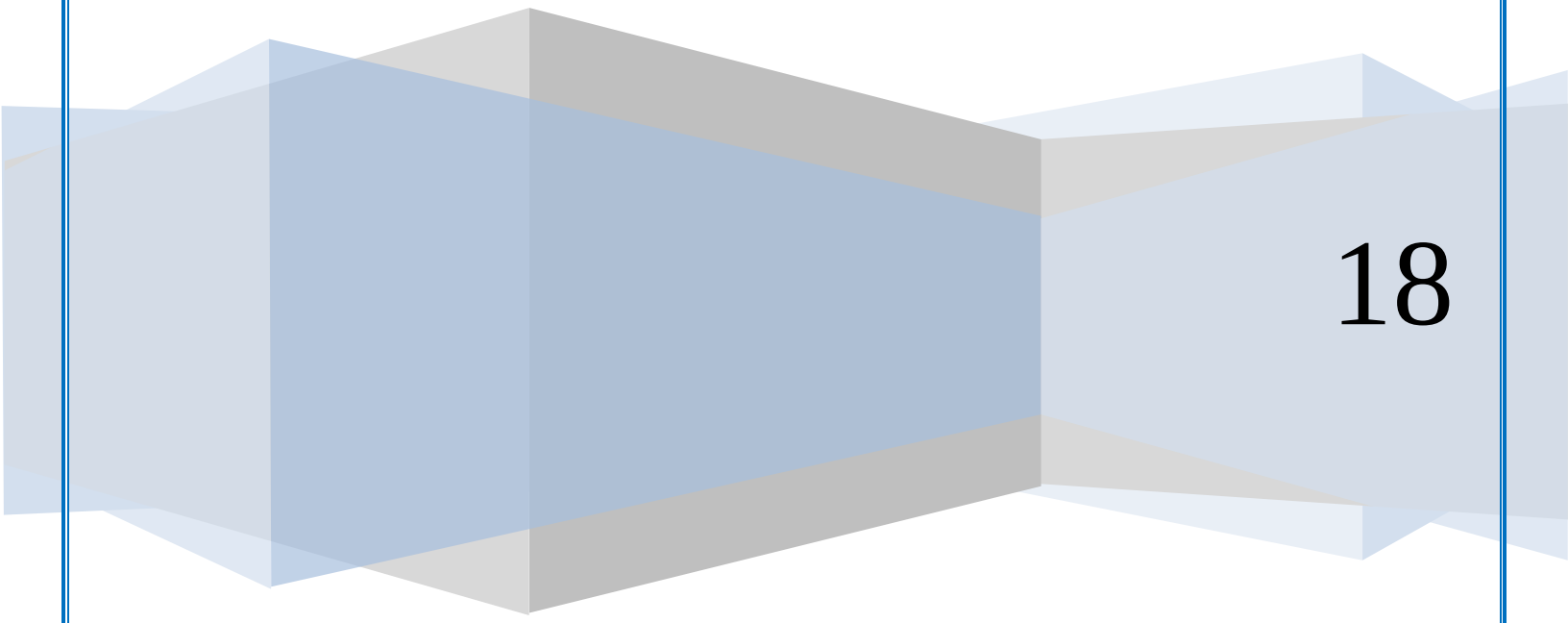
Portland State University - Oregon MESA

MESA EVERYDAY

Requirements Document

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



18

Table of Contents

Introduction

MESA Everyday is a web app that gamifies the MESA experience for students who are part of the MESA 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. It will also help students participate in more MESA-sponsored activities and events that might allow them to obtain college credits for work that they have done. This document describes the purpose, use, functions, and requirements of the MESA Everyday web application.

Purpose and Scope

This document serves as a statement of the required functions and capabilities that the developers should implement, as well as a way to let the client know exactly what will be built. The document includes the requirements for the minimal viable product (MVP) and several extra requirements that are not part of the MVP. Any feature in the document not designated as part of the MVP may end up being implemented, but the developers cannot guarantee that it will be delivered as part of the end product. In this document, the target audience should be able to learn about the user and system requirements the software should implement.

Target Audience

This document targets all the representatives of the Portland State University MESA program who have a stake in this project, as well as the web app developers and faculty members who are part of the Portland State University (PSU) Computer Science (CS) Capstone.

