where they will be able to add, edit, or remove administrative accounts.   
           
         If they choose to edit an account, the same list/search bar for removal will be shown, but this time selection will lead to a prompt to edit the details of the account.
      5. **Post-Conditions:** If an account is edited, the changes will be committed to the database.
      6. **Exceptions:** Two accounts cannot share email addresses or usernames, so an error will be displayed if an account is edited to have a duplicate name or email address.

## Generate Badge Sheet

* + - 1. **Actors:** API
      2. **Pre-Conditions:** API is ready for a request
      3. **Triggers:** An administrator requests attendee badges to be printed
      4. **Scenario Description:** When the admin requests the badges to be printed, the API will generate the badges to fit the specified format and will lay out 4 badges to a page for 8.5x11 paper
      5. **Post-Conditions:** The generated badges will be returned to the caller to be printed
      6. **Exceptions:** None

## Generate QR Code

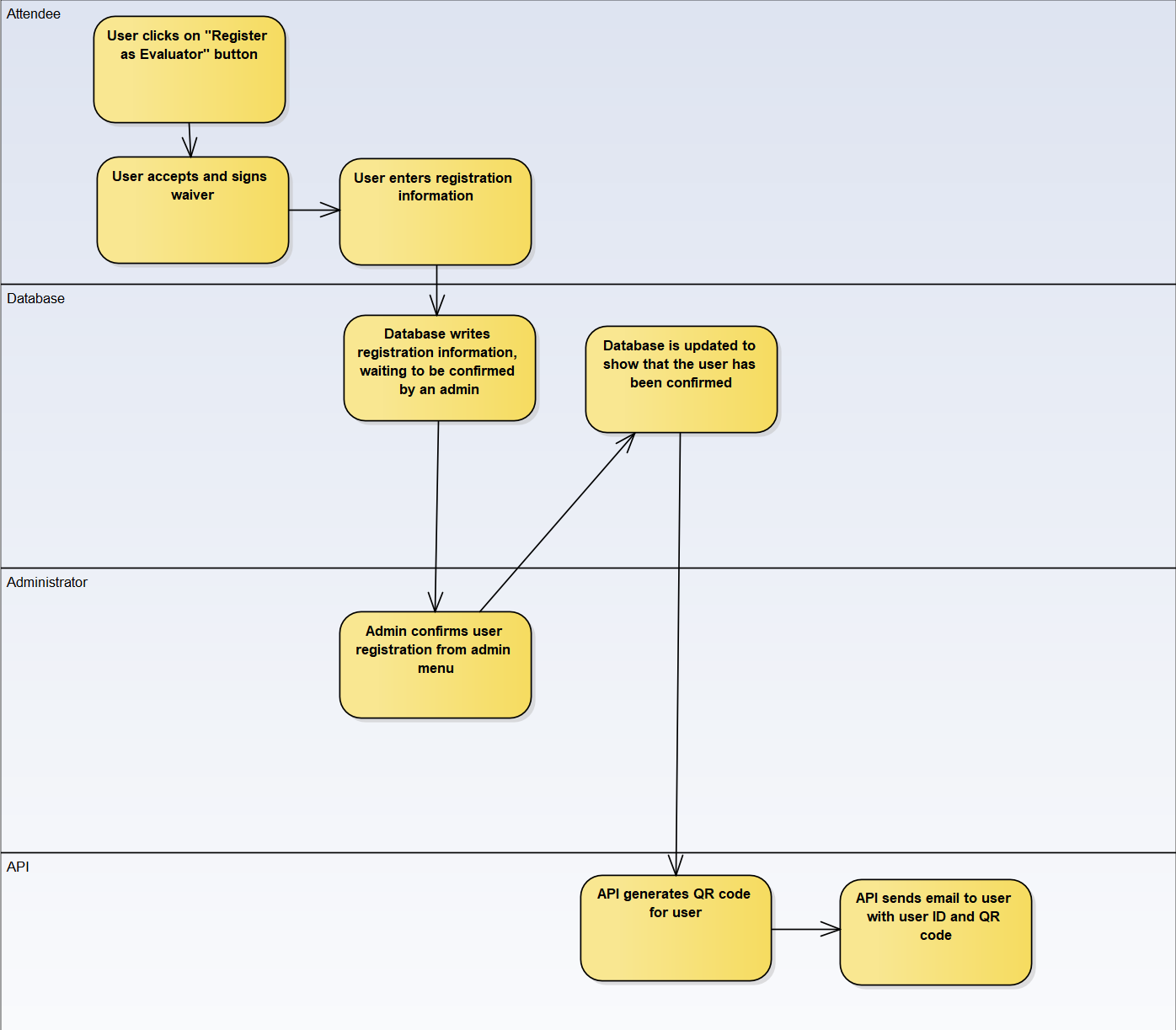
* + - 1. **Actors:** API
      2. **Pre-Conditions:** API is ready for a request
      3. **Triggers:** A user is successfully registered
      4. **Scenario Description:** When a user has completed registration, and they are a supplier/presenter or they have been confirmed for registration, a QR code will be generated for the given user’s information.
      5. **Post-Conditions:** The generated QR code will be stored in the database and sent in an email to the attendee.
      6. **Exceptions:** None

