



Eureka Labs

Software Requirements Specification

Version 1.0
April 29th, 2019

Revision History

Date	Version	Description	Author
April 29, 2019	1.0	Revision of previous SRS	Hunter, Dylan, Saby, Antonio, Geordie
May 6, 2019	1.1	Final revision of SRS	Hunter, Dylan, Saby, Antonio, Geordie

Table of Contents

Introduction	5
Purpose	5
Intended Audience and Reading Suggestions	5
Product Scope	5
Definitions, Acronyms, and Abbreviations	5
References	5
Overview	5
Overall Description	5
Product Perspective.	5
User Classes and Characteristics	6
Operating Environment	6
Design and Implementation Constraints	6
Assumptions and Dependencies	7
Specific Requirement	7
Use Cases:	7
UC001 - Content Creator / Educator request account	7
UC002 - System Administrator approves account	9
UC003 - Content Creator logs into the system	10
UC004 - System Administrator logs into the system	11
UC005 - User searches for lab topics	12
UC006 - User sends a question about the lab	13
UC007 - User looks at the top labs	14
UC008 - Content Creator creates lab	15
UC009 - Content Creator edits lab	17
UC010 - Content Creator deletes lab	19
UC011 - Content Creator views lab analytics	21
UC012 - User rates lab	22
UC013 - View analytics for all labs	24
UC014 - System Owner restores application and content	26
UC015 - System Administrator filters list of user accounts	27
UC016 - User Request Password Change	28
UC017 - System notifies user when account is approved	30
UC018 - System notifies admin of account request	31
Data Requirements	32
Logical Data Model	32
Data Dictionary	32
Reports	34

Data Acquisition, Integrity, retention, and disposal	35
External Interface Requirements	35
User Interfaces	35
Software Interfaces	35
Hardware Interfaces	35
Communications Interfaces	35
Quality Attributes	36
Usability	36
Reliability	36
Maintainability	36
Security	36
Safety	36
Performance	36

Software Requirements Specification

1. Introduction

1.1 Purpose

This documents purpose is to describe the System and User requirements of the project.

1.2 Intended Audience and Reading Suggestions

The intended audience of the document is for Faculty, Future Student Developers, and our client Dr. Liran Ma

1.3 Product Scope

For educators who need computer security teaching materials Eureka Labs is an online education portal that provides step by step instructional labs and online assessment. Eureka Labs will be the only resource you need for your computer security lab materials.

1.4 Definitions, Acronyms, and Abbreviations

See the Project Glossary.

1.5 References

MongoDB Documentation - <https://docs.mongodb.com/>

Glossary- [Eureka Glossary](#)

Flask Documentation - <http://flask.pocoo.org/docs/1.0/>

Ubuntu 18.04 Server Guide - <https://help.ubuntu.com/lts/serverguide/index.html>

Bootstrap 4.0 Documentation - <https://getbootstrap.com/docs/4.0/getting-started/introduction/>

