Table of Contents

Introduction

MESA Everday 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.

Terms and Definitions

Term 1	Definition 1
Term 2	Definition 2

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. GET LIST OF COLORS FROM CLIENT.

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.

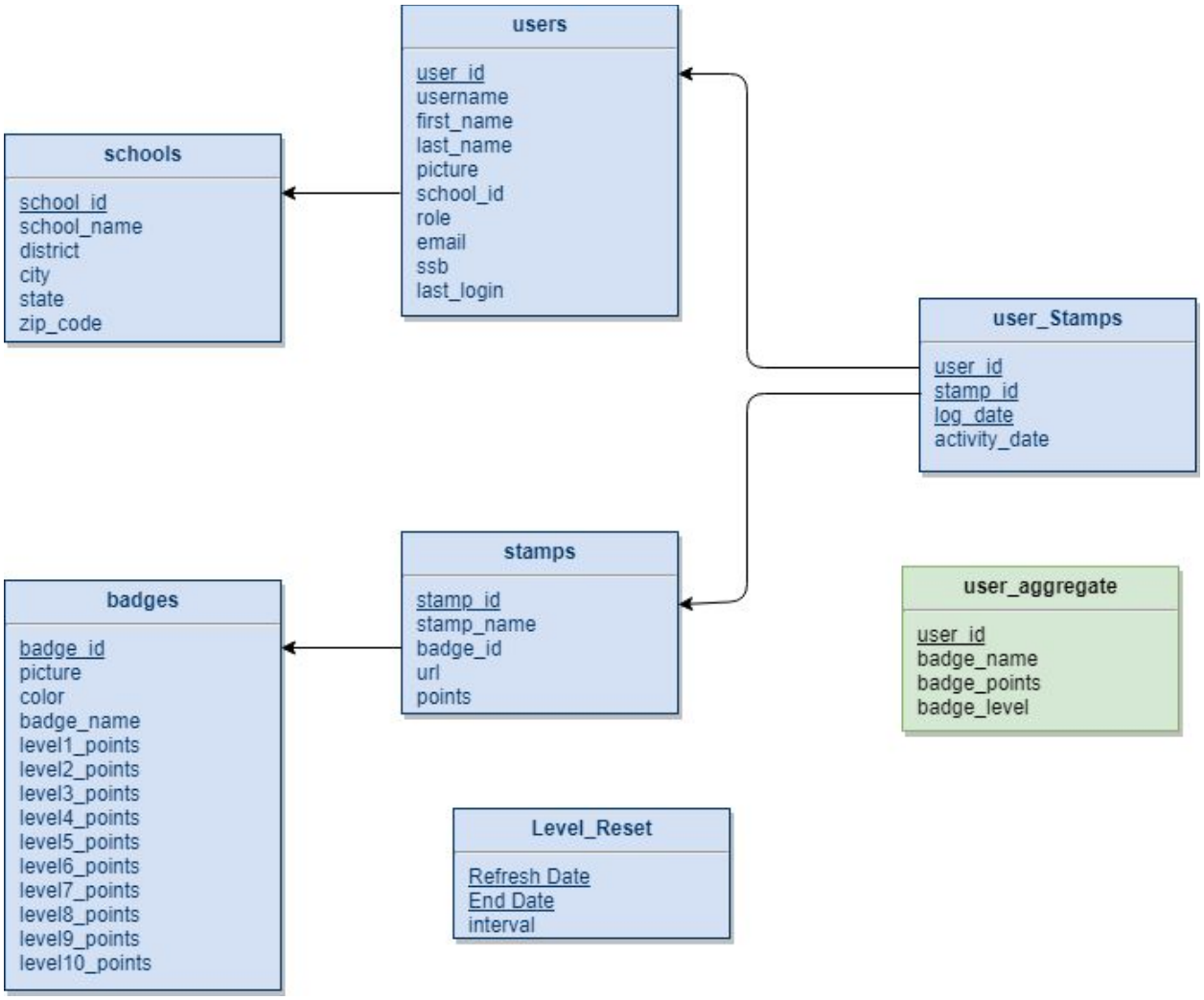
Database Design

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 eventual deployment. 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.



