SE I - Team Project

GitHub Updates

Admin Credentials

Test Admin Account:

UN: waltonca@etsu.edu

PW: YMXH@9J!72kM6Em

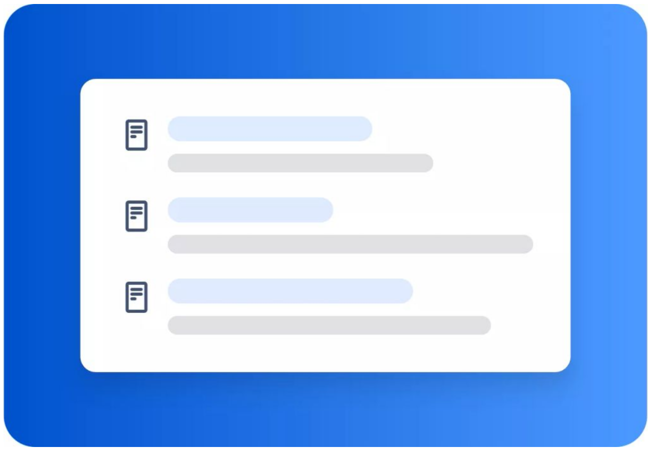
Location of Methods That Access the Database

ScavengeRUs\ScavengeRUs\Services\UserRepository.cs

ScavengeRUs\ScavengeRUs\Services\HuntRepository.cs

GitHub Link

Recently updated



Display a list of recently changed pages

To add the Recently updated element:

1. When editing type /
2. Find Recently updated in the dropdown
3. Select **Insert**

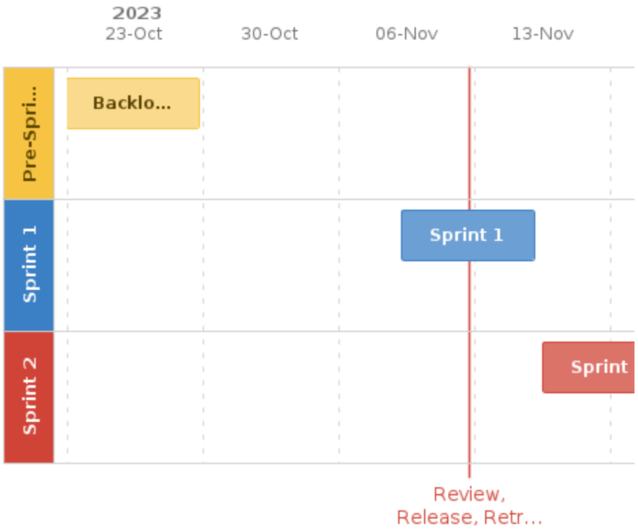
To edit the Recently updated element:

1. Select the placeholder. The floating toolbar appears.
2. Select **Edit**. The right panel opens.
3. Modify the parameters. Your changes are saved as you go.
4. Resume editing the page, and the panel closes.

Roadmap

Adding a roadmap planner

Create simple, visual timelines that are useful for planning projects, software releases and much more.



Roadmaps are made up of:

- **bars** to indicate phases of work.
- **lanes** to differentiate between teams, products or streams.
- **markers** to highlight important dates and milestones.
- a **timeline** showing months or weeks.

You can provide more information about items on your roadmap by linking a bar to a page.

To add the Roadmap planner:

1. When editing type /
2. Find Roadmap planner in the dropdown
3. Select **Insert**

To edit the Roadmap planner:

1. Select the placeholder. The floating toolbar appears.
2. Select **Edit**. The right panel opens.
3. Modify the parameters. Your changes are saved as you go.
4. Resume editing the page, and the panel closes.

READ ME FIRST - Developer Stories

Below, you will find a narrative from each developer in our team on the inherited state of the project when we took it over, our accomplishments, bugs/issues left for your team, and the current state of the project as we ship it to your team.

Developer 1's User Story and Thing I Wish I Knew

1. First thing is that by no means were any of the groups before you actually experts at doing any of this. The more you look at the code the more you are gonna understand it especially if you will use inspect elements on the website to better understand the page you are looking at.
2. The various controllers are where you will find the code that defines what many of the buttons will do. They are named after the folder in which they are going to apply to for example the LocationsController will apply to buttons in the Locations folder.
3. If you are using Github make sure to not include the changes to the database files when you log in to the website and make any changes. Just logging in as an admin adjust a counter so make sure not to include those changes when you go to merge or else you will have conflicts.
4. Keep your changes small for each sprint, don't plan to change the world with your first sprint because you will quickly find out that the various coding techniques used by the previous groups are going to clash and cause you lots of headaches.
5. Depending on what you are trying to change make sure to check the surrounding code because you may find that the code for what you want to do is already created on another page.