NGINX Documentation - <https://nginx.org/en/docs/>

Certbot Documentation - <https://certbot.eff.org/docs/>

1.6 Overview

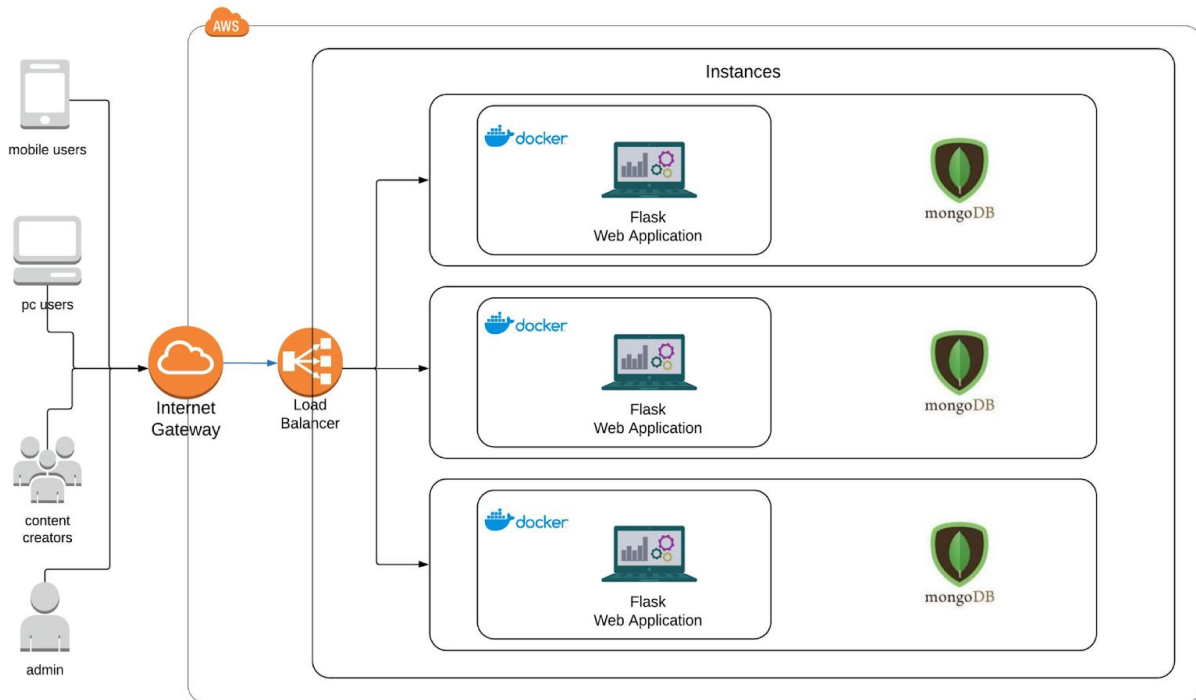
The rest of the document covers the software requirements for Eureka Labs. It explains the the layout of the database and the data being reported. The Document includes specifications on security, usability, etc. This document should be referred to anytime the project is being developed on.

2. Overall Description

2.1 Product Perspective.

Eureka is a new web application that provides a platform for creating and viewing wireless security related labs. The context diagram in Figure C-2 illustrates the external entities and system interfaces for release 1.0. The system is expected to evolve over several releases, ultimately expanding to labs with topics in other areas of Computer Science.

C-2:



2.2 User Classes and Characteristics

User Class	Description
Admin	Anyone who has been assigned a role that gives the full access to labs and users.
Content Creator	Anyone who has been assigned a role that lets them create new labs.
Educator	.Anyone who has been assigned a role that grants them access to lab content such as answer keys.
User	<i>Anyone who is visiting Eureka for purposes other than that of Content Creator, Educator, Admin.</i>

2.3 Operating Environment

OE-1: Eureka shall run on one Ubuntu 18.04 server using NGINX 1.14.0 web server and Flask 1.0.2 web application framework. Eureka shall also run another Ubuntu 18.04 server using MongoDB 4.0.5. The Ubuntu server will be hosted in VMWare.

OE-2: Eureka will be accessible from any device that can render web pages over the internet. This includes all major web browsers like Chrome, Firefox, IE.

2.4 Design and Implementation Constraints

-The Systems code and design must adhere to the flask framework specifications ref:

<http://flask.pocoo.org/docs/0.12/>

-The System shall use MongoDB.

-All HTML code shall conform to the HTML 5.0 standard.

2.5 Assumptions and Dependencies

AS-1: The Web Application will be open to the public, for both viewing labs and requesting accounts.

DE-1: The Database will accommodate all requests.

DE-2: The Admin user will view all incoming requests for accounts.

3. Specific Requirement

3.1 Use Cases:

UC001 - Content Creator / Educator request account

Brief Description:

An Educator would like to upload a lab to the website or use an existing lab for their curriculum. They will need to be verified and given logon credentials.

Primary Actor:

Content Creator / Educator

Level:

Stakeholders and Interests:

System Administrator

Preconditions:

Content Creator / Educator must be on the request page.

Postconditions:

System Administrator is notified of a new request.

Trigger:

Content Creator / Educator request access.

Main Success Scenario:

1. Content Creator / Educator inputs personal information into the system.
2. Content Creator / Educator requests access from the system.
3. The system validates Content Creator / Educator's input.
4. The system shall notify the System Administrator that a new access request has been received.
5. The system shall notify the Content Creator / Educator that the notification was sent.

Extensions:

- 3a. Content Creator /Educator already has access login credentials.

3a1. The system shall alert the Content Creator / Educator that they have an account already.

3b. Content Creator /Educator inputs erroneous data.

3b1. The system shall alert the Content Creator / Educator that they have input erroneous data

3b2. The system shall redirect the Content Creator / Educator to input information again.