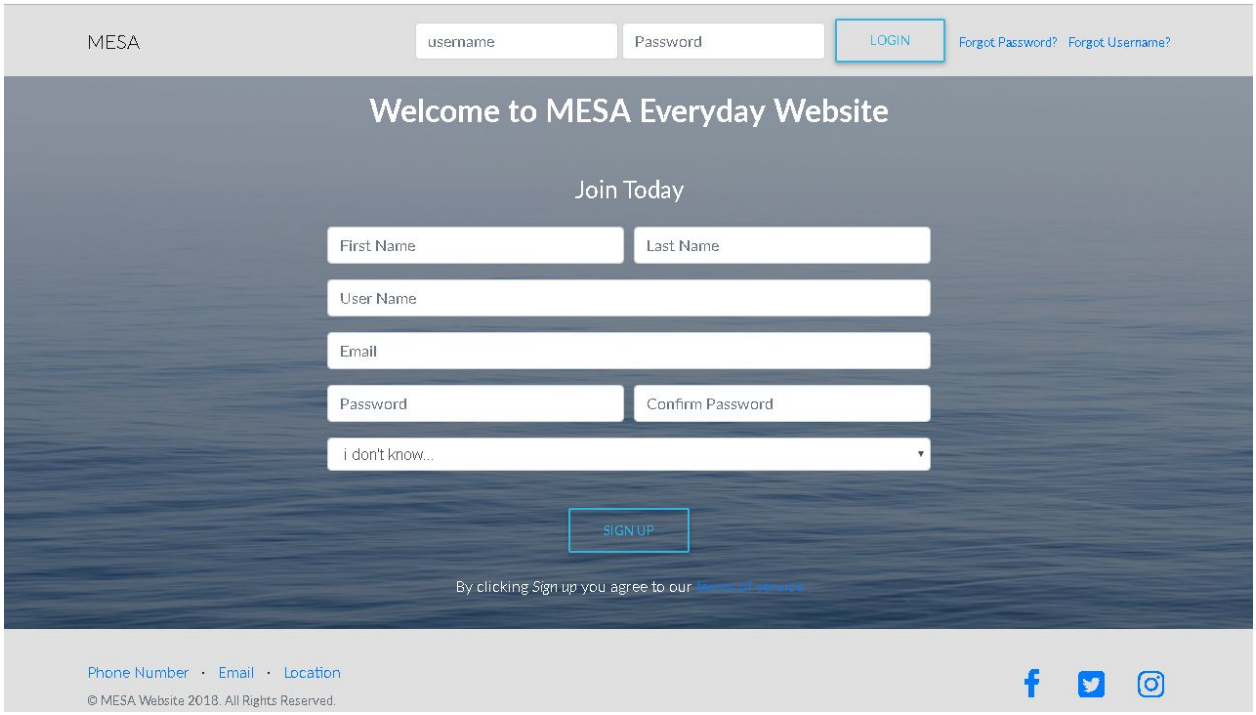
Specific Database Considerations

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, wil have an index for it. Additionally, once the application is completed additional indexes will be made based on the search conditions used throughout the application. 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 have duplicate information throughout the database that could potentially become inconsistent. Additionally, a trigger will be placed on the “role” field in the “users” table to prevent anyone from changing their role to an admin.

System Overview

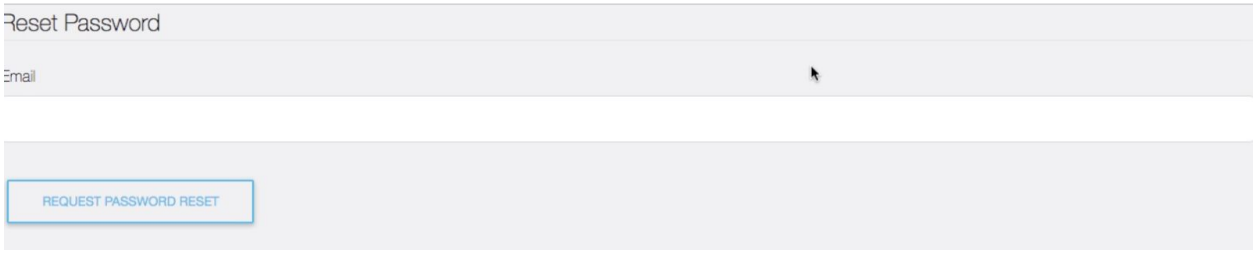
The MESA Everyday web app that our team will be developing during the two terms of capstone will consist of the web pages listed below.

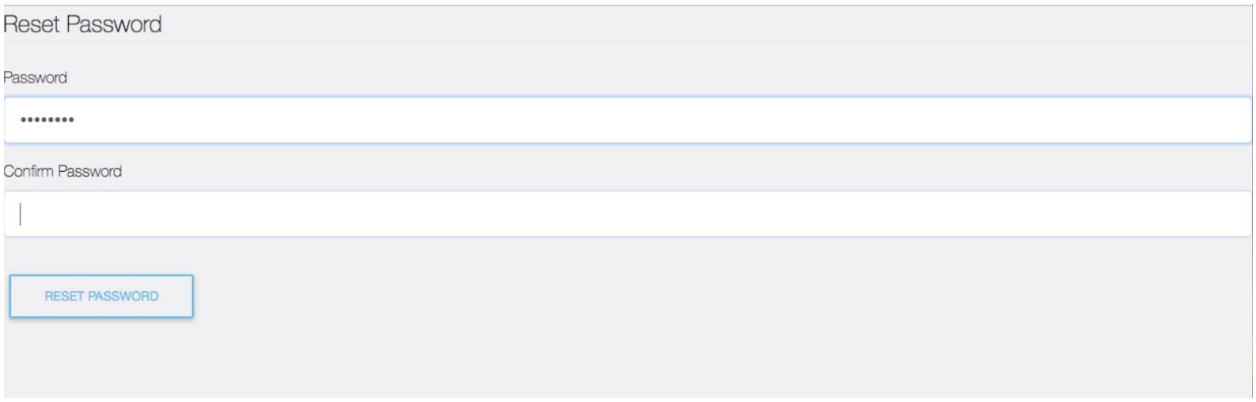
Landing Page



The login form and the registration form are on the same page. The page will not redirect the user to other pages. Moreover, the layout was designed to fix everything in one page without having the user scrolling the page. Mesa logo will be added on the top left of the page. The contact info will be on the footer of the page, the header contains mesa logo, login form, login button, forgot password link and forgot username link