Terms and Definitions

Oregon MESA	MESA stands for Mathematics, Engineering, Science, Achievement. Oregon MESA is a pre-college academic program hosted by Portland State University that equips underrepresented students with science, technology, engineering and math (STEM), invention, and 21st-century skills.
MVP	MVP stands for Minimal Viable Product. It describes the set of requirements that the client should expect to have delivered by March 15, 2019. Any requirements not designated as part of the MVP are not guaranteed to be delivered to the client, but the developers will pursue all needed resources to attempt to make them a part of the end product.
Badges	Different classes of MESA activities and events each containing multiple stamps
Stamps	Detailed list of activities and events that all belong to a category badge
Users	A general term given to people who use or operate the app. A user is classified as either being a Participant or an Administrator. Depending on the class of the User, certain functionalities are enabled.
Participants	A class of Users who are part of the MESA program and who have restricted functionality.
Administrators	A class of Users who are part of the MESA program who are in charge of monitoring participants' progress and thus need more visibility and less restricted functionality.
Client	The organization or a representative of the organization for which the app is built.
Flask Web Framework	Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.
Bootstrap 4	Bootstrap 4 is the newest version of Bootstrap, which is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Bootstrap 4 is completely free to download and use.
MySQL	MySQL is an open source relational database management system.

CAT	CAT stands for Computer Action Team. The CAT provides IT support throughout the Maseeh College of Engineering and Computer Science (MCECS). With a primary focus on instructional needs, the CAT supports many large-scale computer labs (both college-wide and departmental), remotely accessible compute/session servers, various remotely accessible services, and the server and physical network infrastructure that binds it all together. Where possible, the CAT is also able to leverage its infrastructure to support research and special projects in the College.
Virtual Host	Virtual hosting is a method for hosting multiple domain names (with separate handling of each name) on a single server (or pool of servers). This allows one server to share its resources, such as memory and processor cycles, without requiring all services provided to use the same host name.
SSL Certificate	SSL Certificates provide secure, encrypted communications between a website and an internet browser. SSL stands for Secure Sockets Layer, the protocol that provides the encryption. SSL Certificates are typically installed on pages that require end-users to submit sensitive information over the internet, like credit card details or passwords.
Domain	Domain names are used to identify one or more IP addresses and are used in URLs to identify particular web pages. Instead of remembering a group of numbers to access a website, a meaningful name is given to it.
IP address	An Internet Protocol (IP) address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. An IP address serves two principal functions: host or network interface identification and location addressing.
IoT Devices	The Internet of things (IoT) devices are the network of devices, vehicles, and home appliances that contain electronics, software, actuators, and connectivity, which allows these things to connect, interact, and exchange data.
XSS	Cross-site scripting (XSS) is the concept of injecting arbitrary HTML (and with it JavaScript) into the context of a website.
html	Hypertext Markup Language (html) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.
Jinja2 Templates	Jinja is a template engine for the Python programming language and is licensed under a BSD License created by Armin Ronacher. It is similar to the Django template engine but provides Python-like expressions while ensuring that the templates are evaluated in a sandbox.
Developers	The capstone team that is in charge of building this web app. The developers include Christopher Bartlett, Michael Cohoe, Fadi Labib, Minwei Luo, Minh Nugyen, Thong Tran, and Millen Wan.

Required Base Data

This section enumerates the required data that the app will manage for multiple participants.

Badges	Stamps	Pts.
College Knowledge	Attend MESA Day	5
	Attend a MESA college tour	3
	Go on a college tour	3
	1:1 Meeting with College Expert	2
	Complete a Scholarship Application	1
	Create a plan for paying for college	1
	Fill out the FAFSA	1
	Fill out the OSAC	1
	Learning Styles Activity	1
	College Lingo Activity	1
	FAFSA Webinar	1
	Filling out OSAC Webinar	1
	Types of Colleges Webinar	1
	Attend tutoring sessions	1
	Create the perfect study session webinar	1
	Take the PSAT or PACT	1