## Software Interface Description

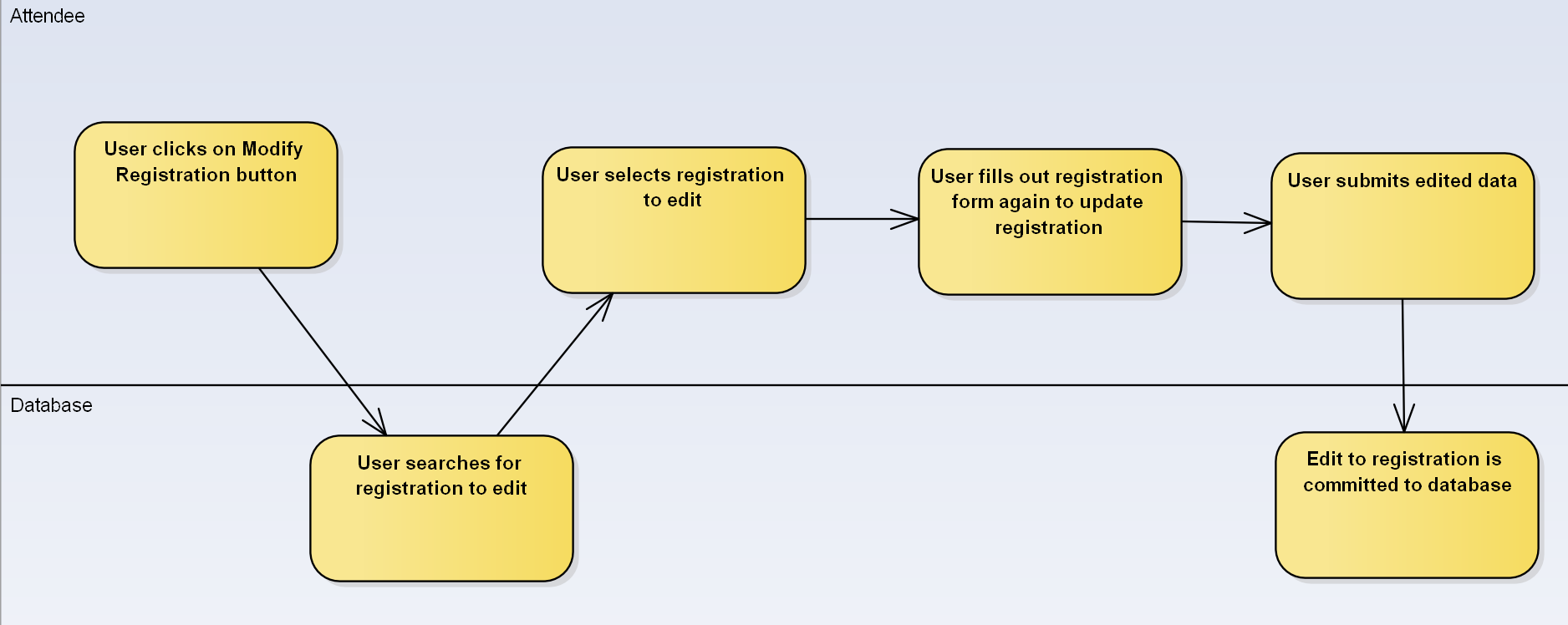
* + 1. Human Interface
       1. Users will be able to interact with the software using any device with a web browser. The software will be designed to be compatible with modern widescreen monitors as well as mobile device touchscreens.
    2. API/Database
       1. The API will run on the server backend waiting for requests sent in from client side code or other API functions.
       2. The API will also handle all database interactions between client and server