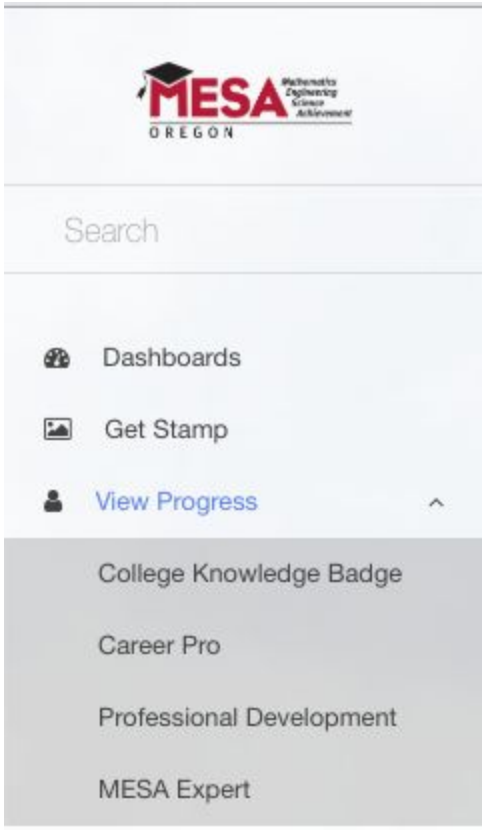
Resetting Password Page





The reset password form has 2 form, when the user click reset password on the landing page, it will redirect user to a page where user can input there email to get the password reset link. Once the user click on the reset link, the user will be redirect to a page for setting up a new password.

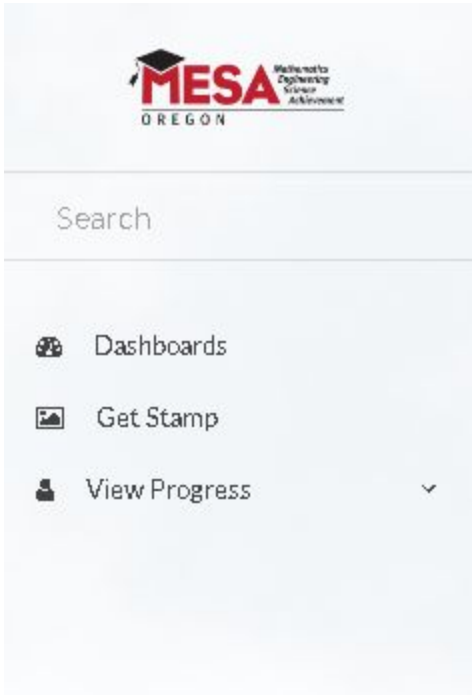
Participant's Side Overview



Participant Side Layout

The participant side layout has a logo on top, a search bar and 3 sections, the first section will be dashboard, where participant can see their progress, calendar, upcoming events in general view. The second section is the get stamp section, where they can add their stamp, the third section is view progress, there is sub menu for 4 badges that pulled dynamically from database.

Side Navigation Bar Section



The side navigation bar can slide out and slide in depending on the size of the screen. It will slide out when the screen size is full, or it will be closed when screen is not full, the user can click on the hamburger icon to manually close or open it.

Top Navigation Bar Section

The top navigation bar will have MESA Everyday title on the left, the navigations will be on the right side. The navigation include a notification dropdown, a calendar button and a profile dropdown. Under profile dropdown, there will be a link to manage account and logout button.

Right Side Calendar Section

There will be a module that will show the current and upcoming months, as well as any events within. Clicking on any events will send users the the google calendar that the module is connected to. This will be done by tweaking some settings on the Google Calendar App, and pulling an iframe that Google generates. We will also grab a link from Google to direct users to the Google Calendar App.
Bartlett