	Take the SAT, SAT II, ACT	1
	Create your schedule activity	1
	Attend a college student panel	1
Career Pro	Attend Demo Day	2
	Career Exploration activity	1
	Go on a Career field trip	1
	Get Real Activity	1
	Complete a Job Shadow	1
	Complete an Internship	1
	Interview someone in your field	1
	Apply to a job	1
	Career Personality Test	1
	Create a Professional Email signature	1
	Hold a part time job	1
	Interview questions Activity	1
	Attend a Career talk	1
Professional Development	Elevator Pitch Activity	1
	Resume workshop	1
	Mock Interviews	1
	Write a mock cover letter	1
	5/10 year plan worksheet	1
	Budget Prediction Activity	1
	How to prioritize	1
	2 - 1 - 6 - 1 plan	1
	Skills vs traits Activity	1
	How to Manage Time Activity	1
	Set up a Google Calendar	1
MESA Expert	Turn in a MESA application	1
	Turn in MESA Day National Form	1
	Attend a Family Night	1
	Compete at MESA Day	1
	Showcase at Demo Day	1
	Invention Ambassadors	1
	Turn in a Video for Fall Competition	1
	Compete on a Nationals Track team	1

Functional Requirements

Functional requirements are the services that the web app needs to provide, including how it responds to users' inputs and how users should interact with it.

Participants' Requirements

This section discusses all the functionality that a user 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

This section discusses the set of participant functional requirements that relate to how the web app manages a participants' session. This includes registering an account, signing in, signing out, dealing with forgetting username or password, deactivating an 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 participant shall sign in by entering their username and password.

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

A participant that has been successfully authenticated to the web app shall be able to gain all functionality listed below to the extent defined in this requirements document.

Overall Visibility
Upon successfully signing in, a user shall be able to see their current level in each badge, including badges that they have not obtained.

Adding Stamps
For each badge, a successfully authenticated participant shall be able to acquire a stamp for that badge by choosing the stamp name and the date they met the requirements to earn the stamp. Upon successfully adding a stamp, the time is logged with the entry to record the actual time the stamp was earned.

Viewing Currently Obtained Stamps
For each badge, a successfully authenticated participant shall be able to view a list of stamps that they have completed for each badge. Only stamps that have been completed since the most recent reset date should be listed.

Viewing Needed Stamps
For each badge, a successfully authenticated participant shall be able to view a list of stamps that they have not acquired for each badge. Only needed stamps that have not been completed since the most recent reset date should be listed.

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
For each badge, a successfully authenticated participant shall be able to check how many points they have acquired for each badge since the most recent reset date.

Viewing Needed Points for Next Level
For each badge, a successfully authenticated participant shall be able to check how many points they need to acquire to reach next level for each badge, which is determined by the total point set for a given level of a badge.

Removing a Stamp
For each badge, a successfully authenticated participant shall be able to remove a stamp for the badge that they have acquired. When the stamp is deleted, the user should be able to add the stamp again in the future.

Participants' Calendar
Upon successfully signing in, a participant shall be able to see a calendar, which will show them upcoming events hosted by MESA. They shall also be able to see notifications and countdowns for important events.

Viewing Current Month's Calendar

Both the Landing Page and Full Calendar Page shall require a Google account with a calendar to link to (this can be a specific calendar, not just the default one). All days and events shall be populated via that Google Calendar.

Viewing General Upcoming Tasks for Current Week

In order for participants to view events, there are two components that MESA needs to update. First, events will need to be added to the correct calendar. Second, the events need to be added on the correct day and with the correct information. Once these two requirements are satisfied, the events shall be shown in the application. Each event for the current week will have a countdown for when it is due.

Viewing Countdown for Three Important MESA Dates

The main events that need to be displayed to users (like MESA day), shall be named according to a predefined format so that the website can check for it. This can be something simple like: “MESA day at Portland High School” as long as it fits the predefined format. The MESA dates for which this shall be needed are MESA Day and all Demo Days.

Viewing Upcoming Tasks Based on Badge

Events shall be tied to badges via color-coding, so the events in the Google Calendar shall be colored according to the badge they belong too. These shall then be brought into the app and categorized.

Calendar Security

All users on the app will only have the ability to read from the calendar. The owner of the MESA’s Google Calendar will need to update it via Google Calendar. The Google Calendar should also be made public if the app is to redirect users to the Google Calendar App.

Administrator Requirements

All administrator-related requirements listed below are not part of the MVP. This section of the requirements discusses all the functionality that a user 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

This section discusses the set of administrator functional requirement that relates to how the web app manages an administrator session. This includes registering an account as admin, signing in, signing out, and dealing with forgetting a username or password.

Registration

A pre-built account created by the client shall be used as the admin account tied to the app. There will be a pre-set admin username based on the email tied to the admin account, and that username will be used to authenticate to the app and gain administrator privilege.

Signing In

The first time an administrator signs into the app, they will need to use a pre-set password and the pre-set admin username. The password shall be resettable through the resetting password functionality.