Priority:

Low

Secondary Actors:

System Administrator, Application, Database

Special Requirements:

There are no special requirements associated with this use case.

Open Issues:

There are no open issues associated with this use case.

UC002 - System Administrator approves account

Brief Description:

The System Administrator needs to give access to a Educator requesting access to the site.

Primary Actor:

System Administrator

Level:

Stakeholders and Interests:

System Administrator

Preconditions:

Identity of the requestor verified, System Administrator is logged into the system.

Postconditions:

Requestor will be able to gain access to the system.

Trigger:

System Administrator receives notice of new user request.

Main Success Scenario:

1. The System Administrator view the list of pending approvals.
2. The System Administrator selects the account.
3. The System Administrator verifies Educator's information and approves the account.
4. Roles are selected for approved Educator.
5. The Educator is notified about the approval and that they are now able to access the site.

Extensions:

- 1a. There are no pending approvals.
 - 1a1. The system displays a message stating that there are no pending approvals.

Priority:

High

Secondary Actors:

Content Creator

Special Requirements:

There are no special requirements associated with this use case.

Open Issues:

There are no open issues associated with this use case.

UC003 - Content Creator logs into the system

Brief Description:

After, being granted access, the Content Creator logs to the system

Primary Actor:

Content Creator / Educator

Level:

Stakeholders and Interests:

System Administrator and Content Creator

Preconditions:

System Administrator has approved login credentials for the Content Creator.

Postconditions:

Content Creator can access the system.

Trigger:

Content Creator clicks the login button.

Main Success Scenario:

1. The approved user inputs their credentials into the system.
2. The system verifies credential.
3. The system grants access to user.

Extensions:

- 2a. Content Creator inputs incorrect credentials into system.
 - 2a1. The system notifies the Content Creator that their credentials were incorrect.
 - 2a2. The system will allow the creator to re-input their credentials.

Priority:

High

Secondary Actors:

System Administrator

Special Requirements:

There are no special requirements associated with this use case.

Open Issues:

There are no open issues associated with this use case.

UC004 - System Administrator logs into the system

Brief Description:

The System Administrator logs into the system.

Primary Actor:

System Administrator

Level:

Stakeholders and Interests:

System Administrator

Preconditions:

The System Administrator is on the login page.

Postconditions:

The System Administrator is now logged into the system.

Trigger:

System Administrator clicks on the login button.

Main Success Scenario:

1. The system validates the admin's password and logs him into the system.
2. The system displays the admin main form and marks the end of this use case.

Extensions:

- 1a. System Administrator inputs incorrect credentials into system.
 - 1a1. The system notifies the System Administrator that their credentials were incorrect.
 - 1a2. The system will allow the creator to re-input their credentials.

Priority:

High

Secondary Actors:

There are no secondary actors associated with this use case.

Special Requirements:

There are no special requirements associated with this use case.

Open Issues:

There are no open issues associated with this use case.

UC005 - User searches for lab topics

Brief Description:

User can search for existing labs on the system.

Primary Actor:

Student user

Level:

Stakeholders and Interests:

System Administrator and Content Creator

Preconditions:

The user must be able to see the search bar and the search button.

Postconditions:

The user must be able to select the lab page if the search is successful.

Trigger:

The user enters the search keywords and presses search.

Main Success Scenario:

1. The search results with matched keywords and displays the required lab.
2. The user is able to go select the lab and go to the lab page marking the end of use case.

Extensions:

- 1a. Lab not found

1a1. If the keywords do not match with existing labs on the system, the user receives a lab does not exist message.

Priority:

Low

Secondary Actors:

There are no secondary actors associated with this use case.

Special Requirements:

There are no special requirements associated with this use case.

When user starts typing stuff autocomplete options come up.

Open Issues:

There are no open issues associated with this use case.

UC006 - User sends a question about the lab

Brief Description:

Student users can send questions and concerns regarding the lab.

Primary Actor:

Student user

Level:

Stakeholders and Interests:

Content Creator and Student users.

Preconditions:

Labs should have a discussion forum.

Postconditions:

Question should be displayed to the Content Creator and Student users.

Trigger:

Student fills out a question and presses the post button.

