

**Portland State University - Oregon MESA**

# **MESA EVERYDAY**

## **Design Document**

**Christopher Bartlett, Michael Cohoe, Fadi Labib, Minwei Luo, Minh Nugyen, Thong Tran, and Millen Wan**

## Table of Contents

Introduction.....	4
Purpose and Scope .....	4
Target Audience.....	4
Design Consideration.....	4
Constraints and Dependencies .....	4
Methodology .....	5
Database Design.....	5
Database Overview .....	6
Specific Database Considerations.....	6
System Overview .....	7
Landing Page .....	7
Resetting Password Pages.....	7
Send Resting Password Link Page.....	7
Resetting the Password from Link Page .....	8
Forgot Username Page.....	8
Send Email with Username Page.....	8
Participant's Side Overview .....	9
Participant Side Layout.....	9
Side Navigation Bar Section.....	9
Top Navigation Bar Section.....	10
Right Side Calendar Section .....	10
Right Side Events Section.....	10
Main Section .....	11
Footer.....	15
Administrator Side Overview .....	15
Administrator Side Layout.....	15
Side Navigation Bar Section.....	15
Top Navigation Bar Section.....	16
Main Section .....	16
System Design .....	19
Participants Design .....	19
Participants' Session Management Design.....	19

Participants Tasks Design .....	21
Participants Calendar Design .....	23
Administrator Design .....	24
Administrator Session Management Design .....	24
Administrator Tasks Design.....	25

## Introduction

MESA Everyday is a web app that gamifies the MESA experience for students who are part of the program. It will help MESA collect core data about student experiences and their MESA journey to give testimonials that are more concrete for grant funders as well as help students participate in more MESA sponsored activities and events that might allow them to get college credits for work that they have done. This document includes the design consideration that is broken down into Constraints and Dependencies and Methodology. From there the document introduces the system overview showing the design of all the different pages in the web app for both the participant and administrator sides of the web app. The document then discusses the general design consideration for each functionality that the web app provides be it an administrator or participant functionality. Finally, the document goes into the algorithm and coding level details of how these functionalities are implemented.

## Purpose and Scope

This document services as a guide to the developers on the design process for the web app. It starts from a general layout of the different pages in the web app to algorithm and coding level details for all the functionalities that the developers will translate into actual code. In this document, the target audience should be able to learn the detailed specification and design of the different functionalities.

## Target Audience

This document targets the developers of the web-app as well as any future software development group that will pick up this project or the person who will be in charge of maintaining the code for the web app after the developers' capstone terms end. This document also targets all faculty members who are part of Portland State University (PSU) Computer Science (CS) Capstone.

## Design Consideration

This section will explain the design choices the developers will take into consideration in the development of this web app as well as the methodology the will be using to implement the web app.

## Constraints and Dependencies

The web app must be designed in a way that is user friendly to students that are as young as sixth graders. It must also be able to adapt to different screen sizes so that participants and administrator can view their web apps through smartphones and tablets and not just computers and laptops with large screens. The web app will use MESA logo, high quality photographs of MESA student that have been taken by MESA representatives during MESA sponsored events, and pre-built design for the different badges that participants can earn. MESA colors will be used in the web app. The colors that will be used in the web-app are the following: Quantum Orange (0xff9e15), Thermal Red (0xea4e46), Atomic Green (0xbcd62f), Wave Blue (0x8195b1), Raw Umber (0x716558), and Sand (0xe7e0d7). For fonts, the following font variations will be used: Merriweather Light, Merriweather Regular, and Merriweather Bold.

All the web app pages rough layout must be built before content is added to them. While colors and pictures are not important to get correctly at the beginning stage, the layout of the different content in the different pages as well as navigation bars must be in place before implementing many of the features that the web app will support. Participants' rough layout must be in place before implementing participants'

functionalities. Administrators' rough layout must be in place before implementing administrators' functionalities. Simple participant management must be in place before developing participants functionalities as most of them are user dependant. Furthermore, sample data must be inputted into the database so that it can be queried as different functionalities are being implemented. Administrators' account management must be in place and guaranteed that it works according to the requirements before implementing administrators' functionalities. If the administrators' account management is not in place administrators' functionalities will not be part of the final deliverables for this project. The documents will still contain the requirement and design for those administrator functionalities so that future developers could pick it up.

## Methodology

Since we have gathered all the needed requirements and have a general idea about how the web app will be designed a modified waterfall model will be implemented. This means that the developers will be building simple prototypes and showing it to the client to elicit the different requirements. The next step is that the requirements will be all compiled in a requirement document that will be shared with the client and the client must sign off on it indicating that this the product that they want built and that the requirements meet their needs. Once requirement is signed, the development process will start. During the development process, the developers will be testing their code using test cases specified in the verification and validation document. After all features are delivered by the deadline specified in the project plan, the developers will move into BETA testing where actual MESA participants will try out the web app and certain features will be modified as time allows. By the end of the second capstone term, the project will be delivered to the client at which points developers will have no responsibility of maintaining or supporting the code.

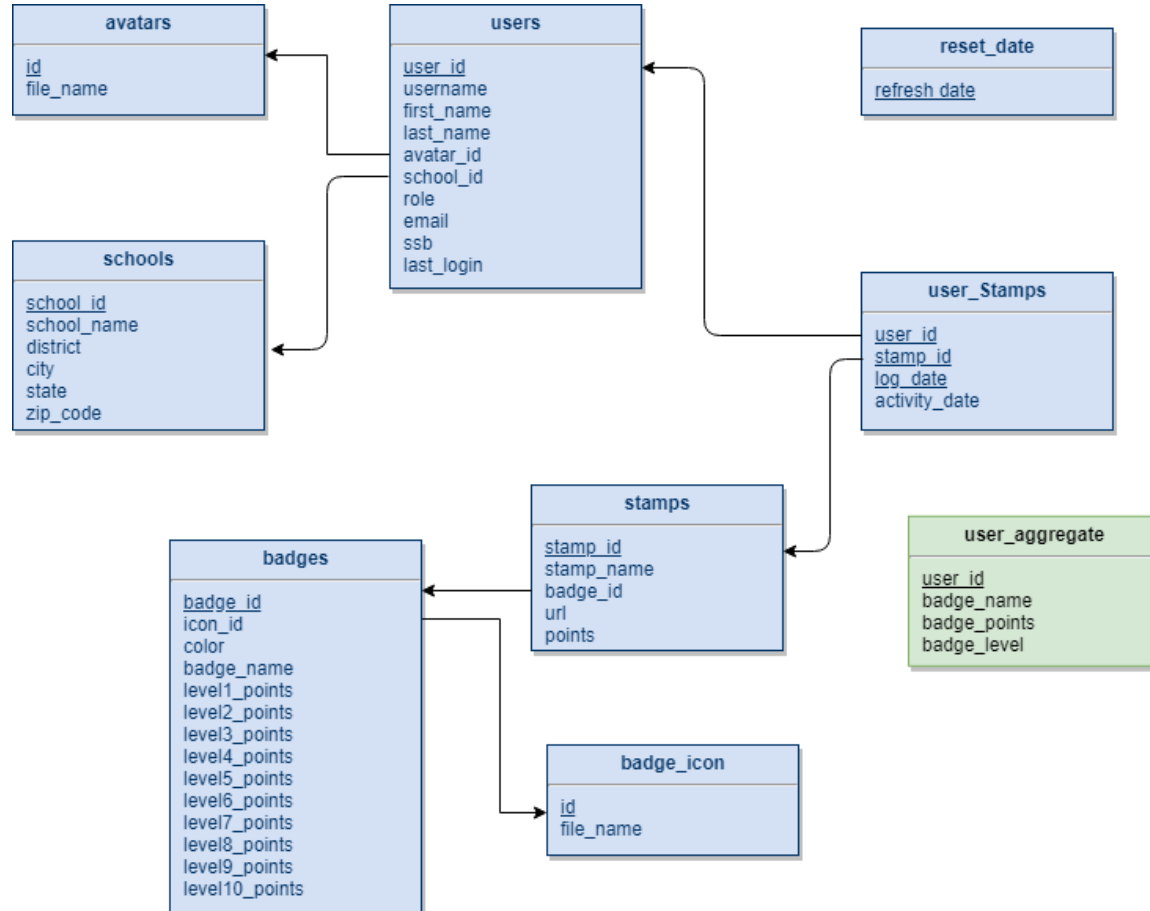
The web will be built using Python 3.7 as the programming language and specifically the Flask framework. For the frontend, the web app takes advantage of Bootstrap 4, HTML, and CSS where all pages are generated through Jinja2 Templates. The backend is managed through a MySQL database, which will keep track of all persistent data that the app will operate on and manage.

## Database Design

Our application will be using a MySQL database to store any relevant data relating to the games rules and student progress. The database will be hosted on the CAT's database servers throughout the design, development and deployment. The application utilizes the SQLAlchemy ORM, and all queries will be written via the built-in SQLAlchemy querying methods. Any changes made to the database structure will also need to be made to the corresponding SQLAlchemy classes in the code. This section will describe the general design of the database, as well as some specific design considerations we have made.