## Activity Diagrams for Core Functionality

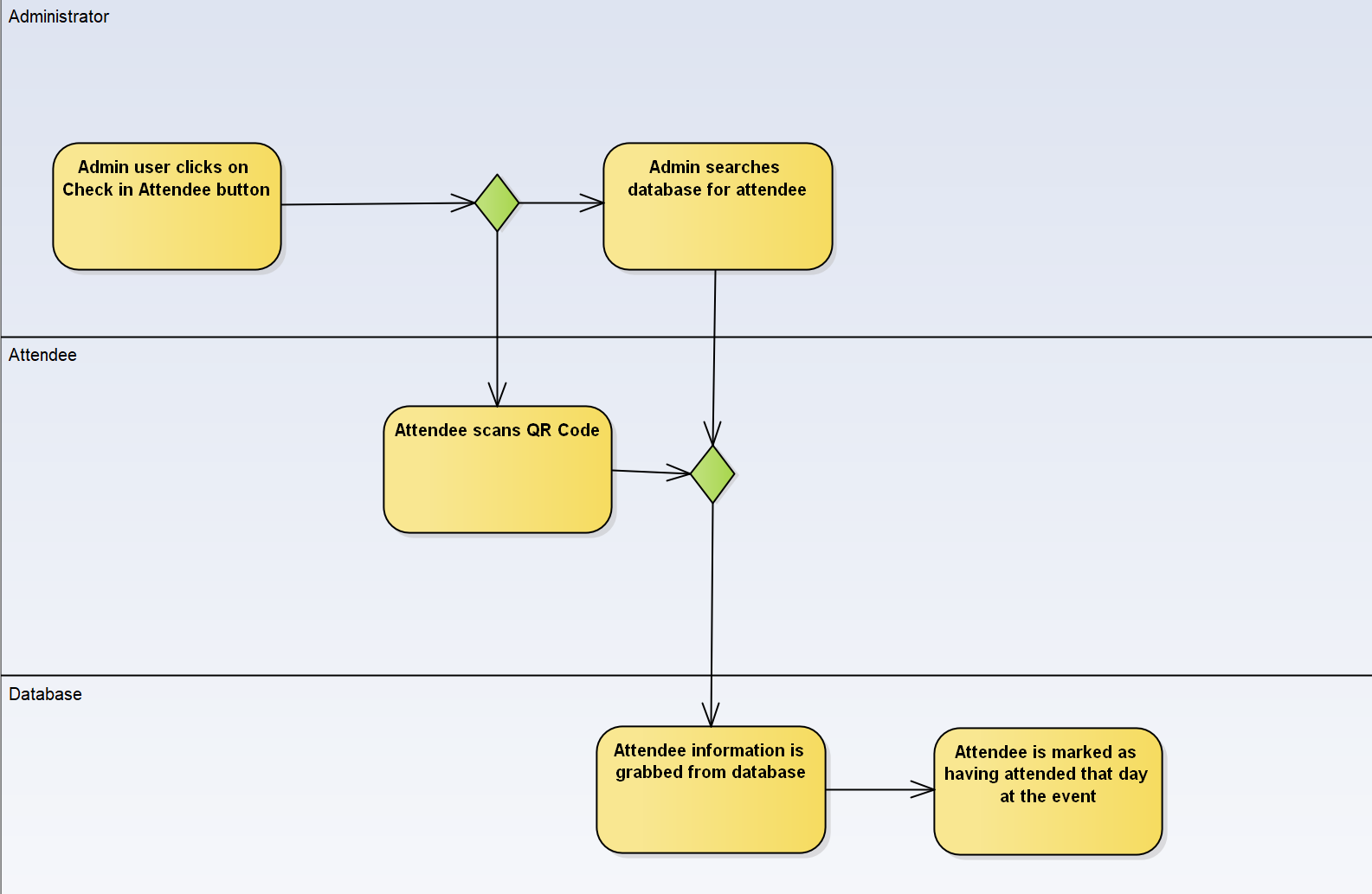
## Attendee Registration



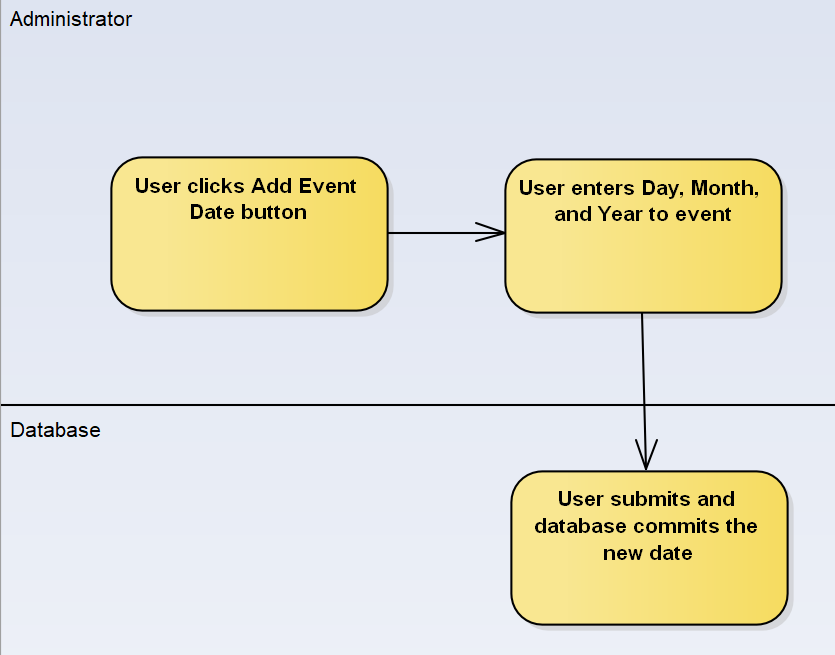
## Modify Registration



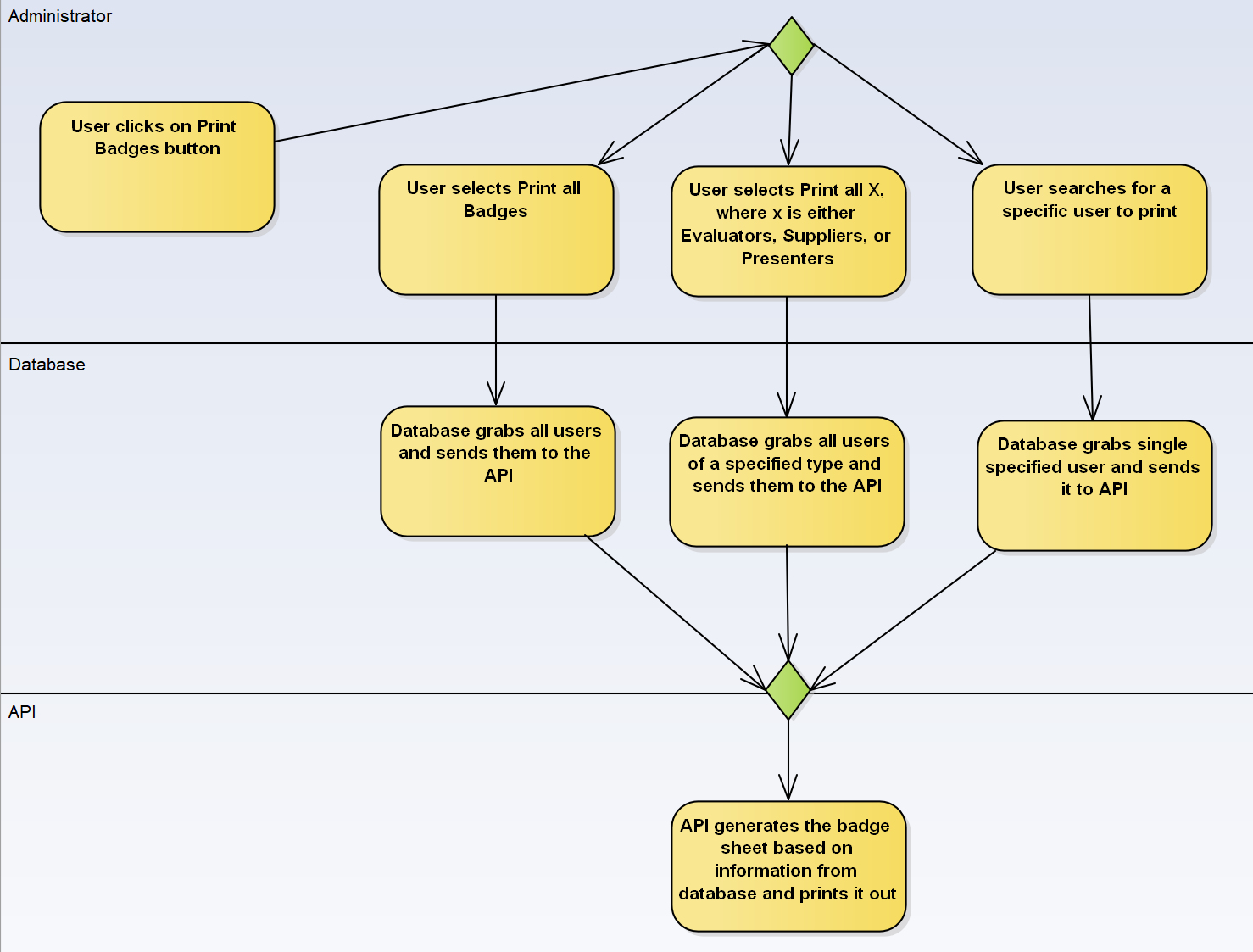
## Checking in Attendee



## Add Event Date



## Print Attendee Badges



# Behavioral Model and Description