Main Success Scenario

1. User successfully submits an appropriate question.
2. The Content Creator receives the question.

Extensions:

- 1a. Failure to post questions
 - 1a1. Student is redirected to the lab view page.
- 2a. Unaccepted questions
 - 2a1. Student is redirected to the lab view page.

Priority:

Low

Secondary Actors:

Application, Database, SendGrid

Special Requirements:

There are no special requirements associated with this use case.

Open Issues:

There are no open issues associated with this use case.

UC007 - User looks at the top labs

Brief Description:

A user without credentials can look for top suggested labs.

Primary Actor:

Student

Level:

Stakeholders and Interests:

Students and Educators

Preconditions:

Student shall be on Websites homepage

Postconditions:

Student indicates a lab they want.

Trigger:

Student indicates that they want to see top labs

Main Success Scenario:

1. The Student indicates that they want to see top labs
2. The system runs a check for top labs based on system criteria.
3. The system presents top labs page to user.
4. The Student indicates a lab they want.

Extensions:

- 1a. The system fails to load top labs page.
 - 1a1. The system shall present error page.
 - 1a2. The system redirects user to home page.
 - 1a3. The user shall return to home page.

Priority:

Low

Secondary Actors:

Application, Database

Special Requirements:

Criteria for lab to be considered a top lab is to be decided.

Open Issues:

There are no open issues associated with this use case.

UC008 - Content Creator creates lab

Brief Description:

A user with a Content Creator profile can create new labs.

Primary Actor:

Content Creator

Level:

Stakeholders and Interests:

Content Creator and System Administrator

Preconditions:

The Content Creator shall be logged in.

Postconditions:

The Content Creator successfully submits a lab.

Trigger:

The Content Creator indicates they want to create a lab

Main Success Scenario:

1. The Content Creator indicates they want to create a lab.
2. The system loads the new labs page.
3. The Content Creator fills out the information required.
4. The Content Creator indicates they are finished filling out the information.
5. The system prompts user to confirm information.
6. The Content Creator confirms the information.
7. The Content Creator submits the lab.
8. The system confirms the lab has been submit.

Extensions:

- 1a. The new labs page doesn't load
 - 1a1. The system loads error page.
 - 1a2. The system redirects Content Creator to home page.
 - 1a3. The Content Creator returns to home page.
- 4a. The Content Creators information is incorrect.
 - 4a1. The Content Creator indicates they want to edit the lab information.
 - 4a2. The system reloads the new labs page, with users lab information.
- 7a. The lab doesn't get loaded.
 - 7a1. The system loads error page.
 - 7a2. The system redirects the Content Creator to new labs page.
 - 7a3. The Content Creator returns to new labs page.

Priority:

Low

Secondary Actors:

Application, Database

Special Requirements:

There are no special requirements associated with this use case.

Open Issues:

There are no open issues associated with this use case.

UC009 - Content Creator edits lab

Brief Description:

A user with a Content Creator profile can view their labs and edit them.

Primary Actor:

Content Creator

Level:

Stakeholders and Interests:

Content Creator and System Administrator

Preconditions:

The Content Creator is logged in.

Postconditions:

The system confirms changes.

Trigger:

The Content Creator indicates they want to make changes to a lab.

Main Success Scenario:

1. The Content Creator indicates they want to view their labs.
2. The system loads users labs page.
3. The Content Creator indicates they want to make changes to a lab.
4. The system loads labs page.
5. The Content Creator makes adjustments.
6. The Content Creator submits updated lab.
7. The system prompts user to confirm changes,
8. The Content Creator confirms changes.
9. The system submits changes.
10. The system confirms changes.

Extensions:

- 2a. The system doesn't load user labs page.
 - 2a1. The system loads error page.
 - 2a2. The system prompts user to home page.
 - 2a3. The Content Creator returns to home page.
- 4a. The system doesn't load labs page.
 - 4a1. The system loads error page.
 - 4a2. The system prompts Content Creator to home page.
 - 4a3. The Content Creator returns to home page.
- 9a. The system fails to submit changes

9a1. The system loads error page.

9a2. The system prompts Content Creator to return to labs page.

9a3. The Content Creator returns to labs page.

Priority:

High

Secondary Actors:

Application, Database

Special Requirements:

There are no special requirements associated with this use case.

Open Issues:

There are no open issues associated with this use case.

UC010 - Content Creator deletes lab

Brief Description:

A Content Creator should be able to delete a lab as it is core to the platform of ever updating security labs. This means that creators will be able to take their lab down if they so choose and update a new revised version.