## Database Overview

The following diagram provides an overview of the various tables needed for our application. Each blue square represents a different table, and the green table represents a view. The name of each table is written in bold at the top of each square, with all the columns listed below it. Primary keys are underlined, and arrows represents foreign keys.



## Specific Database Considerations

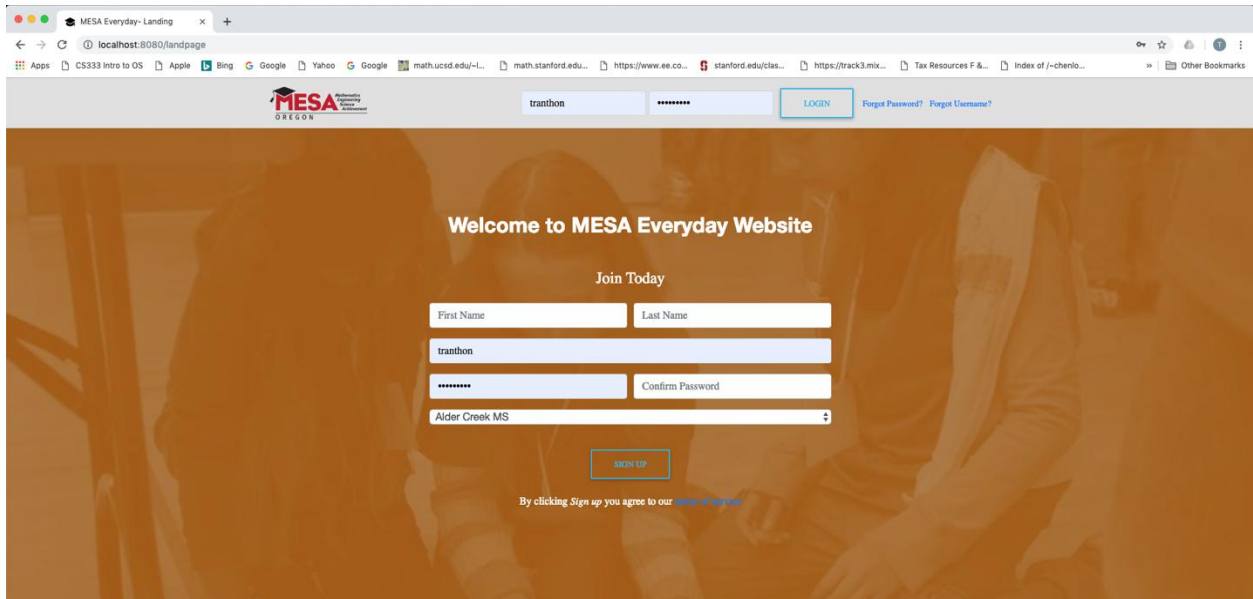
Each table (other than level\_resest) has an ID column. This is to make primary foreign key indexing take up minimal space. Each column that is a foreign key to another table, will have an index for it. Other indexes added are user.username, user.email, and school.school\_name. The user aggregate view provides calculations for the different users based on the stamps they have. The reason this is not a separate table is to avoid having duplicate information throughout the database that could potentially become inconsistent. Additionally, a trigger is placed on the “role” field in the “users” table to prevent anyone from changing their role to an admin.

## System Overview

The MESA Everyday web app that our team will be developing during the two terms of capstone will consist of the web pages listed below.

## Landing Page

Includes a header with logo, login, forgot username, password, a body with registration, and footer with MESA info

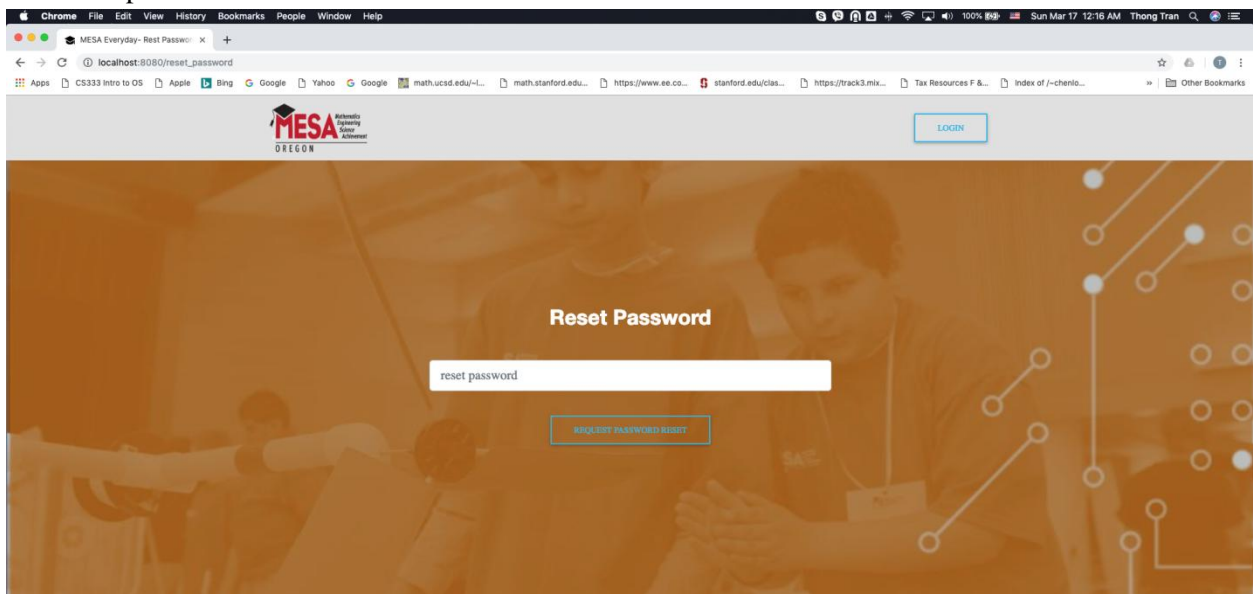


The screenshot shows a web browser window with the URL `localhost:8080/landpage`. The page features a header with the MESA OREGON logo, a login form with fields for username (containing 'tranthon') and password (masked with '\*\*\*\*\*'), and buttons for 'LOGIN', 'Forgot Password?', and 'Forgot Username?'. The main body has a large orange-tinted background image of students. It contains the text 'Welcome to MESA Everyday Website' and 'Join Today'. Below this is a registration form with fields for 'First Name', 'Last Name', 'Email' (containing 'tranthon'), 'Password' (masked with '\*\*\*\*\*'), 'Confirm Password', and a dropdown menu for 'School' (selected as 'Alder Creek MS'). A 'SIGN UP' button is at the bottom of the form. A small disclaimer at the bottom reads: 'By clicking Sign up you agree to our [terms of service](#)'.

## Resetting Password Pages

### Send Resting Password Link Page

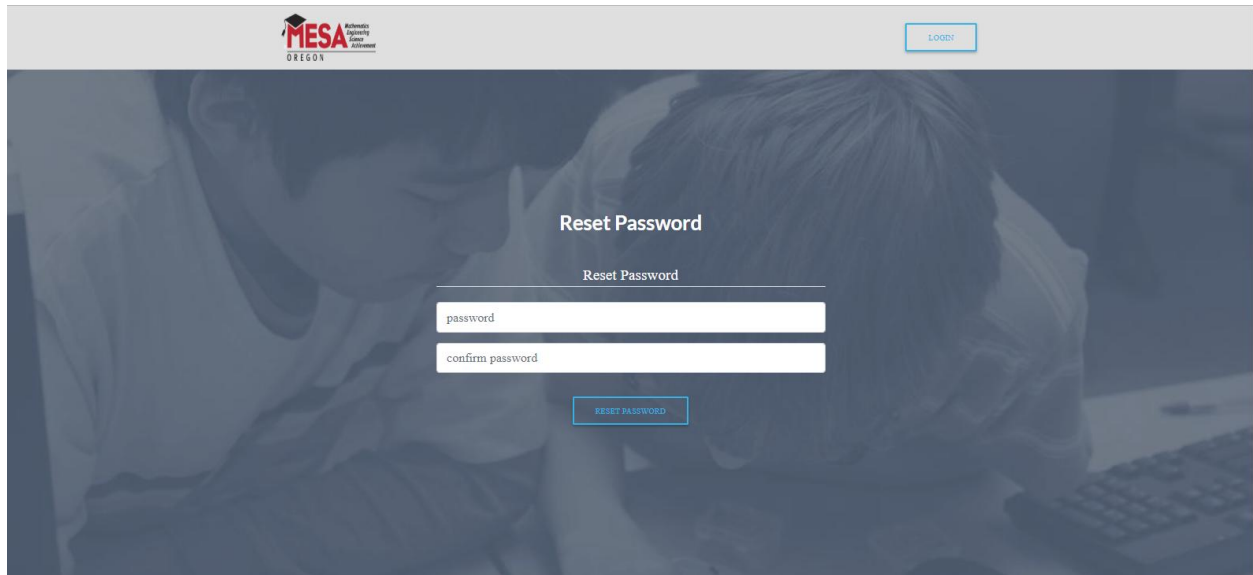
This page contains a field for a user to enter their email address and if validated they get emailed a link to reset their password.