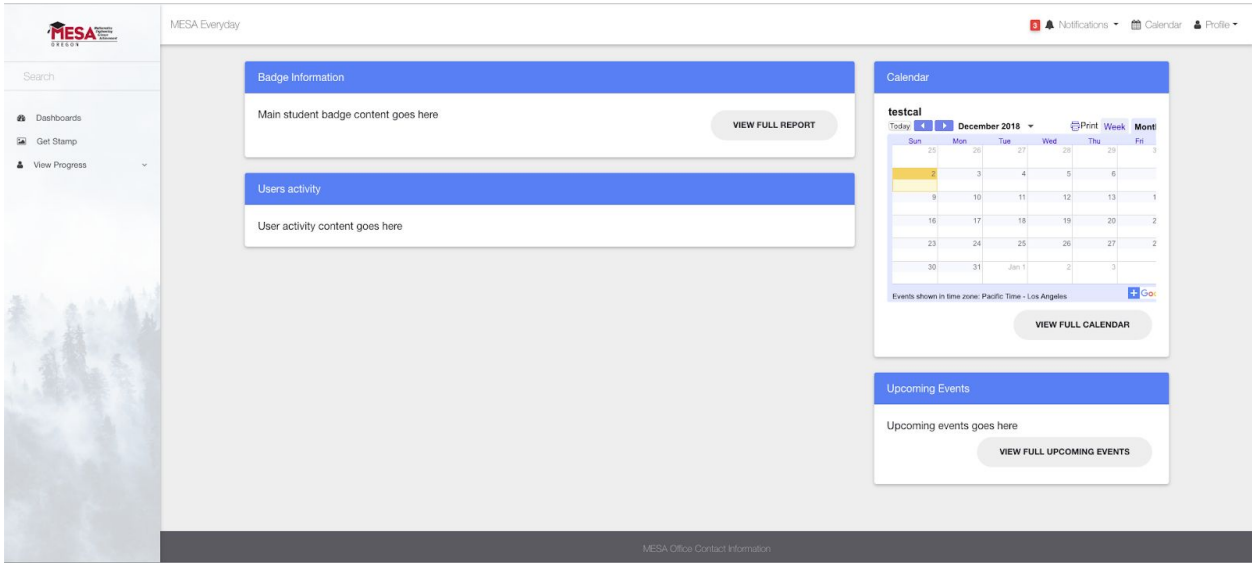
Right Side Events Section

A Section will be dedicated to showing the upcoming events for the month, it will also show major events in the year such as MESA day. This will be done by connecting to the Google API and pulling all Valid Events in the Calendar. There will need to be two types of queries, one to get all events within a month of the current date, and another to grab all of the Large MESA events. The events will then get the days remaining calculated and printed to the user.
Bartlett

Main Section

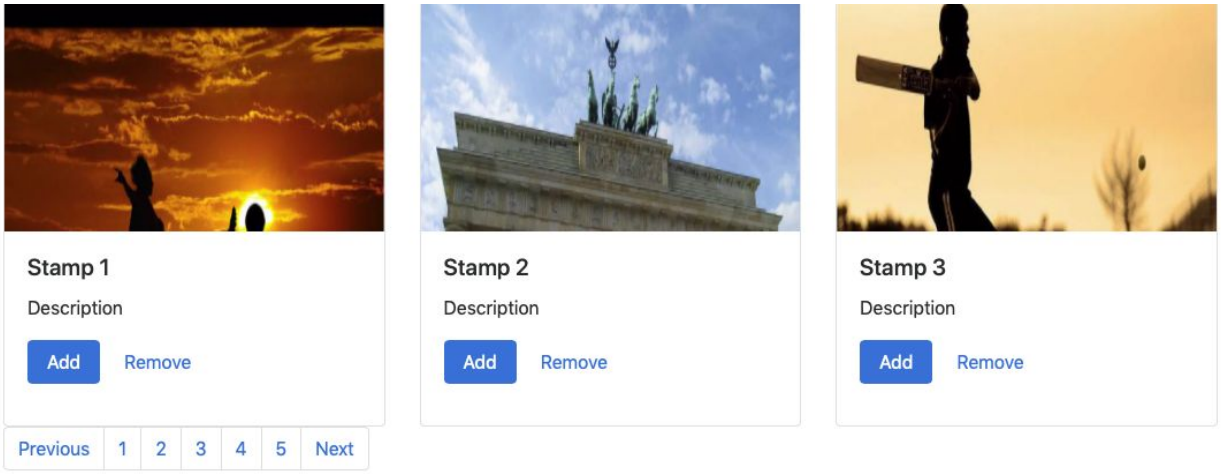
The main section will include 4 cards for badge information, calendar, upcoming event and user activity, the user can click on each card to find more detail on that card.

Dashboard Page



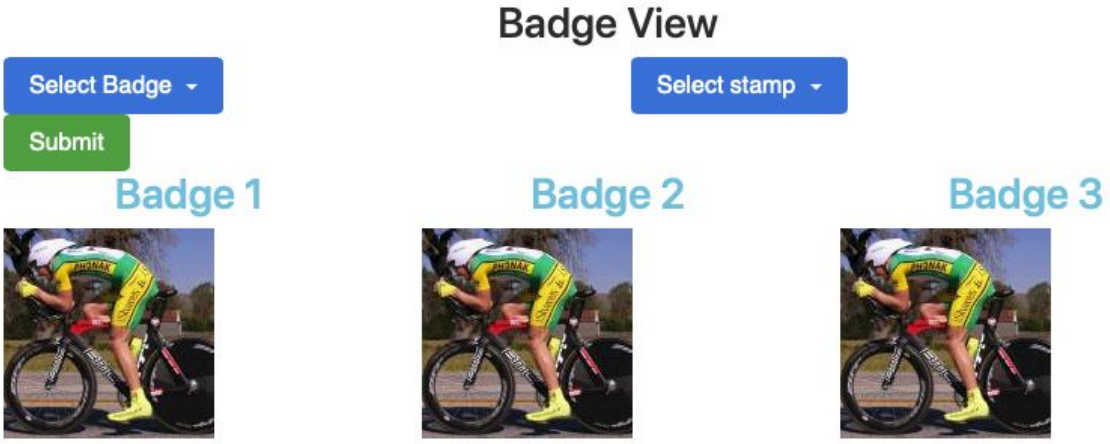
The header of the page is Mesa title and top navigations, the left side will be the side bar that can be slided out or in depending on the screen size, the footer is the contact info section, the main section of the page will contain 4 different cards, each will display general information of each card.

Adding Stamps Page



The adding stamp page will have each card for each stamp with image, title, description and add, remove buttons, the page will be pagination, each page can display upto 3 cards.

Single Badge View Page



The badge view will include title, 1 dropdown for selecting badge, one dropdown for selecting stamp, a submit button, title of the badge, image for badge. The badges will display as a stack. It will display as a list when the screen is small.