Sprint Summaries from my POV

1. Sprint 1: Discovery - We spent the first sprint just trying to make sure that we could all utilize the github and make changes without stepping on each other's toes.
2. Sprint 2: Admin Functionality/Documentation/Standards - We decided the admin did not need to be able to complete the questions so we altered that page to instead be a page where you can add tasks, edit tasks, and delete tasks.
3. Sprint 3: Admin/User Experience - We decided there was no point in having a master task page as eventually it would just be ridiculously long so we corrected links to that page to direct somewhere else and deleted the page.
4. Sprint 4: Shipping the Project - We decided for the final sprint we wanted to finish up what we barely missed in sprint 3 and then also created very clear and informative documentation for the next group.

Here is my discord if you need help but no promises that I will know the answer!! dylan_pogue

Developer 2

The first thing you need to get the project running is Microsoft Visual Studio. Once you have that you just need to open the .sln file to open up the project and then you can run it.

The second and most important thing was setting up source control and familiarizing the team with the workflow of using the source control. We used Github and the first issue we encountered was having conflicting .db files. To solve this we attempted to use the .gitignore but were unsuccessful. The solution we landed on was the following: Database source control solution or simply manually unselecting the .db files when committing code.

The rest of my time on the project has been adding buttons and changing layouts. All of the pages are under Views and then in the relative subfolder. When changing where <a> tags take you, we realized that a controller is used to tell the <a> tags where to take you. These can be found in the Controller folder.

Dev 3

- Inherited status of the project
  - Discovery of the Artifacts and Code
    - We were given a GitHub repository with all of the previous team’s work and documentation. This was decently confusing to use at first, but we eventually began to find what we needed and began breaking it down into reasonably accomplishable tasks.
  - Our questions from the first team’s work
    - The database seems poorly constructed for the purposes of the ScavengeRUS Tier 2 requirements.

- Progress of the Team/Individual Contributions
  - Sprint 1 – Discovery
    - Assisting in setting up the GitHub structure and naming conventions.
  - Sprint 2 - Admin Functionality/Documentation/Standards
    - Privacy page updates
    - Creating a new Add button for View tasks page.
    - Update to Title of Manage tasks page.
    - Adding an Invitation text form to the update hunt page.

- This looks correct but is not connected to update the description text correctly.
  - Sprint 3 - Admin/User Experience
    - Updated title for task page to be dependent upon whether or not you are viewing as a user or an admin.
    - Add button on View task page, renamed to Create Task and made only viewable to admins.
  - Sprint 4 - Shipping the Project

- Conversations and Ideas that were not documented but carry importance to the project.
  - The database seems poorly constructed for the purposes of the ScavengeRUS Tier 2 requirements. In talking with Kinser, it seems that Tier 2 should not require a database at all. And all information could theoretically be held in something like a CSV. None of us were versed in databases enough to fix the issues within 3 weeks, but it was the source of most issues we ran into, and we had to perform workaround fixes leaving the database as is.

Developer 4 Story

Developer 4 Story

Inherited Status of Project

We inherited the project in a mostly functional state, everything was working except for the “Batch Create Users” in the application. Had a bit of trouble finding how to get into the application until we found the previous documentation on GitHub. The previous documentation broke down the coding standards that the previous team had used, admin account credentials, sprint objectives for each sprint, problems faced, and the file structure.

We did have to ask the previous team how they worked with their database instance. We were running into problems on the first sprint with merging since all our database files were different from testing the application. When trying to merge, we had merge conflicts due to the database file. The previous team said that they had a specific database person that would have the master database and would merge. We chose to ignore the database files for our sprints when merging. Specifying the database files in the .gitignore did not work, we had to unselect the database files when merging.

Progress of the Team/Individual Contributions

Sprint 1

The goal of this Sprint was to iron out the source control, coding standards, assess the project where it stood, and get/review the admin account. This is where we found the previous standards used, admin account, and anything else related to the application from the previous Github. We determined our standard of source control that was going to be used this Sprint. Next, I went through the Admin account functionality for this sprint and made sure that CRUD was functional and what was available to the admin. Everything had worked for the admin. Lastly, we demonstrated our understanding of Github by pulling code down, making a small edit, pushing the code, merging our code with each of our edits, and validate that it worked with no errors.