The screenshot shows a web browser window with the URL `localhost:8080/reset_password`. The page features a header with the MESA OREGON logo and a 'LOGIN' button. The main body has a large orange-tinted background image of students. It contains the text 'Reset Password' and a form with a single text input field labeled 'reset password'. Below the input field is a button labeled 'REQUEST PASSWORD RESET'.

## Resetting the Password from Link Page

The reset password page contains two field one for the user to enter the new password and another for the user to confirm the password, and then there is a button to apply the changes.

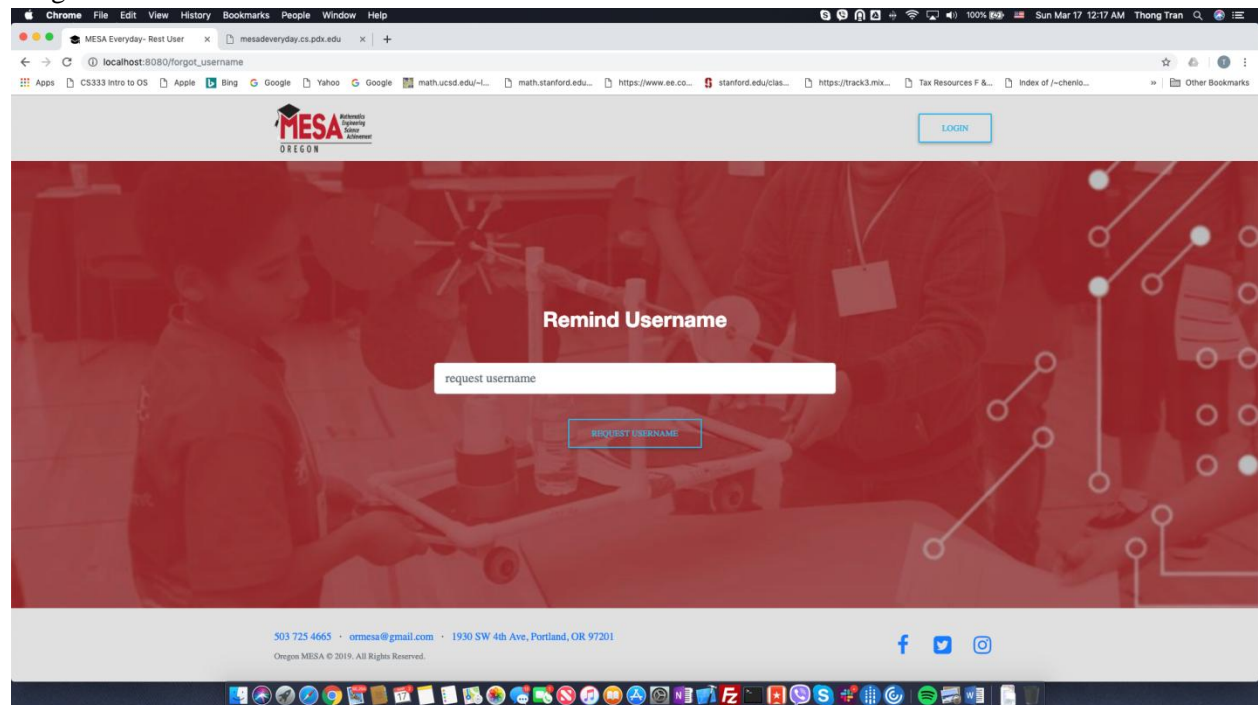


The screenshot shows the MESA Oregon website's 'Reset Password' page. The header includes the MESA Oregon logo and a 'LOGIN' button. The main content area has a blue-tinted background image of two students working. The title 'Reset Password' is centered. Below it, the text 'Reset Password' appears above two input fields: 'password' and 'confirm password'. A 'RESET PASSWORD' button is located below the fields.

## Forgot Username Page

### Send Email with Username Page

This page contains a field for a user to enter their email address and if validated they are emailed their assigned username.



The screenshot shows the MESA Oregon website's 'Remind Username' page. The header includes the MESA Oregon logo and a 'LOGIN' button. The main content area has a red-tinted background image of students working. The title 'Remind Username' is centered. Below it, the text 'request username' appears above a single input field. A 'REQUEST USERNAME' button is located below the field. The footer contains contact information: '503 725 4665 · ormesa@gmail.com · 1930 SW 4th Ave, Portland, OR 97201', social media icons for Facebook, Twitter, and Instagram, and a copyright notice: 'Oregon MESA © 2019. All Rights Reserved.'

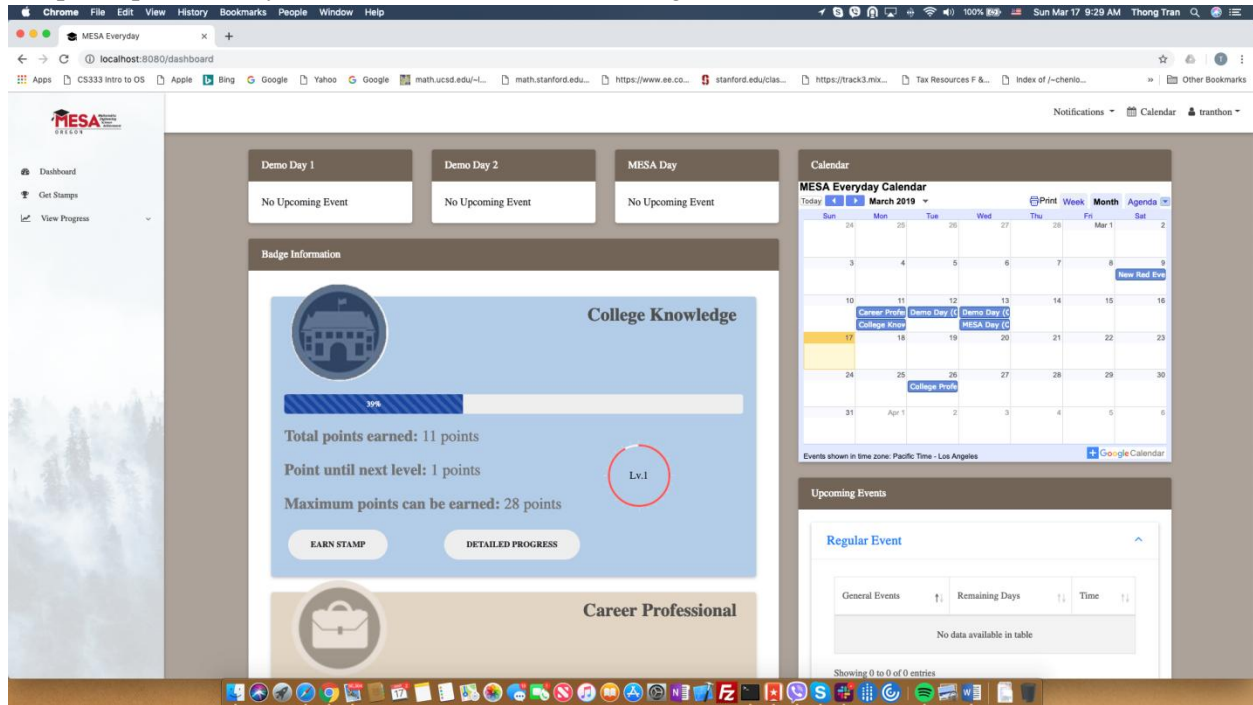


## Participant's Side Overview

The participant side overview contains all of the sections listed below

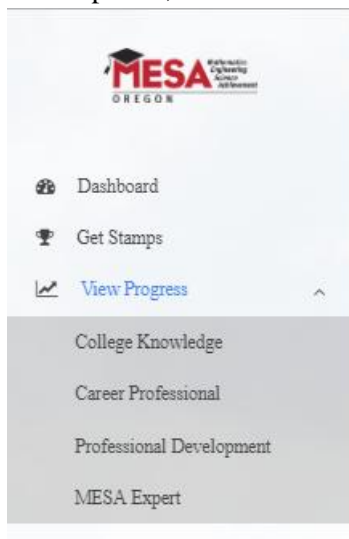
### Participant Side Layout

The participant side layout will look as shown in the image below



### Side Navigation Bar Section

The side navigation bar contains link to access all the different pages accessible to an authenticated user. These include a link to Dashboard, The page where a user can earn stamps "Get Stamps", and The View Progress dropdown that shows detailed analysis of where a user is for one of four different badges. The badges at the time of building the app, were College Knowledge, Career Professional, Professional Development, and MESA Expert

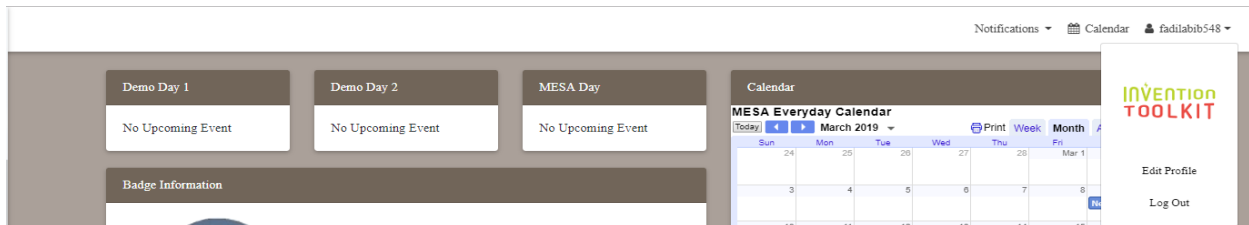


## Top Navigation Bar Section

Includes notifications, calendar, and a user name dropdown where user can find the Edit Profile and logout button