Calendar Page

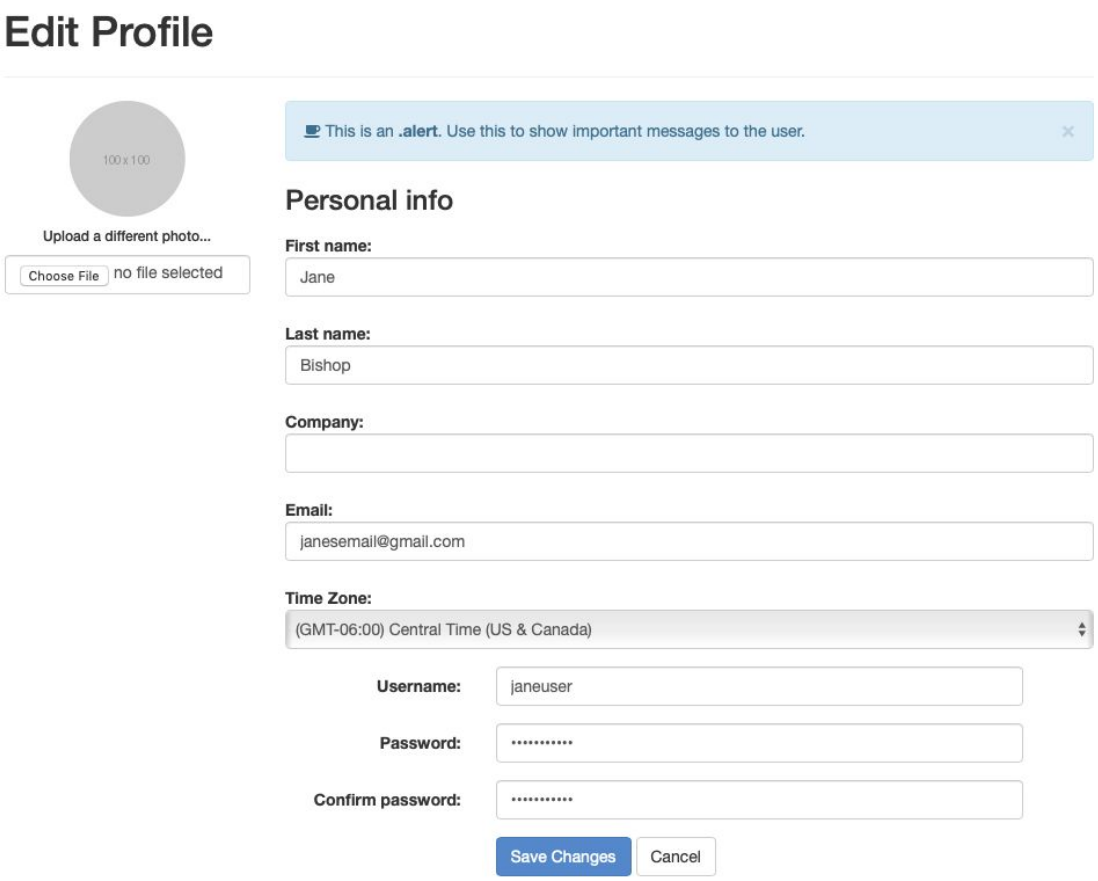
A larger version of the Landing Page's calendar will be located on a separate page. This will show the calendar on a full page using the same method as the Calendar on the Dashboard. There one difference in the two methods and that is just the aspect ratio of the two.

Bartlett

Help Page

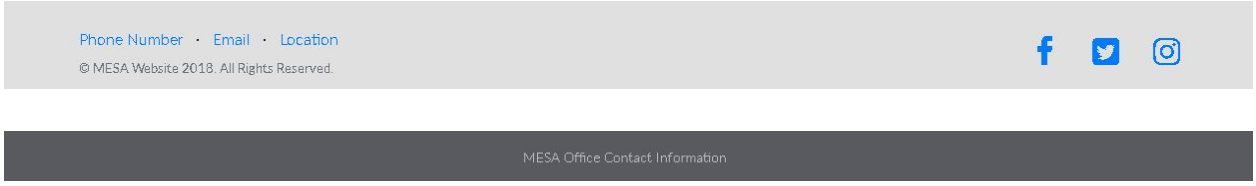
A simple html will contain answers to frequently asked questions about the web app. The page will extend the web app and students can access it with the click of a button. Other notable issues or details the app currently faces may also be updated here.

Editing Profile Page



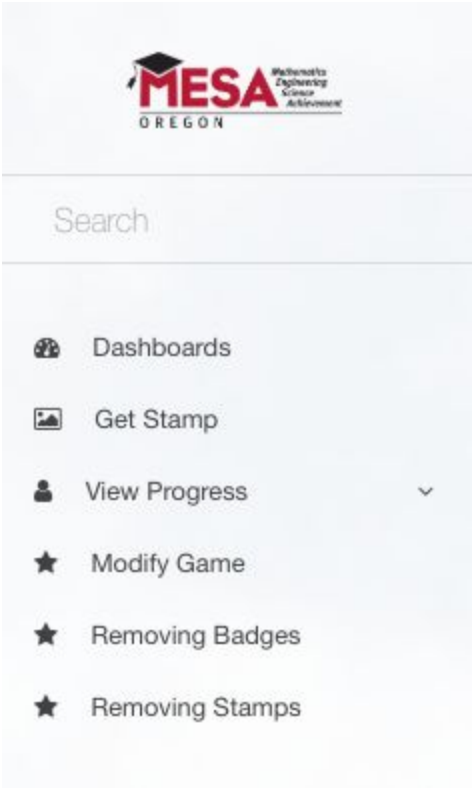
The edit profile page will include a section for upload profile picture (the picture default will from MESA office), a section for user to edit their information like: first, last name, school, email, time zone. There is a section for user to change their username and password.