Sprint 2

The goal of this Sprint was to develop the admin privileges and the interface. To start the sprint, I added documentation of all the admin pages to our Confluence site. I also pair programmed with Developer 5 to add a delete and edit button in place of the Complete button for an admin. We removed the functionality for an admin to “Complete” a task, only users can do so. The complete button for users ended up being altered from our work but was still functional. It also did not look as good as before.

Sprint 3

The goal of this sprint was to clean up the task menu for admins. We also wanted to hide the “complete” button when a user correctly solved a task. I pair programmed with Developer 5 to clean the UI for the “complete” button that we had altered last sprint. We also worked on getting that button to disappear from the page once a user has correctly answered a task. These were completed with no issues.

Sprint 4

The goal of this sprint was to fix one of the items from the last sprint that didn’t merge correctly and to create a shippable product for the next team with documentation for your discovery.

Additional Conversations, Ideas, and Notes

Firstly, I feel like the database is a big problem for the project. The way that the application is structured now, it uses a master task list that each hunt pulls from. There are about 8-10 tasks altogether that you can add to a hunt and use. I feel like each hunt needs to have an individual list of tasks unique to that hunt. This makes the hunts unique and could help with some of the functionality of the application. There were tasks that we wanted to do that couldn’t be done because of how the tasks were set up. We didn’t have enough time to restructure the database to remove the master task list and have each hunt have unique tasks.

Overall, our team really focused on admin functionality versus user functionality. We cleaned up how the admin manages tasks and hunts and cleaned up some items for the users. All sprints focused on the tasks and hunts pages as well. We did not work on anything outside of those two items.

Developer 5

Developer 5
Inherited status of the project:

All the code is very overwhelming at first. The pages themselves are in Views. One of our things we had issues with at first was the admin login. It is UN: waltonca@etsu.edu PW: YMXH@9J!72kM6Em We inherited a few problems that we did not solve. I will discuss those at the end.

Progress of the Team/Individual Contributions:

Sprint 1 - Discovery
This was just exploring the code and documentation we inherited. We documented what the last teams standards were and decided upon our own. Each dev made a comment and pushed it so we could practice source control. This is when Developer 2 discovered we cannot merge the database in GitHub.

Sprint 2 - Admin Functionality/Documentation/Standards
Peer coding with Developer 4 to reformat the Edit/Delete buttons on the admin page for a hunt. Previously all tasks for a hunt were displayed in nested divs. We turned them into a table. This caused problems with the 'complete' button on the user's view. We were able to make it functional by adding the script at the bottom of /Views/Hunt/ViewTasks.cshtml. Also removed and changed some buttons at the top of the page.

Sprint 3 - Admin/User Experience
Fixed the ugly complete button from Sprint 2.

Sprint 4 - Shipping the Product
Create and complete documentation.

Conversations and Ideas that were not documented but carry importance to the project:

The data base is a huge problem. Associated tables in the database would fix the problem, but would require rewriting a lot of code too. Currently, if anyone completes a task, it is marked as complete for all players in all hunts.

The images for tasks cannot be edited. When creating a new task, if you upload a new photo, it does not show up. The images that do show up are stored locally in the project in wwwroot/images

Scrum/Agile Definitions

- Agile
    - Philosophy **not** methodology
    - 4 key principles:
        - Individuals and interactions > Process and tools
        - Working software > Comprehensive documentation
        - Customer collaboration > Contract negotiation
        - Responding to change > Following a plan
    - Twelve principles (paraphrased)
        - Customer satisfaction through valuable software is the highest priority
        - Change at any stage of development is welcomed for the customer’s advantage
        - Deliver working software often, the sooner the better
        - Daily collaboration between both business and technical people
        - Motivated people with a happy and trusting environment builds the best projects
        - Face-to-face is king
        - Working software is the main measurement of progress
        - Agile is used to promote sustainable development with a consistent pace
        - Attention to technical and design excellence grows agility
        - K.I.S.S., keep it simple stupid and don’t gold plate code
        - Self-organized teams make the highest quality process of development
        - Team reflection allows for fine-tuning for increased efficiency
- User Story
    - Smallest unit of work in agile framework
    - Informal explanation from a user’s perspective
    - As a [Persona], I want to [Feature/Intent], so that [End goal]
- Sprint Goal
    - Concise statement (1-2 sentences) outlining the objective of the sprint
- Done (to be determined)
- Product backlog
    - Evolving list of requirements based on customer’s needs (Product owner in this case)
    - Prioritized in a list by Product owner
    - User stories
- Grooming
    - Review backlog items to determine the number of items (Starting with the highest priority) to be developed during the next sprint based on the team capacity
    - Focused on:
        - Eliminating user stories that don’t add value
        - Reevaluating user story priorities
        - Common understanding of the user stories
        - Further breaking down user stories to meet sprint timeline and team understanding
- Scrum Roles
    - Product owner
        - Responsible for what will be developed and in what order
        - Manages product backlog