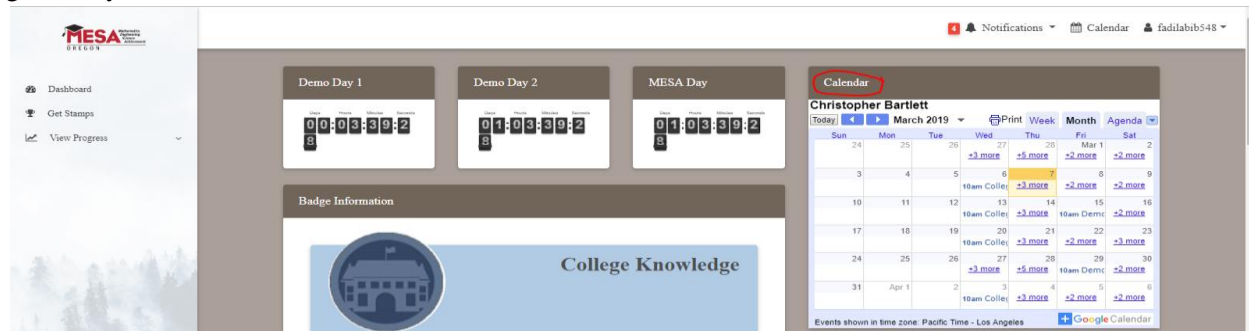


Below is a screenshot showing how one can access the Edit Profile and Logout Button as well as see the avatar associated with their account



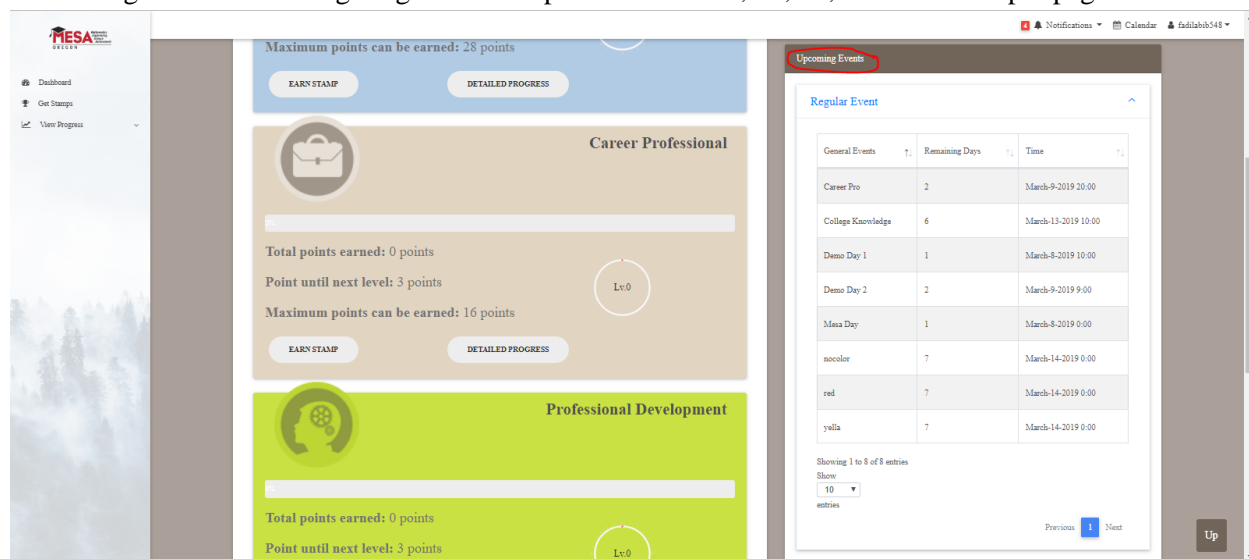
## Right Side Calendar Section

On the Top Right, a small calendar widget displays the current month and all events that will occur on a given day.



## Right Side Events Section

Under the page's calendar, there will be a printed list of all events in the next (7 days); it will display the title, remaining days, and time of the event. The data in the table is can be sorted in ascending and descending order as well as giving users the option to show 10, 25, 50, or 100 events per page.

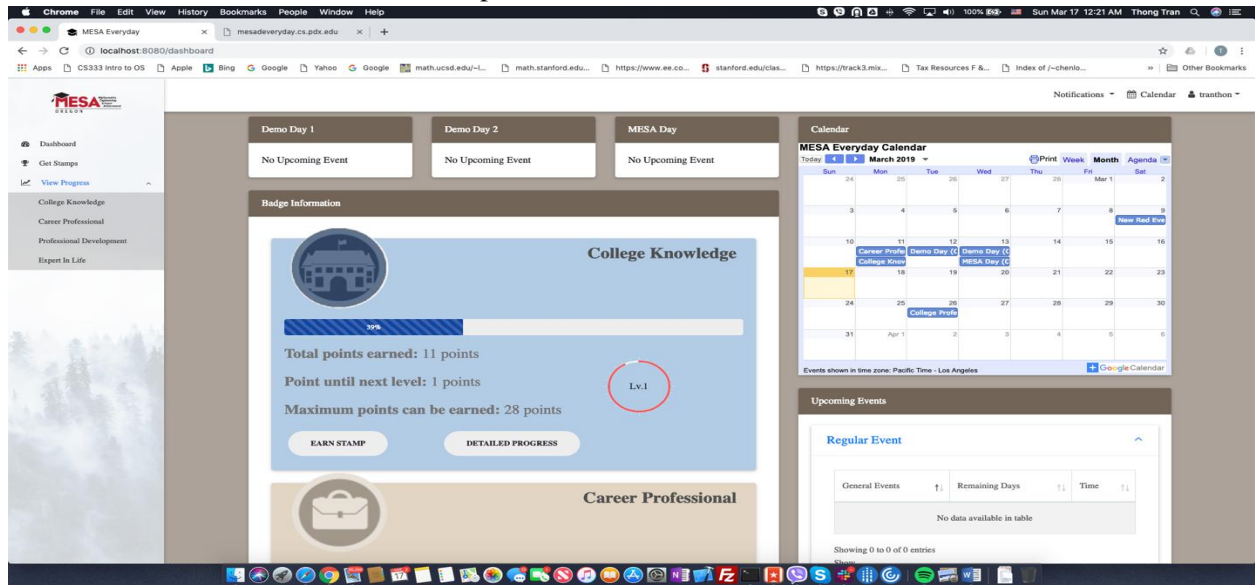


## Main Section

Below is summary of all the different pages contained in the user-side of the web app.

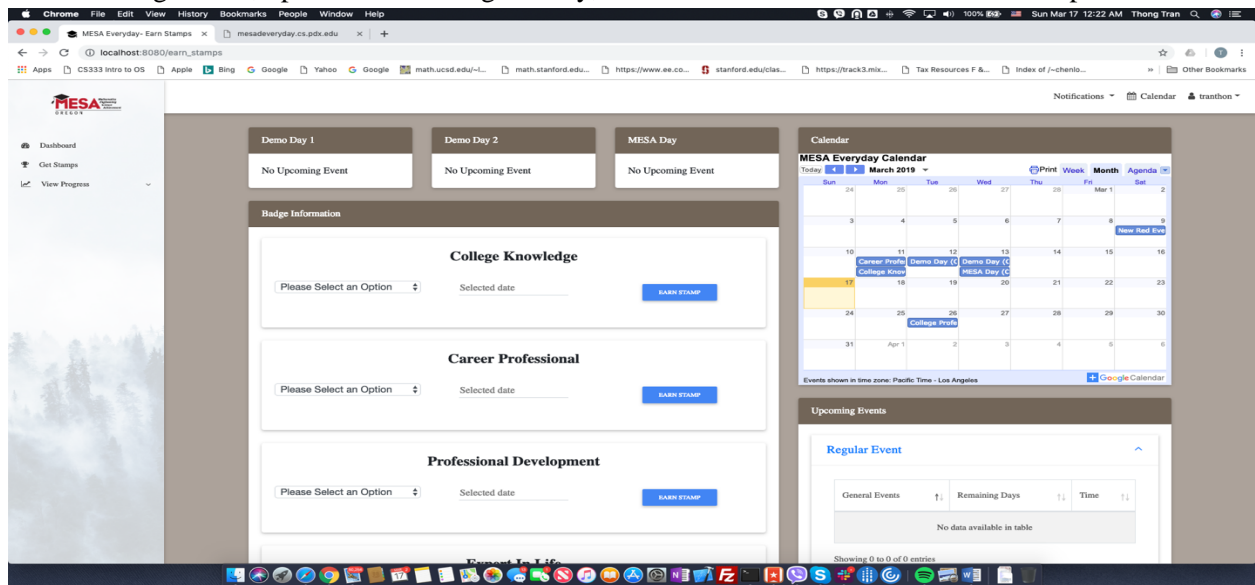
### Dashboard Page

Badge info includes badge name, badge icon, progress bar, total point earned, point until next level, maximum points can be earned, progress bar circle, earn stamp button, and detail progress button. There are also countdown sections for two demo days and MESA day. There is a calendar and upcoming event section on right side and a side navigation bar on the left side. There is also a top navigation with calendar, notification and user info on top.