Footer



The footer is where the user can find the contact information and the link to social network of mesa office.

Administrator Side Overview



Administrator Side Layout

The administrator side layout will look the same as participant side layout, except the administrator side has a menu for modifying game page, removing badge page and removing stamp page

Side Navigation Bar Section

It will include Dashboard Page, Searching a Single Student Page, Manipulating Game Page, removing Badges Page, removing Stamps Page

Top Navigation Bar Section

The top navigation bar of the administrator dashboard page will be the same as the top navigation bar of the participant page.

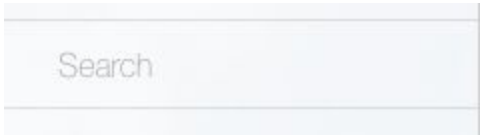
Main Section

The main section will be the same as the main section of the participant page.

Dashboard Page

The administrator dashboard page will show top three scores for each badge

Searching Single Student Page



The user can input the student username to retrieve student dashboard page to view their progress.

Manipulating Game Page

The manipulating game page will Include all that is need to add extra badges and all that is needed to add extra stamps. It will also Include all that is need to alter points for each level and altering max level for badges and all that is needed to set the beginning and end date for which data will be displayed before it's reset.

Removing Badges Page

It will include button for removing badges, a section for display current active badges, a confirmation modal will pop up when the user clicked on the delete button.

Removing Stamps Page

It will include button for removing stamps, a section for display current stamps, a confirmation modal will pop up when the user clicked on the delete button.

Adding and Removing Schools

School dropdown for choosing schools, and buttons below for adding new school or removing a selected school

Removing Accounts

A web page that will be hard to reach so that it can be accessed by accident
It will have the option to search a student by first and last name and picking a student from the returned list and deleting it, but the administrator must confirm that they want to really delete this student by retyping their username and provide a reason from a dropdown menu of reasons to why they are removing the account.
It will also have the option for the administrator to remove an inactive account by picking from a dropdown list from 1-10. Where 1 will be deleting students who have not logged in for a year from the data the button is hit to 10 which will be deleting students who have not logged in for 10 years from the date the button is hit.

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

A participant shall register an account by filling out a registration form where they shall provide first name, last name, and a valid email address. In addition, they shall also provide a valid password and confirm it by entering it again. Finally, they choose the school that they attend. A valid password meets the following minimum requirements: it must be at least eight characters long and include one symbol and one numerical digit. There are restrictions as to which symbols are valid, and they are specified in the design document. The password and confirm password fields must match in order for the registration to be processed. A student must choose a school that they are part of, and if none of the schools in the list apply to them, they can choose “other.” When the registration is processed successfully, the participant will have an immutable username generated for them and will be assigned an avatar from a list of preset avatars. The participant will also receive an email with their user name at the email address they entered.

Signing In

A successfully authenticated participant shall be able to sign out by clicking a “sign out” button.

Signing Out

A successfully authenticated participant shall be able to sign out by clicking a “sign out” button.

Resetting Password

A participant shall have the ability to reset their password by entering their email address and having an email sent to them with a unique link that will open a form with all the necessary information to reset their password. The form shall contain at minimum a new password field and a confirm password field.

Forgetting Username

A participant shall be able to regain their username by entering their email address and having an email sent to them with their username.

Deactivating Account

A successfully authenticated participant shall have the ability to deactivate their account by clicking on a "deactivate account" link that will be located in a place on the website that is not easy to click accidentally. Before successfully processing account deactivation, a participant shall verify that they want
to deactivate their account by entering in their username and hitting a "confirm deactivation" button. A deactivated account shall not be recoverable, and all data associated with it shall be deleted.

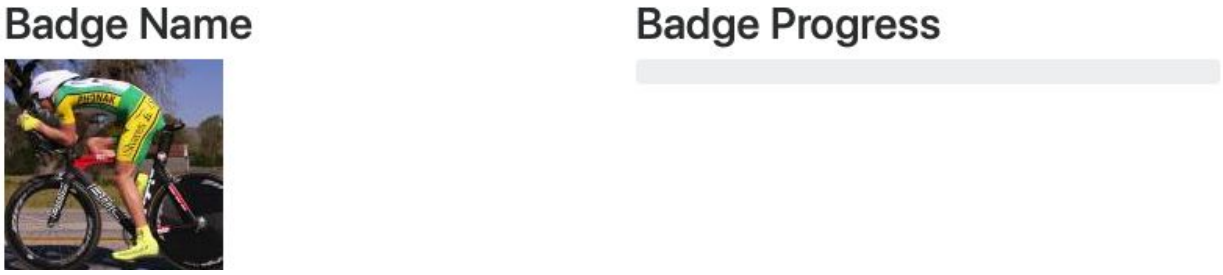
Editing Profile

A successfully authenticated participant shall be able to edit information about themselves. They shall be able to change their first and last name, as well as their school. They shall also be able to change the avatar associated with their profile from a limited list of avatars. They shall be able to change the email address tied to their account and change their password, but they shall not be able to change their username.

Participants Tasks Design

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

Overall Visibility



For badges that a student has not yet obtained, their current level will be zero, there will be a badge picture as well as badge name.