- Orders backlog items
- Ensure backlog visibility
- Writes acceptance tests for each user story

- Scrum Master
  - Responsible for guiding the team
  - Helps team understand the need for backlog items
  - Facilitates Scrum
  - Serve development team
- Development team
  - Responsible for determining how to deliver what the product owner has asked for

Sprint Steps (Forever evolving)

- Sprint planning <= 15min
- Sprint standup <= 15min
- 3 Questions (Document each person’s answers and make them visible for all to see)
    - What have you completed?
    - What do you plan to do?
    - What issues are you running into, if any?
- Backlog grooming <= 30min
- Sprint Review <= 10min
- Sprint Retrospective <= Finish class
    - 3 things done well
    - 3 things to improve upon

Tier 2 - ScavengeRUS

The second tier expands the type of tasks to include Multiple Choice trivia, Single Word answer Trivia, and combinations of these trivia formats with QR codes. Additionally, the tasks can be fixed list, random order (for each access code), or incremental (where tasks must be completed before the next task is shown). The task list presentation, once defined, will be the same for all participants. This tier also introduces the role of System Administrator (Admin). The Admin person has a defined username and password to access the site. At this tier there is only one hunt and it is always active. The Admin is presented with the ability to create/edit/delete tasks for the hunt. Each task has a display label that is shown to the participant, an answer which is free text. They can scan a QR Code and associate it with one or more tasks. (answers are considered a match if they are string matches ignoring case and empty space thus QR Codes will have their associated text stored in association with a task). They can edit how the tasks are presented (fixed list, random, or incremental. They can define the start and end time for the hunt. If a participant tries to access a hunt outside of these times, they get a message letting them know when the hunt will be/was active. The Admin can also edit the introduction text shown at the top of the task list. The Admin is responsible for generating and maintaining a list of access codes for the hunt. Each access code can optionally have either an email or phone number associated with it. The Admin can select from the list of access codes and initiate a notification whereby an email and/or text message is sent out for each selected code.

Key takeaways:

- Only **one hunt active** during this tier and it is always active
- Expanding tasks of scavenger hunt
  - Multiple choice
  - Single word answer trivia
  - Combinations of both
- Tasks can be given in any order
  - Random
  - Fixed
  - Sequential
  - Shared among all participants
- Expanded admin roles
  - User name and password
  - Can create, edit, and delete tasks
    - Edit task order
  - Define the start/end of a hunt
  - Edit the introduction to the tasks
  - Able to generate access code for hunts
    - Assigns access codes to hunt where participants will be notified
      - Email/text notification (both?)
  - Associate QR codes to 1 or more tasks
- QR code
  - String comparison that ignores white spaces and it case insensitive

Tier 2 - Backlog Artifact

Epic: Tier 2 Release

Theme: Admin is able to manage multiple scavenger hunts

User Stories:

1) As an Admin, I want to have secure access to the site so that I can make changes to all aspects of the hunts.
    a) For tier 2, there is only one admin account
    b) The admin account has a username/password
    c) Two factor authentication is optional
2) As an Admin, I want to be able to see all the hunts in the system and their status so that I can manage them effectively
    a) Should be able to sort the list by date created (oldest first or newest first), only active hunts, only expired hunts, only pending hunts, date started (oldest first or newest first)
3) As an Admin, I want to be able to create a new hunt
    a) Set a start date and an end date
    b) Assign a title and theme
    c) Set the invitation text that is included in player invitations
    d) Assign a list of tasks players are to perform
        i) A task consists of a label displayed to players, the Lat/long of the location where they perform the task (must be precise enough to establish if someone is within 50 feet of it), a QR code value which must be human consumable and relevant to the task in question, a QR code which encodes the value
        ii) A single task can be Multiple Choice trivia or Single Word answer Trivia.
            (1) A hunt can include a list of tasks using any combination of these two types
    e) Set the order the list appears to players during the hunt
        i) A list of tasks can be fixed order, random order (for each access code), or incremental (where tasks must be completed before the next task is shown).
            (1) Fixed order is where all tasks are displayed in the same order for all players
            (2) Random order is where each player sees a potentially different order of the same tasks as other players. The order for each player is determined at random at the start of the hunt and remains in that order for the duration of the hunt.
            (3) Incremental is where the tasks are of Fixed Order (see above) but only the first task is displayed. Each subsequent task isn't shown until they complete the displayed one.
        ii) The task list presentation (or order type), once defined, will be the same for all players in the same hunt.
    f) Assign a url for the hunt
4) As an Admin, I want to be able to edit an existing hunt
    a) Can only edit hunts that are Pending or Active
    b) Admin can edit any part of the hunt except the status and creation date
    c) If an admin changes the status to Active, all the players associated with the hunt at that moment will receive a text message telling the hunt has started
5) As an Admin, I want to be able to quickly create accounts for a list of people so that I don't have to manually enter each person
    a) Create a new account using a person's email address and phone number
    b) Assign an access code unique to that player for that hunt (can be duplicated for that player across hunts)
    c) Find an existing account using a person's email address and phone number
    d) Find an existing account using a person's access code
    e) Invite someone to participate in a hunt by sending the hunt's url and a invitation message to their email, then to send their unique access code to their phone via text message
    f) Access codes can be active or disabled or pending invite
6) As an Admin, I want to be able to invite one or more people to play in a specific hunt so that I can control/manage who is involved
7) As an Admin, I want to be able to send notification to a subset of players in a hunt so that I can keep them updated of any changes or alerts
    a) Notifications can be sent as an email or text message (admin chooses)
    b) Admin can select all or a subset of players in the hunt

c) Admin manually enters the notification text

Theme: User is able to play in a scavenger hunt

User Stories:

8) As a Player, I want to be able to join a hunt so I can participate
    - a) All players have an access code that is unique to them and a specific hunt
    - b) The player enters that code on the hunt url page in order to play

9) As a Player, I want to be able to team up with other people so we can participate in the hunt as a group
    - a) More than one person can use the same access code at the same time on different devices in order to “play as a team”
    - b) Player screens need to be refreshed automatically so they can see progress other team members have made

10) As a Player, I want to be able to easily record in the game that I completed one of the tasks in the hunt so that I can get credit for my progress
    - a) For the first release, all tasks are locations the player must go to
    - b) If the player has location enabled, they can hit “I am here” to compare their location to the lat/long associated with the list of locations in the hunt (must be within 50 feet)
    - c) If the player has a camera enabled, they can hit “scan QR Code” to read in the QR code symbol located at the hunt location
    - d) Alternatively, the player can enter the text that appears below the QR code symbol located at the hunt location
    - e) The player should be given a status of their submission
        - i) Valid entry, progress is recorded
            - (1) A valid entry updates their status on the server with a timestamp
        - ii) Invalid entry, no progress recorded
        - iii) Unable to validate, no internet connection

11) As a Player, I want to be able to see how well I’m doing compared to other players so that I can be competitive
    - a) A player that is logged in, can see the list of tasks associated with the hunt
    - b) The list is automatically ordered to show unfinished tasks first

12) As a Player, I want to be able to see, on a map, which task locations I have visited and which ones I haven’t so that I can determine where to go next

13) As a Player, I want to be able to customize my username and profile picture so that I can manage the identity others can see for m

Standards

Source Control

- Main Branch → Dev Branch → Task Branch
- Branch naming convention
  - Start with 2.0.0 since we are on Tier 2 of the project
    - 2.1.0_Dev will be Sprint 1 Dev branch
      - 2.1.0_1 for Task Group 1
      - 2.1.0_2 for Task Group 2
    - 2.1.0_Test
      - 2.1.0_Staging
        - 2.1.0 Production
          - 2.2.0_Dev
  - Tasks will be assigned to individual branches created by each member of the development

Coding Standards

- “Clean” code: Indentation and curly braces
- Naming convention
  - Self-explanatory
    - If there are questions about the name, comment it
- camelCase

Definition of “Done”