### Adding Stamps Page

Includes badge name, option for choosing activity, select date from calendar and earn stamp button



## Single Badge View Page

Includes badge info: badge name, level, badge icon, total point earned, point until next level, icon that that user to event tied to a given badge.

The screenshot shows the 'Single Badge View Page' for the 'MESA Expert' badge. The page layout includes a sidebar with navigation links (Dashboard, Get Stamps, View Progress), a top navigation bar with Notifications, Calendar, and a user profile (fadilabb548). The main content area is divided into three sections: 'Demo Day 1', 'Demo Day 2', and 'MESA Day', each showing 'No Upcoming Event'. Below these is the 'Badge Information' section, which displays the badge icon (a flame), the badge name 'Badge MESA Expert', the level 'Level 5', the total points earned '27 points', and the points needed to reach the next level '1 points'. To the right of the badge information is a 'Calendar' section showing the 'MESA Everyday Calendar' for March 2019. The calendar highlights several events: 'Demo Day' on March 11, 12, and 13; 'College Pro' on March 25; and 'New Red E' on March 8. Below the calendar is a section for 'Upcoming Events'.

It also includes a list of all the stamps that a user has earned for a given badge including details of when they earned it (as reported by students), when they actually logged it in to the system, and a button to allow them to delete a given stamp.

The screenshot shows the 'Earned Stamps For Badge MESA Expert' section. It features a search bar and a 'Show' dropdown set to '10 entries'. Below this is a table listing the earned stamps. The table has columns for 'Earned Stamps', 'earned date', 'logged date', and a 'DELETE' button for each entry. The entries are as follows:

Earned Stamps	earned date	logged date	DELETE
Acknowledge MESA Developers are the Best	2019-03-22	2019-03-13 15:46:25	DELETE
Attend a Family Night	2019-03-13	2019-03-13 15:39:10	DELETE
Compete at MESA Day	2019-03-30	2019-03-19 12:01:31	DELETE
Invention Ambassadors	2019-03-26	2019-03-19 12:02:55	DELETE
Showcase at Demo Day	2019-03-20	2019-03-19 12:00:47	DELETE

Below the table, it shows 'Showing 1 to 1 of 1 entries' and a 'Show' dropdown set to '10 entries'. To the right of the table is a 'Regular Event' section showing details for 'College Professional (Green)' with 7 remaining days and a time of March 26, 2019, 0:00. It also includes a 'Showing 1 to 1 of 1 entries' and a 'Show' dropdown set to '10 entries'.

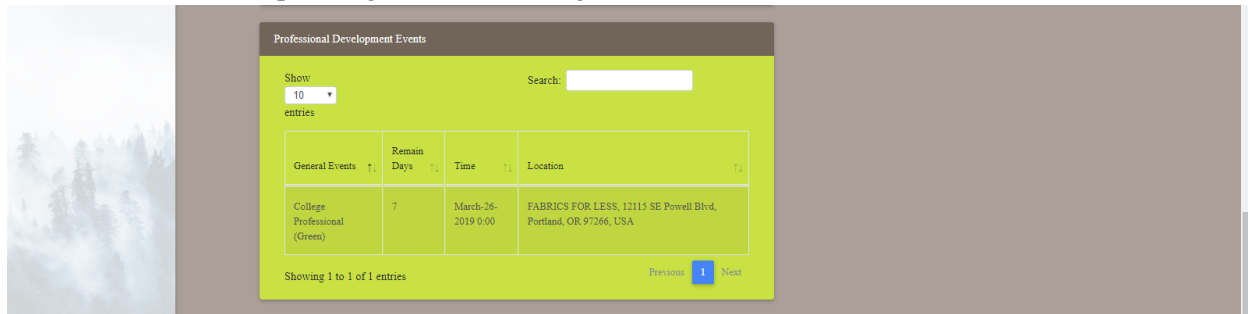
It also includes a list of all stamps that are unearned yet

The screenshot shows the 'Unearned Stamps For Badge MESA Expert' section. It features a search bar and a 'Show' dropdown set to '10 entries'. Below this is a table listing the unearned stamps. The table has columns for 'Unearned Stamps' and a 'DELETE' button for each entry. The entries are as follows:

Unearned Stamps	DELETE
Compete on a National Track team	DELETE

Below the table, it shows 'Showing 1 to 1 of 1 entries' and a 'Show' dropdown set to '10 entries'.

It also shows a list of upcoming events for a badge



This list can be accessed for the summary of progress for a given badge visible at the top, by clicking at the little calendar icon with count of number of upcoming events



### Notification Bell

At the top of the web app there is a toolbar and the first icon to the right side of the app is a bell shaped icon called notification, this will show urgent upcoming events (next 2 days) and it will display a count next to the icon for the number of events coming up. There is also a drop down arrow that when clicked will show what those events are.

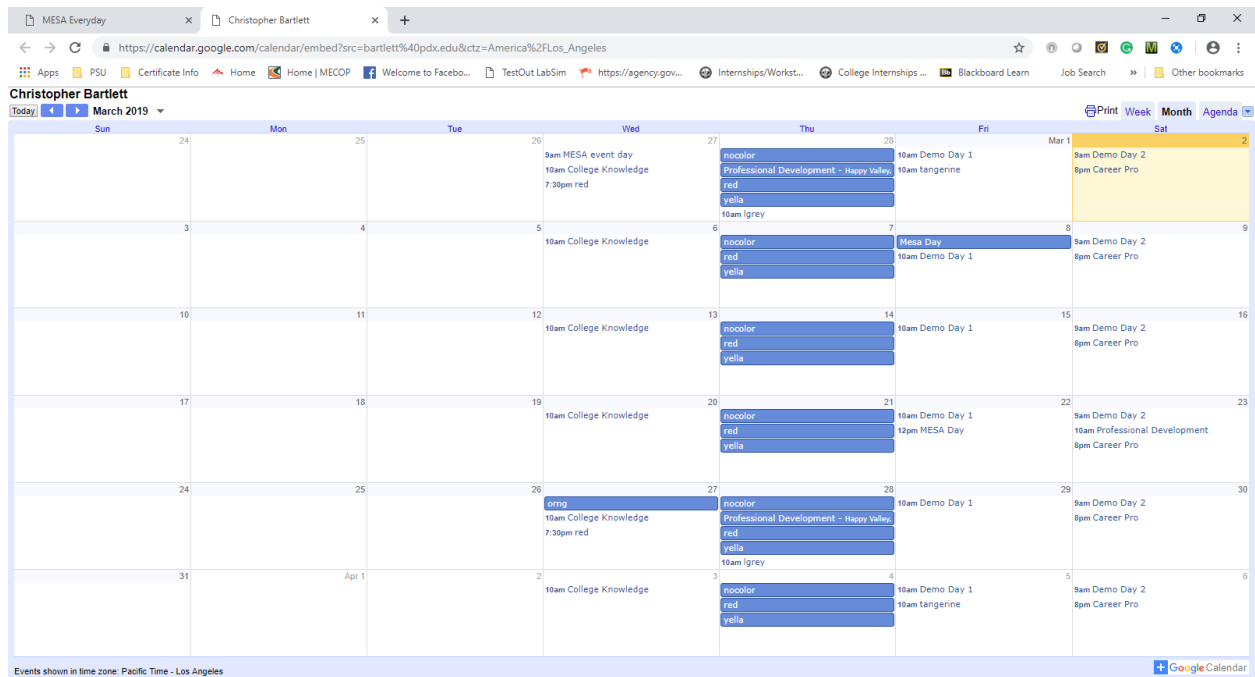


### Calendar Page

The Calendar Page will be accessed using the Toolbar at the top of the screen, by clicking the Calendar Button; this will take the user to the shared Google Calendar that is linked to the App. The Calendar will be accessed as a separate browser tab