Adding Stamps

To acquire a stamp for a badge, there will be a query to the database that looks for all the stamps a authenticated participant has not acquired. The participant should choose one from them as well as choosing the date they met the requirement to earn the stamp. Upon successfully adding a stamp, the actual time it was acquired will be logged in the database.

Viewing Currently Obtained Stamps

To view the currently obtained stamps, there will be a query to the database that looks for all stamps a user has earned since the last reset date. The results will be displayed into a table with the following information: Stamp name, the date of the completed activity, and the points it is worth. This section will display at most 10 stamps at once, with buttons at the bottom to select which stamps should be visible (i.e. 1-10, 11-20, etc...).

Viewing Needed Stamps

To view the list of stamps that an authenticated participant has not acquired for each badge, there will be a query to the database that looks for all stamps a user has not acquired since the last reset date. The result will be displayed as a table as obtained stamps, but without the visible selecting button.

Viewing Current Level

For each badge, a successfully authenticated participant shall be able to view their current level, which is determined based on the number of points the participant has accumulated since the most recent reset date. The current level of a user shall increase and decrease based on the number of points set for each level of a badge. The number of points for each level is subject to change as the rules of the game change.

Viewing Currently Accumulated Points

To check the currently accumulated points, there will be a query to the database looking for the points an authenticated participant has earned for each badge. The result will be displayed as a table with the following information: badge name and the accumulated points for that badge.

Viewing Needed Points for Next Level

To check the needed points for next level, there will be a query to the database looking for the total points set for a given level of a badge. The needed points will be the subtraction of the total points and currently accumulated points.

Removing a Stamp

To remove an acquired stamp, an authenticated participant shall select a stamp that has been acquired. The selected stamp will be removed from the obtained list in the database, the action will be logged for adding back in the future.

Participants Calendar Design

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

Viewing Current Month's Calendar

The Current Month’s Calendar is displayed based off of a given Google Calendar. The Owner of that calendar will need to make sure that they have allowed sharing and embedding of that calendar. After that, the Calendar will provide the user with some HTML to paste into the site.
Bartlett

Viewing General Upcoming Task for Current Week

To view general upcoming Tasks, the App will pull events through the Google API and add them to a list of the next 30 days, then they will have the days remaining calculated. The ending list will then be split into groups based off the color of the events (which corresponds to the badges they belong to. These can then be presented to the User.

Bartlett

Viewing Countdown for Three Important MESA Dates

This will be solved like the general events except that they will need to be parsed via name. Other than that, the queries are the same.

Bartlett

Viewing Upcoming Tasks Based on Badge

Following the same steps as before, we can divide the events into groups by color (as before) and display the right events on the Badge pages.

Bartlett

Calendar Security

Users will be unable to edit the calendar, and all events will be created by the owner of MESA's Google Calendar.

Bartlett

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

An administrator(admin) will fill out a registration form. They must provide a valid email address, a valid password, and a password confirmation. A valid password must be at least eight characters long and include at least one symbol and one numerical digit. The password and confirmation fields must match in order for the registration to be processed. After the registration is processed successfully, the admin will have an avatar generated for their profile page and receive their username by email.

Signing In

An administrator will be able to sing in once registered successfully. From the landing page, they must enter their username and password and click the "sign in" button to sign in. Once authenticated, the they will be redirected to their dashboard page.

Signing Out

An authenticated administrator will be able to sign out by clicking a “sign out” button from their dashboard. Once signed out, they will be redirected to the landing page.

Resetting Password

From the landing page, an administrator can click on “Forgot Password?” if they do not remember. They will be prompted with the email they entered at registration. An email will be sent to them with a unique link that will open a form. The form will contain new password and confirm new password fields. Upon successful submission, their password will be reset.

Forgetting Username

From the landing page, an administrator can click on “Forgot Username?” if they do not remember. They will be prompted with the email they entered at registration. Their username will be emailed to them.

Administrator Tasks Design

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

Adding Schools

A successfully authenticated administrator shall have the ability to add schools that join the MESA program to the web app.

Adding Extra Badges

A successfully authenticated administrator shall have the ability to modify the rules of the game by adding extra badges in addition to the badges that come pre-built with the web app.

Adding Extra Stamps

A successfully authenticated administrator shall have the ability to modify the rules of the game by adding extra stamps for any given badge of their choice, including stamps associated with badges that they have added and that did not come pre-built with the web app. A set of pre-built stamps shall be associated with the pre-built badges as stated in the "required base data" section of this document.

Altering Points Required for Badge Levels

A successfully authenticated administrator shall have the ability to modify the rules of the game by increasing or decreasing the total number of points necessary to earn a higher level of any given badge of their choice, including badges they have added that did not come pre-built with the web app.

Altering Badges Max Level

An administrator can achieve this by simply inputting the number of points required for each level of the badge up until the max level badge that they want to set. Alternatively, to remove a badge level they input a zero for the given level.

Viewing Single Student Detailed Progress

A list of all students who match a search criteria will be displayed and by clicking on a name from the displayed list, an admin shall be granted a view only access to all the details that a participant can see by logging in to their account. The admin will be able to see each level a student is at, the stamps each student have earned, and what stamps the student can still earn. This information will be obtained and displayed in the same way it is done on the participant side of the application.