## Description for Software Behavior

### Events

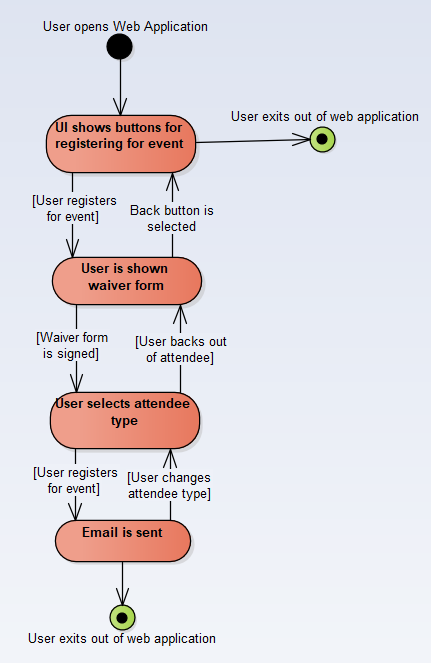
* + - 1. User launches Web application
      2. User selects Register for event
      3. User signs the waiver form
      4. User selects to attend the event as a Supplier
         1. User selects to attend the event as a Presenter
         2. User selects to attend the event as an Evaluator
      5. User fills out the registration form
      6. Administrator account verifies attendee registration
      7. User receives email with QR code
      8. User modifies registration information
         1. Updated email is sent out with QR code and notifying user of updated changes