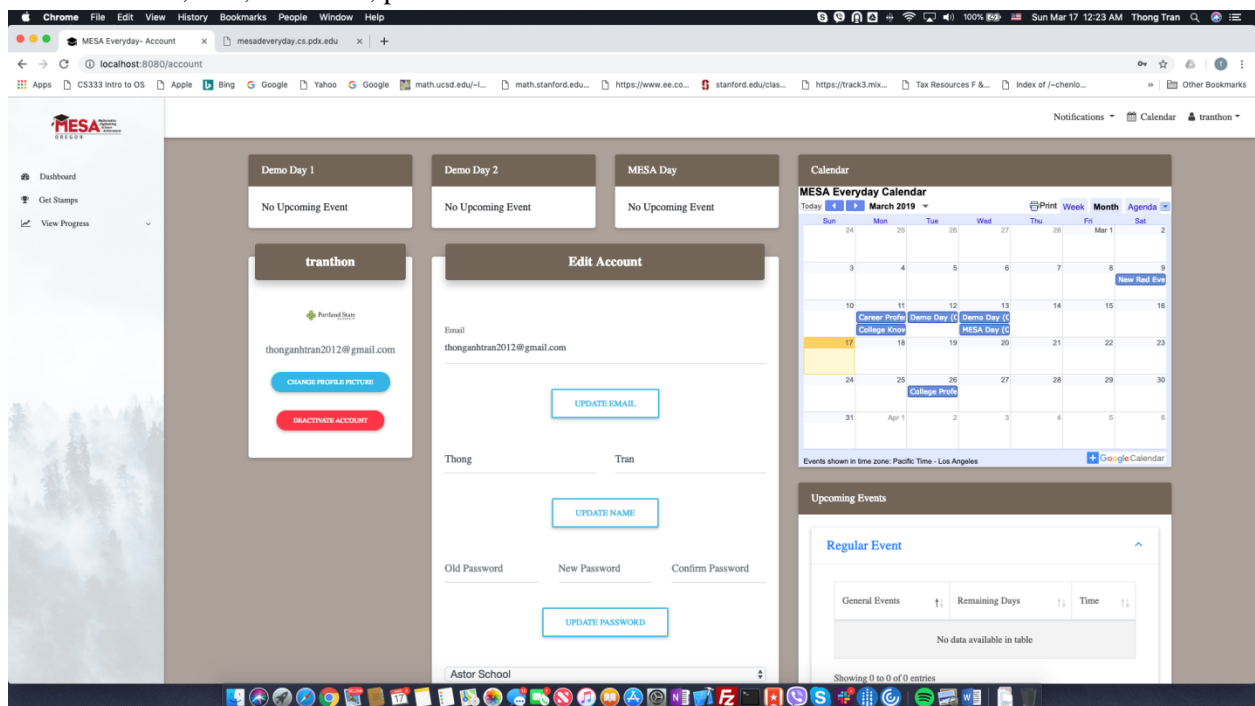


Here is how the calendar itself will look like when the calendar icon is clicked



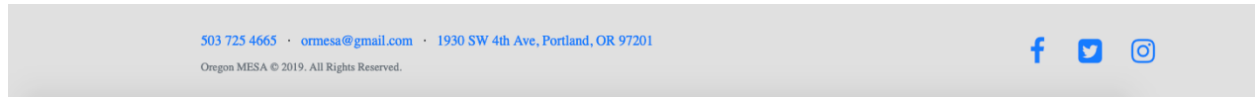
## Editing Profile Page

Includes username, user profile pic, change profile picture button, deactivate account button. User can edit their email, first, last name, password and school.



## Footer

Includes phone number, email, address, social media pages.

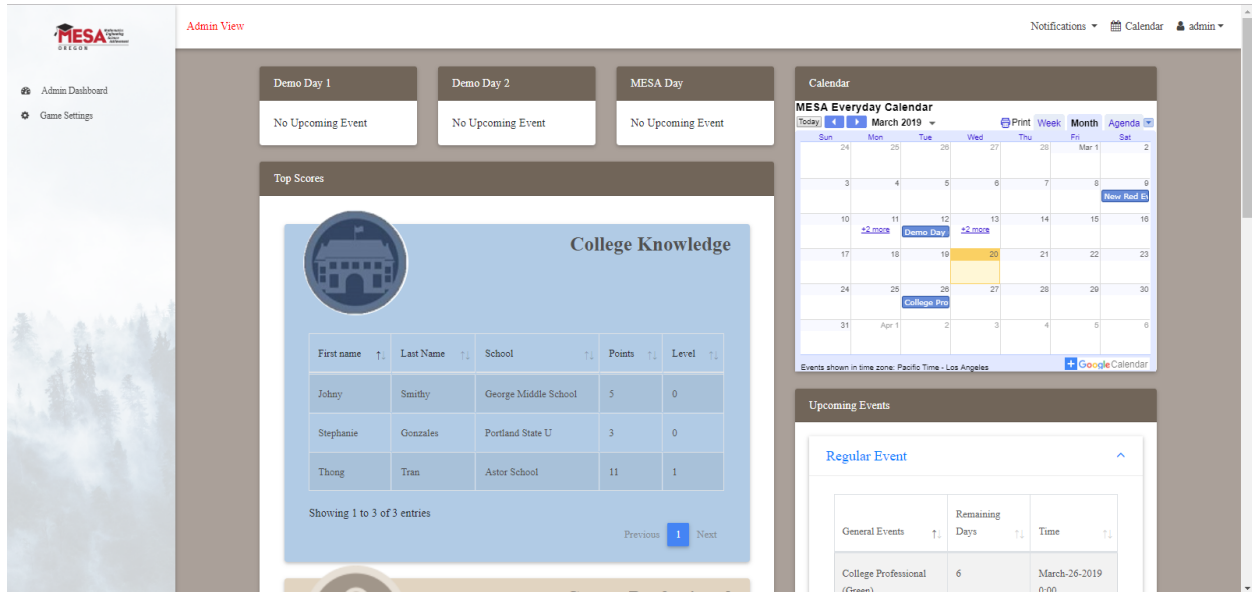


## Administrator Side Overview

Below is the detailed description of the administrator side for the web app.

### Administrator Side Layout

The administrator layout has a top navigation bar, side navigation bar, countdown for Demo Days and MESA days, calendar, upcoming events, and a content section that is modified based on the page.



### Side Navigation Bar Section

The side Navigation bar includes both the dashboard page and the game setting page.



## Top Navigation Bar Section

The top Navigation bar is the same as the user side, except for the username will be the username for the admin and the red text to the left side of the navigation bar that says Admin View

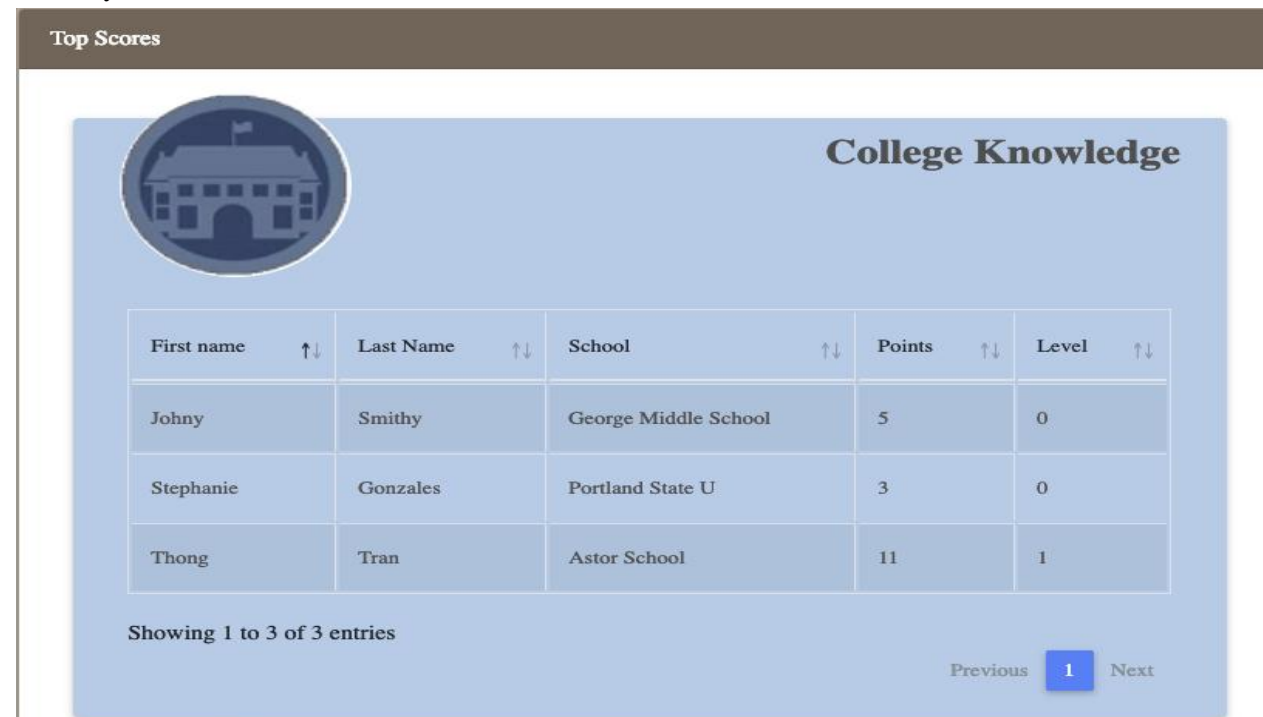


## Main Section

The main section could have one of three variations: the dashboard page containing top three scores for each badge, the Game Setting page where an admin can modify the rules of the game, and the Admin Setting page that allows for change of account settings.

### Dashboard Page

It will include the badge icon, the name of the badge and the First and Last names of the top three scoring students for each badge as well as their school, the total points they have earned and the level they are currently one.