Viewing top three scores for each badge

A table will be displayed with the following information: badge name, the students with the top three scores for that badge. If there are multiple students with the same top score, all of them will be listed for that badge.

Removing Schools

A successfully authenticated administrator shall have the ability to remove schools that are no longer part of the MESA program and/or schools they have entered by mistake from the web app.

Removing Badges

A successfully authenticated administrator shall have the ability to modify the rules of the game by removing badges, including badges that come pre-built with the web app. A successfully authenticated administrator shall also be able to remove any badges that they have added manually that did not come pre-built with the system.

Removing Stamps

A successfully authenticated administrator shall have the ability to modify the rules of the game by removing stamps for any badge of their choice, including stamps associated with badges that they have added that did not come pre-built with the web app.

Exporting Data to Excel Sheet

To export data to excel, first there will be a query to obtain user stamps from the database (including those from before a refresh date). Then the data will be written to file using already existing methods from various python libraries. Finally, the Flask method “send_file” to send the file to the admin’s downloads folder.

Removing Old Accounts

To remove old accounts, an administrator must first select how long the accounts should be inactive for before removing them. This number will be set at default to 5 years, no less than 1 year, and no greater than 10 years. Once this time has been selected and the admin pushes a submit button, a query will run that deletes any accounts that have not been logged in since the time period selected by the admin. All stamps that user has earned will also be deleted from the database.

Search and Delete a Single User Account

An administrator should be able to delete a student's progress: first, the admin should search for a student through a combination of first and last name, there will be a query to the database that looks for the student, once found, the admin should confirm that he(she) wants to delete the account and provide the reason for the account deletion. Account deletion history will be logged in a text file.

Setting Academic Year Date

An administrator should be able to set the beginning and end date for which a game starts and ends. The dates will be recorded in the database, any game with an expired time will not be displayed.

Detailed System Design

The section provides the detailed algorithm of how the different functionalities that the web app will support should be implemented. For some sections it can be as detailed as pseudo code.

Participants Detailed Design

This section of the design discusses in details 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 Detailed Design

This section discusses the algorithm level 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

TBD

Signing In

TBD

Signing Out

TBD

Resetting Password
TBD

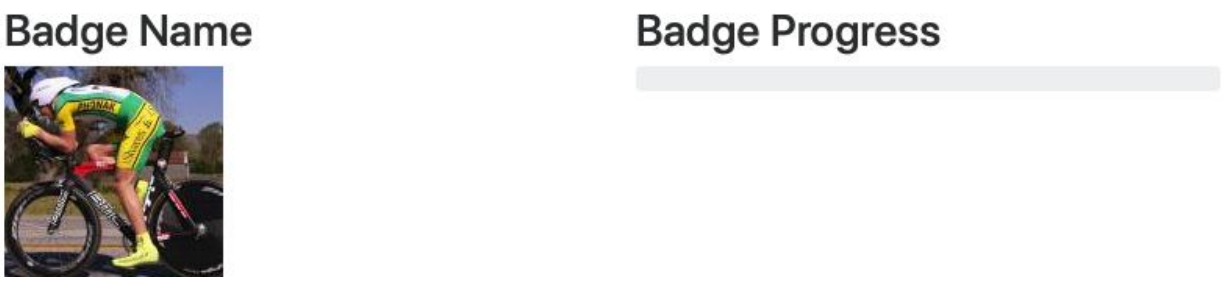
Forgetting Username
TBD

Deactivating Account
TBD

Editing Profile
TBD

Participants Tasks Detailed Design
Below is the detailed design for functionalities that a participant gets upon their successful authentication to the web app.

Overall Visibility



For badges that a student has not yet obtained, their current level will be zero, there will be a badge picture as well as badge name.

Adding Stamps
Text
Luo

Viewing Currently Obtained Stamps
Text
Cohoe

Viewing Needed Stamps
Text
Luo

Viewing Current Level

TBD

Viewing Currently Accumulated Points

Text

Luo

Viewing Needed Points for Next Level

Text

Luo

Removing a Stamp

Text

Luo

Participants Calendar Detailed Design

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

Viewing Current Month's Calendar

Text

Bartlett

Viewing General Upcoming Task for Current Week

Text

Bartlett

Viewing Countdown for Three Important MESA Dates

Text

Bartlett

Viewing Upcoming Tasks Based on Badge

Text

Bartlett

Calendar Security

Text

Bartlett

Administrator Detailed Design

All administrators related designs listed below are not part of the MVP. This section of the design discusses the algorithm level details for 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 Detailed Design

This section discusses the algorithm level 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

Text goes here

Wan

Signing In

Text goes here

Wan

Signing Out

Text goes here

Wan

Resetting Password

Text goes here

Wan

Forgetting Username

Text goes here

Wan

Administrator Tasks Detailed Design

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

Adding Schools

TBD

Adding Extra Badges

TBD

Adding Extra Stamps

TBD

Altering Points Required for Badge Levels

TBD

Altering Badges Max Level

Text goes here

Cohoe

Viewing Single Student Detailed Progress

Text goes here

Cohoe

Viewing top three scores for each badge

Text goes here

Luo

Removing Schools

TBD

Removing Badges

TBD

Removing Stamps

TBD

Exporting Data to Excel Sheet

Text goes here

Cohoe

Removing Old Accounts

Text goes here

Cohoe

Search and Delete a Single User Account

Text goes here

Luo

Setting Academic Year Date

Text goes here

Luo