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

Resetting Password
An administrator shall have the ability to reset their password by having a password resetting email sent to the email address tied to the admin account. From this email, they will be able to click a link that will allow them to change their password.

Forgetting Username
An administrator shall have the ability to have the pre-set username sent to them by e-mail if they forget it. An administrator will be able to see their username in the email, but there will be no way to change it.

Deactivating Account
The administrator account cannot be deactivated.

Editing Profile
There is no profile maintained for the administrator account and thus no profile to edit.

Administrator Tasks
An administrator that has been successfully authenticated to the web app shall be able to gain all functionality listed below to the extent defined in this requirements document.

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
A successfully authenticated administrator shall have the ability to modify the rules of the game by raising or lowering any badge to any level, including badges they have added that did not come pre-built with the web app. Each badge shall not be lower than a single level and not higher than 10 levels.

Viewing Single Student Detailed Progress

An administrator shall have the ability to view a student's progress by searching for them through a combination of their first and last name.

Viewing top three scores for each badge

An administrator shall have the ability to see the students with the top three scores for each badge. If there are multiple students with the same top score for a given badge, all of them will be listed for that score.

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

An administrator shall have the ability to export all the stamp/badge/user data to an Excel file. This file shall be saved to the downloads folder of the computer that the administrator is on.

Removing Old Accounts

An administrator shall have the ability to remove accounts that have not been active for a certain period of time. The administrator can enter the time to be checked. This number should be no lower than 1 year and no greater than 10 years.

Search and Delete a Single User Account

An administrator shall have the ability to delete a student's progress by searching for them through a combination of their first and last name, confirming that they want to delete the account, and explaining the reason for the account deletion. Account deletion history shall be logged in a text file.

Setting Academic Year Date

An administrator shall have the ability to set the beginning and end date for which the game will start and end. The administrator can set the number of years. When the time expires, no new data will be displayed until the date range is updated.

Administrator Calendar

The administrator side of the application shall have no control over the calendar through the app. In order to modify the events in a calendar, an administrator must log into the account that the Google calendar is tied to and do any modification there. After this, it shall be updated on the participant's side of the app.

Non-Functional Requirements

Technology

The web app shall be built using Python 3.7 as the programming language and Flask as the framework. For the frontend, the web app takes advantage of Bootstrap 4, and the backend is managed through a MySQL database. HTML and Jinja2 templates are used to generate the different web pages in the web app.

Hosting

Testing Stage Hosting

During the duration of testing, the web app shall be hosted through the PSU Computer Action Team using their infrastructure. A virtual host with the flask web app running on it serves the web app. The web app will be accessible through the domain *mesaeveryday.cs.pdx.edu* and will have an SSL Certificate. The site shall be set up in a way where it will not be accessible outside of the PSU address space.

Production Stage Hosting

If the web app makes it to production, the web app shall be hosted through the PSU Computer Action Team (CAT) using their infrastructure. A virtual host with the flask web app running on it serves the web app. In order for the web app to make it to production hosting, the client will purchase a domain that will be bound to the virtual host serving the web app. The client domain will be bought through Gandi and will be *mesaeveryday.org*. The CAT will be able to obtain an SSL certificate for the web app. If the client will not be able to buy a domain, then production hosting will not be a deliverable of this project.

Capacity

Capacity, as defined here, refers to the amount of physical computer storage available to store the web app's persistent data. Given the type and volume of information stored by this web-app projected to the year 2025 and the amount of storage available through the CAT's database (db.cecs.pdx.edu), capacity should not be an issue. The web app shall have three different databases in *db.cecs.pdx.edu* tied to it. The first database is devmed, which will contain fake data used for developing the app functionality and conducting basic testing. The second database is testmed, which will be used during the actual testing stage of the web-app once all functionality is built. The third database is prodmed, which will be the production database storing real student data if the app makes it to production.

Scalability

Due to a limitation of the framework, the app has limited scalability, but due to the type and volume of data stored and managed by this web app, scalability shall not be an issue in the near future.

Adaptability & Accessibility

The web app shall be built to be adaptable to variable screen sizes and variable internet browsers. This way the app is accessible in any device, including desktops, tablets, smartphones, and any other IoT devices that have web access and a web browser.