The screenshot displays a section titled 'Top Scores' with a dark brown header. Below it is a light blue box containing a circular badge icon of a school building and the title 'College Knowledge'. A table lists the top three scoring students:

First name	Last Name	School	Points	Level
Johny	Smithy	George Middle School	5	0
Stephanie	Gonzales	Portland State U	3	0
Thong	Tran	Astor School	11	1

Below the table, it says 'Showing 1 to 3 of 3 entries'. At the bottom right, there are 'Previous', '1' (highlighted), and 'Next' buttons.

### Game Settings

Includes badge name, badge icon, 10 level boxes for admin to edit the point for each level and it also includes a section for adding additional stamps to any of four different badges, Removing stamps for any of four different badges, changing badge name for any of the four badges, and changing the badge icon for any of the four badges.



**Modify Badge Level:** Badge Icon, Badge Name, 10 levels with empty boxes for scores and button to apply modification to the badge levels.

**Add Stamps:** Drop down for the name of the stamp to pick, the stamp name, and the points that a user will earn if they earned the stamp.

**Remove Stamps:** Drop down for the name of the stamp to pick, and a dropdown for the stamp name to remove.

**Change Badge Icon:** Includes a button that when clicked will show a pop-up with a dropdown to select badge name, an icon to be assigned to the given badge, and buttons to update the icon or cancel the change.

## Admin Settings

Includes username, profile pic, tab school includes: school name, delete school, tab account includes: account info, update password, remove old account, tab reset day: set reset day

### School Tab:

The screenshot shows the 'SCHOOL' tab selected in the Admin Settings. The left sidebar contains the 'admin' profile and the 'SCHOOL' tab is highlighted in the top navigation bar. The main content area has a 'School Name' section with a text input field and an 'ADD SCHOOL' button. Below this is a 'Select School to Delete' section with a dropdown menu showing 'Alder Creek MS' and a 'DELETE SCHOOL' button. On the right, there is a Google Calendar widget showing events for March and April, and a section for 'Upcoming Events' with a 'Regular Event' link.

### Account Tab:

The screenshot shows the 'ACCOUNT' tab selected in the Admin Settings. The left sidebar contains the 'admin' profile and the 'ACCOUNT' tab is highlighted in the top navigation bar. The main content area has an 'Account Info' section with an email input field and an 'UPDATE EMAIL' button. Below this is an 'Update Password' section with fields for 'Old Password', 'New Password', and 'Confirm New Password', and an 'UPDATE PASSWORD' button. Further down is a 'Remove Old Accounts' section with a warning message and a 'Delete Old Accounts' button. On the right, there is a Google Calendar widget showing events for March and April, and a section for 'Upcoming Events' with a 'Regular Event' link.

### Reset Day:

The screenshot shows the 'RESET DAY' tab selected in the Admin Settings. The left sidebar contains the 'admin' profile and the 'RESET DAY' tab is highlighted in the top navigation bar. The main content area has a 'Set Reset Day' section with a text input field and a 'CHANGE RESET DATE' button. On the right, there is a Google Calendar widget showing events for March and April, and a section for 'Upcoming Events' with a 'Regular Event' link.

## System Design

This section provides the general consideration and restriction for the different functionalities that will be implemented by the developers for the web app. It also includes details of how requirement will actually be implemented.

### Participants Design

This section of the design discusses all the functionality that a user, of the web app, with a participant role, should have when interacting with the app. This includes everything from session management to tasks they can perform with the app.

#### Participants' Session Management Design

This section discusses the details of how participant functionalities related to how the web app manages a participant session. This includes registering an account, signing in, signing out, dealing with forgetting username or password, deactivating account, and editing basic profile features.

#### Registration

In order for a user to register with the web app, they will be filling a form as detailed in the Landing Page section of the design document. The email field will be checked for a valid email using a built in function provided by flask WTF library. The password will be checked to be at least eight characters long, using a built in function provided by flask WTF library, and be checked against a regular expression using a built-in function that checks inputs against a provided regular expression. The regular expression enforces the password policy to be any characters but at least one of them must be a number and another to be a special character. The two password fields are also checked that they match using a built in function that is provided in the second password field and it checks if the input of the second password field matches the input of the first password field.

Once a user hits the registration button, the first eight letters of their provided first name (all of first name if less than or equal to eight characters) and the first eight letters of their provided last name (all of last name if less than or equal to eight characters) will be concatenated together. Then a random number between 000 and 999 will be generated by a separate function and will be concatenated to the combination of first and last name to generate a unique username for each registered user. In the case that collision happens as two people match on their first and last name, the three randomly, generated digit would insure the uniqueness of the name. In the case that the function generates the same three-digit number for two colliding names, the randomly generated digit will be incremented by one until a unique string is generated. In the extreme case that 1000 people match in their first and last names, the function will fail to generate a username for that user, and suggests to them that they change the spelling of their name. Given the number of users of this website and length of the generated username, the chance of colliding name is minimal, but it should still be accounted for so that the program fails safely.

Once the user hits the registration button, they will also be assigned a pre-set hard-coded default avatar picture that they can later change in the editing profile section of the web app. They will also see a green banner message telling them that their account has been created successfully and tell them their generated username. Furthermore, an email will be sent to the email provided by the user notifying them of their generated username. This is so they can keep it for their record since the username is required to login to the web app.

### Signing In

In order for a user to sign-in to the web app, they will be filling a form as detailed in the Landing Page section of the design document. Once they click the Login button, the username, password combination will be verified by checking the database of users, and if the user exists, they will be signed in successfully. In the case that they are not verifiable, a red message will be displayed to them indicating that they do not have an account with the app and they will be advised to register an account. The web app will not provide any information to the user about which field is wrong so that it does not compromise the security of the web app.

Upon successful sign-in, the time stamps of the login for the user will be recorded and saved in the database, as this will be used to check for inactive accounts that need to be administratively removed.

### Signing Out

Once a user logs-in, they will be able to log-out by clicking on the profile icon at the top right corner of the top navigation bar and picking the sign-out button.

### Resetting Password

In order to reset password a user must click a link in the landing page called reset password. When that link is clicked, the user will be prompted with a simple form (check the Resetting Password Page, Sending Resting Password Link Page section under System Overview). When the user enters a valid email, an email will be sent to them with a unique link using a generated token that lasts for 30 minutes. When the link is clicked within the 30 minutes, it will redirect the user to page where they will be able to reset their password by filling out a form (check the Resetting Password Page, Resetting the Password from Link Page section under System Overview). The password will be checked that it matches the minimum requirement and that the fields are valid, just like in registration, before changing the user's password. After 30 minutes, if the link is not clicked, the token will expire and the user will have to generate a new unique token by clicking on the landing page reset password link. The user will be directed to the landing page if the token expired or for other reason, the token becomes invalid.

### Forgetting Username

If the user forget their immutable username, they can click on a link in the landing page called forgot username, which will prompt them with the same simple form (check the Forgot Username Page, Sending Resting Password Link Page section under System Overview). When the user enters a valid email, an email will be sent to them with their username.

### Deactivating Account

The user will re-enter their first and last name in a form. The form will validate that the first and last name entered match the current user. If there is no name entered than the deactivate account button will not be active and will not work. If the information matches they will be met with a pop-up screen that will say are you sure you want to deactivate your account. With a YES and a NO buttons. Yes will be in red, and No will be in green. If the user hits YES, the query that will clean all their data will be activated and their account will be erased. They should be logged out and redirected to the landing page immediately after deactivation is processed. If they enter their credentials again to login it will tell them that their credentials are invalid.

## Editing Profile

### *Change First and/or Last Name*

When a user changes their first and last name from the Editing Profile form (Check Participant's Side Overview, Main Section, and Editing Profile Page for details of form design), an update query will be executed and it will update the first and/or last name for that user based on their inputs in the form. The user must fill both first and last name even if they want to change only one of them (they can just re-enter the part of the name they do not want to change).

### *Change School*

When a user changes their school from the Editing Profile form, (Check Participant's Side Overview, Main Section, and Editing Profile Page for details of form design), an update query will be executed and it will update the school id for that user based on the option they pick from the drop-down list.

### *Change Avatar Picture*

When a user changes their avatar picture from the Editing Profile form, (Check Participant's Side Overview, Main Section, and Editing Profile Page for details of form design), an update query will be executed and it will update the avatar id for that user based on the picture they pick from the drop-down list.

### *Change Email Address*

When a user changes their email address from the Editing Profile form, (Check Participant's Side Overview, Main Section, and Editing Profile Page for details of form design), an update query will be executed and it will update the email address for that user based on the new email that they enter. The email will be checked to be valid just like in Registration.

### *Change Password*

When a user changes their password from the Editing Profile form, (Check Participant's Side Overview, Main Section, and Editing Profile Page for details of form design), the password will be updated in the same way that the password is changed when a user resets their password. The difference is that the user will not be sent a unique link but rather they will be able to update it on the spot, as they must be successfully authenticated in order to perform this change. They will instead be required to re-enter their old password before changing to a new one.

## **Participants Tasks Design**

Below is the design for functionalities that a participant gets upon their successful authentication to the web app.