Primary Actor:

Content Creator

Level:

Stakeholders and Interests:

Content Creator, System Administrator

Preconditions:

There is a lab created by the Content Creator

Postconditions:

The selected lab will no longer exist

Trigger:

Content Creator selects to delete a lab of theirs

Main Success Scenario:

1. System receives delete request
2. Content Creator verifies the delete request
3. System deletes file
4. System returns successfully deleted

Extensions:

- 1a. System is unable to receive the request
 - 1a1. The System will show a sorry can't process request screen
- 2a. System doesn't receive verification
 - 2a1. system shows error screen
- 3a. System can't delete
 - 3a1. System shows error screen
- 4a. System returns unsuccessful in deleting
 - 4a1. System shows error screen

Priority:

High

Secondary Actors:

Database / File Storage, Application

Special Requirements:

Needs to be usable to non technical users.

Open Issues:

There are no open issues associated with this use case.

UC011 - Content Creator views lab analytics

Brief Description:

A Content Creator should be able to view lab analytics because they can then improve based on metrics. You can't track what you don't measure so Content Creator's should be able to see some basic metrics.

Primary Actor:

Content Creator

Level:

Stakeholders and Interests:

Content Creator, System Administrator

Preconditions:

There is a Content Creator account

Postconditions:

The Content Creator has seen the analytics

Trigger:

Content Creator selects to view a labs analytics

Main Success Scenario:

1. Analytics request sent
2. Analytics calculated
3. Analytics shown

Extensions:

- 1a. Application is unable to receive the request
 - 1a1. The Application will show a sorry can't process request screen
- 2a. Analysis can't be generated
 - 2a1. Application will show an error screen
- 3a. Analysis can't be shown
 - 3a1. Error is shown instead

Priority:

High

Secondary Actors:

Application, Database

Special Requirements:

Needs to be usable to non technical users.

Open Issues:

There are no open issues associated with this use case.

UC012 - User rates lab

Brief Description:

The user can select a rating from one to five. The rating of the lab will update after rating has been submitted

Primary Actor:

User

Level:

Stakeholders and Interests:

Content Creator, System Administrator

Preconditions:

The user shall be on the desired lab page.

Postconditions:

The labs rating updates.

Trigger:

The user selects a rating for lab.

Main Success Scenario:

1. The system submits lab rating
2. The system updates the lab rating.
3. The system shall not allow the user to submit rating for the same lab during current session.

Extensions:

- 1a. The system does not submit rating.
 - 1a1. The system shall alert the user to retry submission or contact System Administrator if issue continues.
- 1b. The user did not select a rating.
 - 1b1. The system shall alert the user to select a rating.
 - 1b2. The system shall allow the user to submit after rating has been selected.
- 3a. The user in the same session submits rating for lab that the user already rated.
 - 3a1. The system shall alert the user that a rating has already been submitted.
 - 3a2. The system shall not update the rating.

Priority:

Low

Secondary Actors:

Application, Database

Special Requirements:

User shall only submit one rating.

Open Issues:

There are no open issues associated with this use case.

UC013 - View analytics for all labs

Brief Description:

User with analytics role can view analytics for all labs created by other Content Creators, including their own.

Primary Actor:

Analytics User

Level:

Stakeholders and Interests:

System owner and analytics user

Preconditions:

The analytics user shall be logged in.

Postconditions:

All lab analytical are displayed on one page.

Trigger:

The Analytics User selects analytics page.

Main Success Scenario:

1. Analytics display the rating for all labs.
2. A count of the labs with the most comments is displayed.
3. The Analytics User filters the lab analytics.
4. The average amount of time spent on each lab page is displayed.
5. The analytics displays how many times each lab was downloaded.
6. Pretest and posttest analytics is displayed for each lab.

Extensions:

- 1a. There are no ratings for the labs
 - 1a1. An info message shows that there are no ratings.
- 2a. There are no lab comments in the system.
 - 2a1. An info message shows that there are no comments.
- 3a. There are no analytic to be displayed.
 - 3a1. The filters are disabled and an info message is displayed.
- 4a. There are no analytic for time spent on page.
 - 4a1. An info message shows that there are no analytics available.
- 5a. There has been no downloads yet.
 - 5a1. An info message is displayed to inform the user that there has been no downloads.
- 6a. No pretest and posttest have been submitted.