Reliability & Maintainability

The app shall be highly reliable since it is maintained by the CAT, but no system is one hundred percent reliable. In addition, any code is inherently unreliable if not regularly maintained. In order to keep the web app at the same reliability level as other CAT services, the client must insure that the code is maintained. While CAT will tend to their infrastructure by making sure it is secure and up-to-date and by insuring its reliability, they will not tend to the code of this web app and will not maintain it. Making sure the app is functional, bug fixed, and maintained in case of failure resulting from OS/Software update is the responsibility of the client. The developers building this app will not be responsible for maintaining the code after the duration of their capstone service expires.

Availability

The app will be highly available as it is maintained by the CAT, but no system has one hundred percent availability. In the case that the app is not available, CAT can be contacted to start the app, however if staring the app results in errors, they will not attempt at fixing it and the web app will be no longer available until the person responsible for the maintenance of the app fixes whatever is wrong with it. Neither the CAT nor the developers can insure long-term availability of the system. The developers building this app will not be responsible for insuring the availability of the code after the duration of their capstone service expires.

Recoverability

The database and the server the app will be hosted on will be snapshotted every 14 days (number of days is up for negotiation with the CAT). The CAT will be in charge of recovering the backups if something goes awry.

Security

XSS Mitigation

While flask deals with most of XSS by escaping all values in Jinja2 templates, any html not generated through Jinja2 templates is vulnerable to XSS. In addition, calling markups on data submitted by users and sending out html or text files from uploaded files cause the web app to be vulnerable to XSS. The web app shall generate all html pages using Jinja2 templates and shall not accept any uploads of any type or have any fields that need to be exempted from escaping. Furthermore, most of the filed have limited input size or are dropdowns and date type inputs, which are not vulnerable to XSS vulnerabilities.

Jinja2 Templates cannot protect from XSS by attribute injection. To mitigate this, developers shall always quote attributes with double quotes when using jinja expressions (ex. *<input value="{{ value }}">*). Also the *a* tag's *href* attribute shall not contain a JavaScript URI, as Jinja2 escaping doesn't protect against it. This is ensured as there is no part of the web app that will allow a user to input a URL into a jinja template (i.e., not a single Jinja2 template will have *click here* where the *value* is requested from the user).

SQL Injection Mitigation

Flask-Security has some SQL injection countermeasures, but they require the use of SQLAlchemy. Instead, the web app will replace SQL's special characters, validate user inputs, and utilize stored procedures, templates, and prepared statements to execute SQL commands.

Input validation, the usual Achilles heel, will be handled with white listing instead of blacklisting. All inputs will be matched against authorized input syntax, discarding everything that does not qualify. Because input validation alone cannot guarantee app safety, stored procedures will be called to perform SQL functions. Pre-written SQL code will be saved for reuse and control of the database.

Privilege Escalation Mitigation

Since the web app allows both an admin role and a participant role, there is a risk of a participant escalating their privilege to be an admin, which would give them admin rights to the web app. To prevent this, the user role stored in the database shall not be modifiable. Furthermore, the app has only one built-in admin account, and users cannot register for admin accounts.

Compromising Data Integrity

The amount of data that can be modified in the admin account is restricted to not compromise data integrity. Administrators cannot modify students' data through the admin account; they can only view it. That is, they only have read access to students' data. The admin account does have the ability to delete user(s), but they must confirm that they want to delete the user(s) and select from a dropdown list of reasons why they want to delete the accounts. Furthermore, admins have the power to run scripts to specify the number of years for which they want to delete inactive users. The least they can enter is one year, so students who have logged in to their account in less than a year cannot be deleted through this method.

Malicious URLs Mitigation

In order to mitigate against a malicious URL being added to the site, the admin account will not have the capability of modifying a URL that links to external pages. Any link to external pages must be provided to the developers upfront and will be inputted manually to the database.

Data Integrity

Since most of the data inputted to the site are entered by students as young as 6th grade, and since most of the data is reported by students about their own progress, data integrity cannot be guaranteed. The data will be as accurate as the person who added the data. Integrity is guaranteed in the sense that data entered by a participant or an admin will be exactly as entered. Data is updated in real-time and will be available immediately after entering, but may require a page refresh.

Data Retention

While the database will maintain all the data from the date the app is launched, only the data for a date range that an admin specifies will be tracked and displayed on the web app.

Interoperability

The web app will include links to other web pages that are maintained by MESA. While neither the app nor the developers will maintain those pages, they will be accessible through the web app by clicking on a pre-provided hyperlink.