### **Overall Visibility**

From the dashboard page, a user can see a high-level summary of their progress in each of 4 different badges. The summary includes the total cumulative points a user has earned, the points they need to get to the next level and the maximum number of points they can earn for a badge. A circular progress bar shows the student current level and fills up completely once a level is earned and then it jumps to the next level and is cleared again. A progress bar shows the amount of points user has earned from total possible points is also available to give user a brief summary of their progress. From the overall visibility card for

each badge a user can get to the earn stamp page to earn more stamps or click on a different button called detailed progress and be able to view details of how they earned the current points and levels they have.

### Adding Stamps

When the user picks "Get Stamps" from the left side navigation bar they will be transferred to a page where they will be able to add stamps for each of the different badges available with the web app. For specific details of what is included in the form check Participant's Side Overview, Main Section, Adding Stamps Page. When a user picks a certain section based on the badge title they will be prompted to pick a stamp from a drop down menu of stamps and also to pick the data that they earned the stamp from a data picker field, finally they will hit an "Earn Stamp" button. Upon hitting the button an insert query will be executed that will add the information inputted by the user into the user stamps table. The query will insert the badge name based on the section for which the "Earn Stamp" button was hit, the stamp name based on the drop down menu choice of the user, and the reported date based on the date picker field. In addition, a function will be called to determine the current system time and that information is added to the table as the actual date where the stamp was reported as earned. The time the stamp earned was reported cannot be controlled by the user, it is simply the time the submit button has been hit.

### Viewing Currently Obtained Stamps

For each badge, a user can access details of every stamp they earned for the badge, when they earned it and when they reported it as earned. The fields of the table can be sorted in ascending or descending order. Every time a stamp is added in the add stamp page it will show up in the currently obtained stamps for the badge to which the given stamp was added. The system will record the time stamps the user hit submit to add the stamps to their earned stamps table and that time stamp is displayed as logged date. The earned data is whatever the user picked as the date when they were filling out the adding stamps form. The database stores all that information about each stamp and for each badge for each of the users and a query displays the correct information of all the badges for the user who is currently logged in to the system.

### Viewing Needed Stamps

In order to determine needed stamps for each badge, the system will query the database for a certain user and badge to see what stamps that user has earned. The returned list of stamps earned will be compared to a query that returns all possible stamps for a badge. Each stamp that exists in the list of possible stamps for a badge and does not exist in the list of earned stamps for a user and a badge is a needed stamp. The list of such stamps will be displayed to the user so that they know the stamps they need to try to earn. If they have earned all the stamps for a badge, a message will be printed to them in the section where the missing stamps list is, that will say, "You have earned all the stamps for that badge. Great Job!".

### Viewing Current Level

Current level is calculated using a database view and is updated in real time as users add and remove stamps. The current level is determined based on the points needed for a certain level and the current points a user has earned.

### Viewing Currently Accumulated Points

In order to get currently accumulated points for a badge the database will be queried for all the stamps that have been earned for a badge for the current user and the points for each one will be tallied up. The

tallied up results is then displayed to the user in the web app according to the location specified in the single badge view page.

#### Viewing Needed Points for Next Level

The needed point for next level is a simple calculation of how many points are left until a user gets to the next level. By subtracting the cumulative points needed for the next level and subtracting the total points a user has earned thus far from it, the needed points for the next level are determined and displayed to the user in the detailed badge page as well as in the overall visibility summary from the Dashboard.

#### Removing a Stamp

As a user earns stamps, their Earned stamps for the badge will have a delete button next to it, when that button is clicked the user would be able to remove the stamp and their points will be updated accordingly as well as their level. When a user clicks on the delete button, they will see a page that will ask them if they are sure that they want to delete the stamp, if they hit cancel the stamps will stay there. However, if they hit delete, the stamp will actually be deleted and cannot be recovered back or undone.

#### Participants Calendar Design

Below is the design for calendar functionalities that a participant gets upon their successful authentication to the web app.

##### Notifications Bell

There is a notification bell at the top of the web app which is the first icon visible at the right side of the web app; this bell icon shows count of number of upcoming events for the next 2 days.

##### Viewing Current Month's Calendar

On the Top Right, a small calendar widget displays the current month and all events that will occur on a given day. When the user clicks on a day in the calendar, where there are some events, the calendar will expand to show the details for those events.

##### Viewing General Upcoming Task for Current Week

Under the page's calendar, there will be a printed list of all events in the next week, it will display the title, time, location, and badge it represents.

##### Viewing Countdown for Three Important MESA Dates

The three events will show in the top of the website and they will be viewable in all pages. MESA Day and the Two Demo Days will have a current countdown accurate to the minute until it starts. The event names will be MESA Day, Demo Day 1, and Demo Day 2.

##### Viewing Upcoming Tasks Based on Badge

Users can navigate to the different badges, when they do so, they will see different events that occur in the next couple of weeks and they will be visible under all of their stamps and progress.

##### Calendar Security

Security will be handled via Google, the calendar will be readable by all and editable only by the owner, the calendar widget will use these permissions.

## Administrator Design

All administrators related designs listed below are not part of the MVP. This section of the design discusses all the functionality that a user, of the web app, with an administrator role, should have when interacting with the app. This includes everything from session management to tasks they can perform with the app.

### Administrator Session Management Design

This section discusses the details of how administrator functionalities related to how the web app manages an administrator session. This includes registering an account as admin, signing in, signing out and dealing with forgetting username or password.

#### Registration

The admin will not be able to register an account with the app, their account is created for them as part of the app development process and it is shipped with the built in admin account.

#### Signing In

In order to sign-in an admin will follow the same steps a user follows to sign-in.

#### Signing Out

In order to sign-out, an admin will follow the same steps a user follows to sign-in.

#### Resetting Password

Admin can reset their own password in the same fashion that user can reset their own password and it must meet the same complexity and length requirement just like any other user. Admin will simply hit the Forgot Password link from the landing page and provide the email address they used when admin account was created or the one that has been last modified whichever comes later. Then they will get an email sent to them with a complicated link that when they click on it within 30 minutes they will be able to reset their password just as any regular user does.

#### Forgetting Username

Admin can have their username sent to them in case they forget it, but they must provide the email they used to register the admin account. The process for getting their username back is the same process that any regular user would follow to get their username back.

#### Editing Profile

##### *Change Email Address*

In order for the admin to change their email address, they simply put in a new email address and click on the update email button, which will call a query that will update the email field for admin to be the new email that the user has chosen.

##### *Change Password*

In order for admin to change password they must provide their old password and then enter a new password that meets the length and complexity requirements and confirm that password by entering it again. The validation process is the same as the validation for changing password for any regular user. The Old password will be checked that it is actually the old password for admin account. Then once the



verification happen and succeeds, the password will be updated via a query and then from the next login attempt the new password must be used to login in as admin.

### Administrator Tasks Design

Below is the design for functionalities that an administrator gets upon their successful authentication to the web app.

#### Adding Schools

The administrator to add a school name and then they will click a button that will add the school to the school

#### Removing Schools

The administrators will be able to pick any of the schools and hit a delete button and it gets deleted.

#### Editing Badge Names

Any of the four stamps available in the system can be changed. To change a stamp, you pick it. A name from a drop down (that name will be based on whatever it is in the database).A blank input field next to the dropdown will allow the user to enter the new name for the badge.

#### Badge Picture Picking

Any of the four stamps available in the system can be changed..To change a stamp, you pick i. A name from a drop down (that name will be based on whatever it is in the database).A button will be clicked that will show a pop-up of all possible badge pictures.

#### Adding Extra Stamps

The administrator will pick a badge name from a drop-down of badge names. There will be another empty field next to it for them to add the stamp name they want .The stamp name will be added to the badge that was picked from the drop down, a stamp has been added.

#### Removing Stamps

The administrator will pick a badge name from a drop-down of badge names .There will be another drop-down field that will be populated with all the stamps in the database that belong to the badge name the user picked.

#### Altering Points Required for Badge Levels

For each badge, there are 10 levels. Admin can set points for each level by entering the necessary point for each level of each badge and click on the Change Badge Point button to set the points.

#### Altering Badges Max Level

For each badge, an admin can set how many points are required for each level in the the Admin Settings form, (Check Administrator Side Overview, and Settings Page for details of form design). An admin can alter multiple levels at once, but only one badge at a time. However, there are rules that must be followed:

1. To set the max level, leave the points required blank for each level after the desired maximum.
2. Level 1 cannot be blank for any badge.
3. Each level must require more points than the previous level.
4. The points required must a positive whole number.

### Viewing top three scores for each badge

Upon entering the admin side of the application, they will be shown the three highest scores earned on each badge, and who has earned them. This is done in 3 steps:

1. Find out what the top three scores are in each badge (do not consider 0 a score).
2. Find all the users that have those top three scores.
3. Display all those users and their scores for each badge.

### Removing Old Accounts

When an admin selects a year and presses the remove accounts button in the Admin Control form, (Check Administrator Side Overview, and Control Page for details of form design), an update query will be executed that will look through the last login time of each user, and delete the users that have not been active for longer than the years selected. This will also delete all the stamps the user has earned.

### Setting Academic Year Date

When an admin selects a date (month/day) and presses the update academic year button, (Check Administrator Side Overview, and Control Page for details of form design), an update query will be executed that will set the reset date for the game. Only stamps that have been earned during the most recent occurrence of this date will be considered in the game.