6a1. An info message is displayed to show that no test have been submitted yet.

Priority:

Low

Secondary Actors:

Application, Analytics Framework, Database

Special Requirements:

Must have the analytics role.

Open Issues:

There are no open issues associated with this use case.

UC014 - System Owner restores application and content

Brief Description:

System creates automated backups of content, database, and application.

Primary Actor:

System Owner

Level:

Stakeholders and Interests:

Preconditions:

The System Owner must be logged into the AWS system.

Postconditions:

The system is recovered or backed up

Trigger:

The System Owner select backup to restore. Backups are automated

Main Success Scenario:

1. The backup creates a clone of the application and all content every week.
2. The System Owner is notified that the content has been backup.
3. The System Owner selects previous backup to restore from.

Extensions:

- 1a. The system runs out of storage
 - 1a1. The system shall notify the system owner about limited resources.
- 2a. The backup doesn't run.
 - 2a1. The System Owner reconfigures the backup system.
- 3a. There are no available backups.
 - 3a1. The System Owner creates a new backup.

Priority:

High

Secondary Actors:

Application, Database, and AWS

Special Requirements:

Easy to maintain storage and restore from a backup instantly.

Open Issues:

There are no open issues associated with this use case.

UC015 - System Administrator filters list of user accounts

Brief Description:

The System Administrator filters a list of users by their approval status or role.

Primary Actor:

System Administrator

Level:

Stakeholders and Interests:

System Administrator

Preconditions:

System Administrator is logged in and on the user page

Postconditions:

List of users is filtered according to the selection

Trigger:

System Administrator select filter and clicks update button

Main Success Scenario:

1. List is filtered by approval status
2. List is filtered by role

Extensions:

- 1a. There are no users with the selected approval status.
 - 1a1. List will return empty with info message.
- 2a. There are no user with the selected role.
 - 2a1. List will return empty with info message.

Priority:

High

Secondary Actors:

Application and Database

Special Requirements:

Ten users per page

Open Issues:

There are no open issues associated with this use case

UC016 - User Request Password Change

Brief Description:

A Content Creator/Educator has forgotten their password and needs to change it to login.

Primary Actor:

Content Creator/Educator

Level:

Stakeholders and Interests:

System Administrator

Preconditions:

A Content Creator / Educator is on the login screen and is unable to login because they do not know their password.

Postconditions:

A Content Creator / Educator is able to login after resetting their password.

Trigger:

Content Creator/Educator indicate they need to reset their password.

Main Success Scenario:

1. The system will prompt Content Creator/Educator to enter their username and email address associated with their account.
2. The system will verify the entered username and email address are correct.
3. The system will send an email to the entered email address with a randomly generated password and send an email to notify the System Administrator that the Content Creator/Educator has requested a new password.
4. Content Creator/Educator will navigate to the login page and submit received credentials.
5. The system will prompt Content Creator/Educator to reset their password upon access to system.

Extensions:

- 2a. Content Creator/Educator enters an email address that does not correspond to their account.
 - 2a1. The system will indicate that the username or email address are not correct.
 - 2a2. The Content Creator/Educator will acknowledge the system indicator.
 - 2a3. The system will prompt Content Creator/Educator to re-enter username and email address.

Priority:

Low

Secondary Actors:

Database

Special Requirements:

Content Creator/Educator must already have an account.

Open Issues:

There are no open issues associated with this use case

UC017 - System notifies user when account is approved

Brief Description:

After a System Administrator approves an account, the user who requested the account will receive a notification with a link to setup their password.

Primary Actor:

System

Level:

Stakeholders and Interests:

Admin, User

Preconditions:

Admin has approved account

Postconditions:

User receives a confirmation email

Trigger:

Account is approved

Main Success Scenario:

1. Admin approves account
2. System finds user in database
3. System sends confirmation email to user
4. User receives a confirmation email

Extensions:

- 2a. System does not find user in database
 - 2a1. System will notify Admin that user was not found.
- 3a. Users email is incorrect
 - 3a1. System will notify Admin that user email is incorrect.

Priority:

Medium

Secondary Actors:

Admin, User

Special Requirements:

User must give correct email

Open Issues:

How to notify User email is incorrect.

UC018 - System notifies admin of account request

Brief Description:

When user requests an account, System Administrators will receive notification about a pending request.