### States

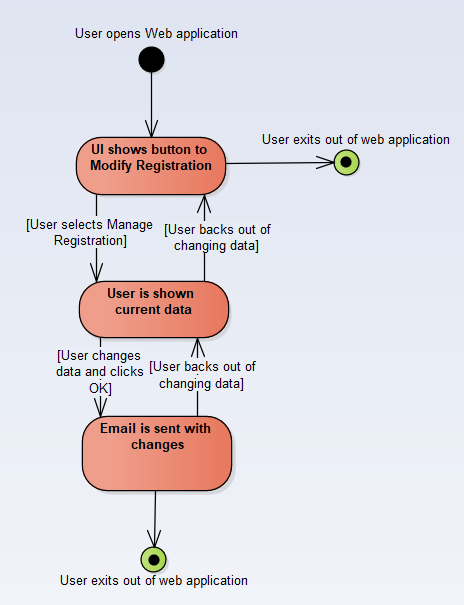
* + - 1. UI shows buttons for Evaluator, Supplier, Presenter, and Modify Registration
      2. Prompt to sign waiver is shown
      3. Text fields that are required by user input are shown
      4. Registration is filled out
      5. Administrator is shown a notification of registered attendee
      6. Email is sent to attendee with QR code attached

## State Transition Diagrams

**State Diagram: User Registers for event**

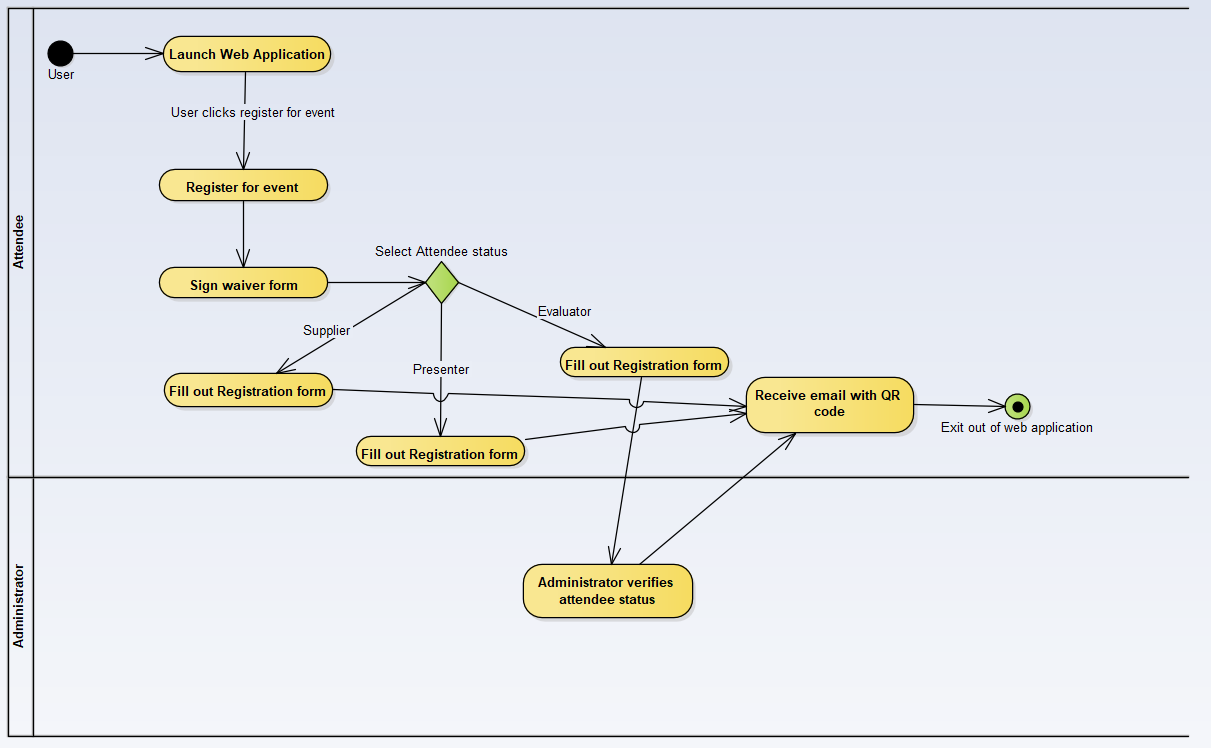
****

**State Diagram: User modifies registration**

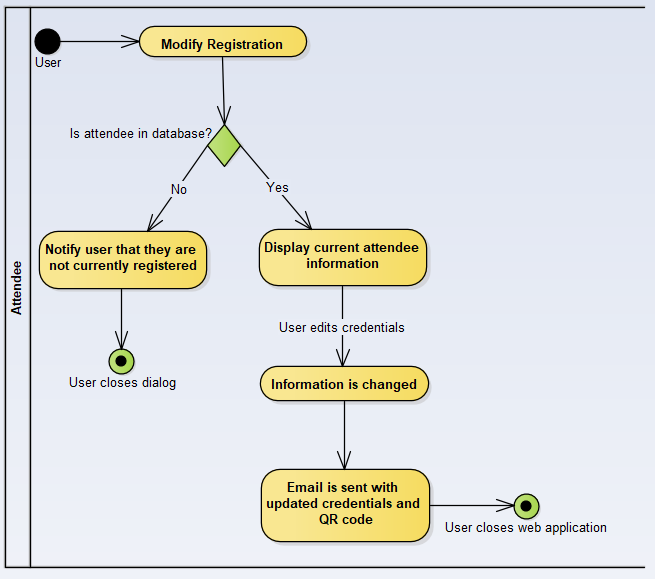
****

## Control Flow Description

**Activity Diagram: User Registers for event**

****

**Activity Diagram: User Modifies Registration**



# Restrictions, Limitations, and Constraints

* 1. A proposed earlier date for the project may prove difficult to meet each respective requirement for the software. Additionally, the expedited schedule may sacrifice some overall software stability for an earlier delivery.
  2. Due to the single threaded nature of Node, CPU heavy processes may slow down the software as a whole while new functions are queued. This also means that if a part of the software causes Node to crash, all requests on the thread will crash as well.
  3. Each team member will be using his own personal computer when working on the project, and as such each machine may perform differently when trying to use the software.

# Validation Criteria

## Classes of Tests

* + 1. Logging in
       1. User goes to log in page
       2. User enters username and password
       3. User clicks “Submit”
    2. Registering for an Event
       1. Once logged in on the home screen, user selects their registration type.
       2. User signs waiver and agrees to the terms and conditions and clicks “Next”