Primary Actor:

System

Level:

Stakeholders and Interests:

Admin, Content Creator and Educator

Preconditions:

A content creator/educator is on the register page and has filled out the register form.

Postconditions:

The request is received by the admin and ready for the admin to process.

Trigger:

System sends an email to the admin and notifies him of a new user request.

Main Success Scenario:

1. System displays the user with the register form.
2. User completes the register form and clicks submit request.
3. Systems records the request and notifies admin that there is a pending request.
4. Admin reads the email and processes the request

Extensions:

1. User's submit request fails
 - 1a. User gets notified that the register failed.
2. User can only proceed the request if provided email is .edu.

Priority:

Medium

Secondary Actors:

Content Creator and Admin

Special Requirements:

There are no special requirements associated with this use case.

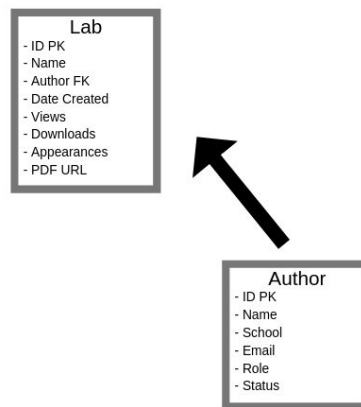
Open Issues:

There are no open issues associated with this use case

4. Data Requirements

The system needs inputs of lab data which consists of pdfs, images, and text. This input is then rendered to the output of a lab page. This lab page displays all of the information in a pleasant manner for students and learners to view and digest the lab. This is the input and output for our system.

4.1 Logical Data Model



4.2 Data Dictionary

Data Element	Description	Data Type	Length	Required
User - id	Unique identifier generated by mongodb	String	50	Yes
User - first name	Users first name	String	50	Yes
User- last name	Users last name	String	50	Yes
User- role	Users row on website	String	15	Yes
User- Active	Users status on website	Boolean	N/A	Yes

Eureka Labs

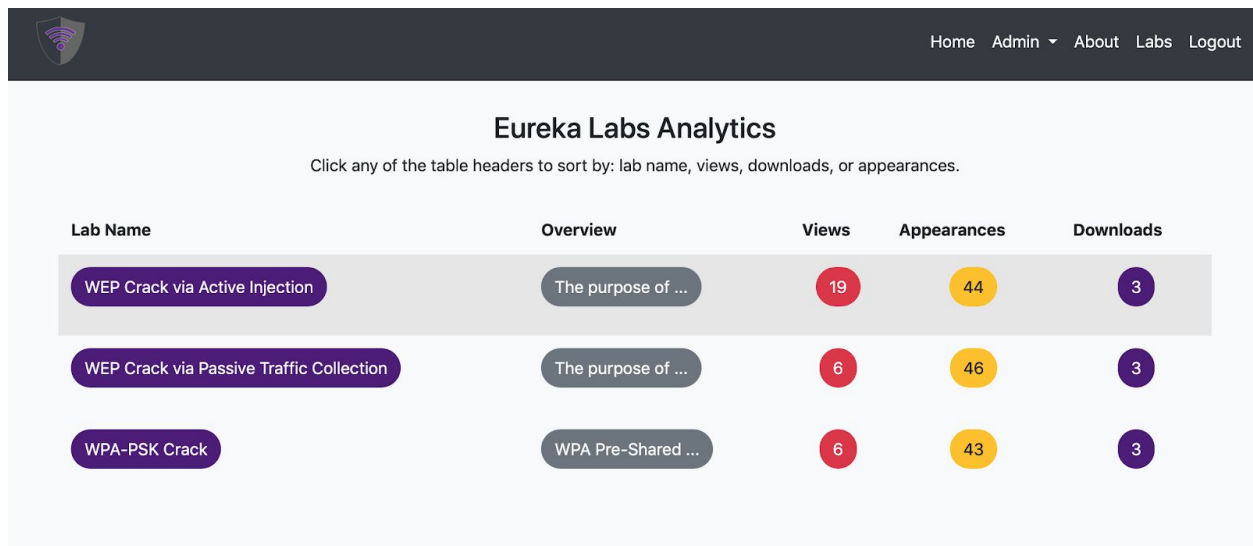
User- Pending	Describes whether User is waiting for approval or not	Boolean	N/A	Yes
User- Date Created	The date users profile was created	Integer	N/A	Yes
User- email	Email registered for user	String	50	Yes
User- username	Username user will use to login	String	50	Yes
User- password	Hashed string provided by user for login	String	50	Yes
User- bio	Information about the user	String	50	No
User- school	School user is associated with	String	50	Yes
User- website	Personal website associated with user	String	50	Yes
Lab - Title	Title of the lab	String	50	Yes
Lab - Overview	Short description of the lab	String	400	Yes
Lab - Author	Reference to the current user creating the lab	Reference (User)	N/A	Yes
Lab - Icon	File name of the icon for the lab	String	N/A	Yes
Lab - Lab Manual	File name of the Lab Manual for the lab	String	N/A	Yes
Lab - Instructor Manual	File name of the Instructor Manual for the lab	String	N/A	No
Lab - Additional	HTML text used to	String	N/A	No

Eureka Labs

	format the style of a page			
Lab - Attachments	List of addition attachment file names	Array	N/A	No
Lab - Views	Number used to keep track of the views	Integer	N/A	N/A
Lab - Downloads	Number used to keep track of the downloads	Integer	N/A	N/A
Lab - Appearances	Number used to keep track of the appearances on a page	Integer	N/A	N/A
Lab - Votes	Number used to keep track of the upvotes for a lab	Integer	N/A	N/A
Lab - Creation Date	Date and time the lab was created	Date Time	N/A	N/A
Lab - Modification Date	Date and time lab was last modified	Date Time	N/A	N/A

4.3 Reports

Our application gathers information regarding labs currently posted. This information includes, numbers of views, downloads, impressions and comments. The application also sends traffic reports to Dr. Ma's google analytics account.



Lab Name	Overview	Views	Appearances	Downloads
WEP Crack via Active Injection	The purpose of ...	19	44	3
WEP Crack via Passive Traffic Collection	The purpose of ...	6	46	3
WPA-PSK Crack	WPA Pre-Shared ...	6	43	3

4.4 Data Acquisition, Integrity, Retention, and Disposal

Our data is acquired through user input as well as automatic collection. We take user input for our lab creation which is done by verified users and the labs last in perpetuity. The data we collect from our analytics is done through automatically processing user clicks and views and is held in perpetuity as well.

5. External Interface Requirements

5.1 User Interfaces

- Users will require access to the internet.
- User will interface the website and web forms using internet browsers.
- The user interface is geared to be minimalistic and achieve maximum functionality.
- The website provides interface for loading and saving files and images.

5.2 Software Interfaces

- The software interfaces will be HTML compatible, by a modern web browser, such as Chrome, Firefox, Safari, Internet Explorer or Edge. The only outside technology we make use of is sendgrid and aws s3. Both of these services are extremely reliable with large corporate backers. Our program runs on python 3 with Flask and MongoDB as the database. We make use of Bootstrap 4 as well as Font-Awesome for front end components.

5.3 Hardware Interfaces

- The user will require access of modern hardware such as desktop computers, laptops, tablets and smartphones.

5.4 Communications Interfaces

- The user-interfaces shall be communicated via hardware or virtual keyboards, and a mouse or trackpad.

6. Quality Attributes

6.1 Usability

- The system shall provide a uniform look with similar buttons and input fields across the site.
- The img tags shall have alt tags to aide with usability for disabled users.
- The system shall load all pages in less than one second so as to not cause severe delays.

6.2 Reliability

- The website shall never crash or hang, other than as the result of an operating system functioning error. The website shall provide graceful degradation in the face of network delays and failures (In the form of writing to the log file and restarting).

6.3 Maintainability

- All code shall be fully documented. Each function shall be commented with pre- and post-conditions. All program files shall include comments concerning authorship and date of last change.
- The code shall be modular to permit future modifications. (Make use of functions and classes)

6.4 Security

- All HTML pages will use HTTPS.
- TLS 1.2 will be the minimum standard for HTTPS
- Communication to the database will only be conducted through the web server.
- All system passwords will be 16 characters containing a minimum of 2 upper case letters, two lower case letters, and two digits.
- User passwords will be hashed and salted.
- Using Flask WTF to prevent cross site scripting.
- Google Recaptcha added to register form to prevent bot request.
- Flask Bleach to remove malicious script tags from HTML editor.

6.5 Safety

- No safety requirements have been identified.

6.6 Performance

- No performance requirements have been identified.