       3. User enters personal information and clicks “Next”
       4. User selects the dates they will be registering for
       5. For each respective date, the user selects a technology number.
       6. User clicks “Next”
       7. A summary of the entered information is displayed and verified by the user for it’s accuracy; once the user is satisfied with the entered information, they click “Submit”.
    3. Changing Registration Info
       1. Once logged in on the home screen, user selects “Modify Registration”.
       2. User searches for their name and clicks “Submit”.
       3. User updates any information as needed, and clicks “Submit”
    4. Deleting Attendee
       1. Once logged in on the home screen, user selects “Modify Registration”.
       2. User searches for their name and clicks “Submit”.
       3. User clicks “Delete Attendee”
       4. User confirms deleting attendee
    5. Attendee Check-in
       1. Once logged in on the home screen, user selects “Admin”.
       2. User selects “Attendee Check-In”
       3. User searches for attendee information and selects “Submit”, or scans the QR code on the attendees ticket.
    6. Adding Technologies
       1. Once logged in on the home screen, user selects “Admin”.
       2. User selects “Technologies”.
       3. User enters technology number and technology name.
       4. User selects “Submit”.
    7. Adding Categories
       1. Once logged in on the home screen, user selects “Admin”.
       2. User selects “Categories”.
       3. User enters category and selects “Submit”.
    8. Setting Registration Dates
       1. Once logged in on the home screen, user selects “Admin”.
       2. User selects “Set Event Dates”
       3. User selects months and days for the dates.
       4. User selects “Submit”

## Expected Software Response

* + 1. Logging in
       1. Login will be granted upon verifying the entered credentials were correct. If the credentials were invalid, the user will not be logged in.
    2. Registering for an event
       1. User’s information will be recorded and the individual will be registered for the entered dates.
    3. Changing Registration info.
       1. The desired information will be updated according to the users needs.
    4. Deleting Attendee
       1. The user will no longer be registered for the entered dates, and registration information for the individual will be removed from the database.
    5. Attendee Check-in
       1. User will be checked into the event if they had registered prior to attempting to check-in.
    6. Adding Technologies
       1. A new technology will be added and available for selection by users when registering.
    7. Adding Categories
       1. A new category will be added and available for selection by users when registering.
    8. Setting Registration Dates
       1. Dates for the event will be set and able to be selected by attendees when registering.

## Performance Bounds

* + 1. 7.1.3 and 7.1.4 will also be used to test the software’s response for when an attendee’s information cannot be found within the database.

# Appendices

## System Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| U/R | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 | R10 |
| US1 | X | X |  |  |  |  |  |  |  |  |
| US2 | X | X |  |  |  |  |  |  |  |  |
| US3 |  |  | X |  |  |  |  |  |  |  |
| US4 |  | X |  |  |  |  |  |  |  |  |
| US5 | X | X |  |  |  |  |  |  |  |  |
| US6 | X | X |  | X |  |  |  |  |  |  |
| US7 |  |  |  | X | X |  | X |  |  |  |
| US8 |  |  |  |  |  |  |  |  |  | X |
| US9 |  |  |  |  |  |  |  |  | X |  |
| US10 |  |  |  |  |  |  |  | X |  |  |
| US11 |  |  |  |  |  |  | X |  |  |  |
| US12 |  |  |  |  |  |  |  |  | X | X |
| US13 |  |  |  |  |  |  | X | X |  |  |
| US14 |  |  |  |  |  |  | X | X |  |  |
| US15 |  |  |  |  |  | X |  |  |  |  |
| US16 |  |  |  |  |  | X |  |  |  |  |
| US17 |  |  |  |  |  |  |  |  | X |  |
| US18 |  |  |  |  |  |  |  |  | X |  |

## Product Strategies

* + 1. The software is designed to be used for Ford’s IDTE event. This is to allow attendees to register for the event and have an organized way of making sure people who attend are supposed to be there on a given day. On the days of events, a badge would be printed out for each attendee.

## Analysis Metrics to be used

* + 1. Data that is compiled through the database by user registration can show how many people show up to the event every year. It would also give an idea to what type of attendees show up whether it be presenters, evaluators, or suppliers.

# 

|  |  |  |  |
| --- | --- | --- | --- |
| Review and Signoff of the Software Requirements Specification | | | |
| LAST NAME | FIRST NAME | SIGNATURE | DATE |
| Achtman | Maxx |  |  |
| Ciolek | Christopher |  |  |
| Spencer | Nicholas |  |  |
| Cazares | Apolonio |  |  |
| Belwafa | Jamel |  |  |
|  |  |  |  |
|  |  |  |  |