- Code that works, meets the requirements, has some error handling, and uses agreed upon documentation
- “Done” Process in Trello:
  - Convert checklist item into a separate card
  - Comment what you have done to it in the new card
    - Gives good documentation for who did what and what was done
  - New card should be placed in “Dev” to allow others to know what task(s) is being worked on
    - Others can work on the branch concurrently
  - New card should be moved to “Code Review” for other peers to validate completion of task
  - Once the “Code Review” is completed, move card to “Testing” for PO to check off
  - ONLY PO can determine the task to be “Done”
  - When PO checks off the task as “Done”, merge to Dev branch

Documentation

- Commenting when needed (usually is)

Github Standard

- Push at least once a week
  - Merge should happen the day before the end of the Sprint at the latest
    - Approved and marked as “Done” by PO before merged

Sprints

Sprint 1 (11/8-11/13)

Goals:

What do we want to learn/do during the first sprint?

- Demonstrate a clear understanding of the current state and progress of the project.
  - Find the coding standards.
  - Create and find documentation related to the project.
    - Consolidate existing documentation into Confluence.
    - Neatly organize the documentation.
    - Have a working understanding of the structure and organization of the documentation.
- Update the existing code base to firmly establish an understanding of source control.
  - Determine the type of source control to be used.
  - Each dev member should successfully:
    - Pull down the current code from the repo.
    - Make edits to the code.
    - Push code back into the repository.
    - Merge code into the working main branch.
    - Validate all code and functionality continues working without breaks or errors.
  - Administrators should have secure access to the site.
    - Create or access admin credentials.
    - View admin page(s).
    - CRUD for a user.

11/1/2023 Grooming Session

- Admin role
    - huntcontroller.cs
    - We have found the admin role but not the login
        - Need to figure out how to make an admin account
        - Austin Hamilton?
        - Potential login, brought to Access Denied Page
            - test@test.com
            - Etsupass12!
    - Database table
        - Roles has admin option
- Two factor
    - Option for it but doesn’t seem to have ever been implemented
- User stories in Trello
    - Will be broken down next class (11/6/2023)

11/6/2023 Sprint Planning

Sprint Planning Goal

- Determine the most important backlog items to build in the next sprint (Starting this Wednesday)
- Involves all 3 parties : Product Owner (PO), Development Team, and Scrum Master (SM)
- Steps:
  - Participants set a **Sprint Goal** to define what the sprint should achieve
  - Participants review the product backlog, specifically over the high-priority items
  - Participants then decide what backlogged-items are <u>realistically achievable</u> during the next sprint while working at a <u>sustainable pace</u>
    - This could also mean breaking down features into smaller and more specific tasks or refining user stories to add more clarity and specificity

Notes before Sprint Planning

- Kinser suggestion: update existing code base with our code to exercise source control
  - Pull a user story (simple) and implement it so we can see if we understand source control together
- A **Sprint Goal** gives the development team an idea of what to work towards if there are tasks that cannot be completed in the given sprint
  - I.E. Guides team towards a common goal
- The team decides who does what tasks
  - Agile principles state that we need to help one another since we share a common goal
- Tasks are **not** picked during Sprint Planning
  - Tasks are picked during **Daily Standup**
  - Tasks are picked one at a time
- Next Daily Standup goal
  - “I got this task done and now I am going to work on this”
- Vertical slicing during Sprint Planning
  - Example: CRUD
    - Create: Task 1
    - Read: Task 2
    - Update: Task 3
    - Delete: Task 4
- On status report for each week
  - What you or the team did with purpose and forethought relating to SE 1 topics
- Sample schedule
  - Monday 11/6: Sprint Planning
  - Wednesday 11/8: Start Sprint 1
  - Monday 11/13: End Sprint 1
    - First half of class
      - Review Sprint 1 (Code should be merged by Sunday 11/12 at 11:59)
      - Release
      - Retrospective Sprint 1
    - Second half of class
      - Planning of Sprint 2
      - Daily Scrum Standup
  - Wednesday 11/15: Start Sprint 2

11/13/2023 End of Sprint 1

First half of class to finish out Sprint 1

- Sprint Review <= 15min
    - Dev team presents the code to the PO (only shows the new and working changes)
    - PO's idea of our progress:
        - Only 2 items moved back into backlog, "Great job" - PO
            - The 2 items were done, just not documented
    - Completed
        - Comments made
        - Privacy page updated
        - Branches made and pulled
- Sprint Retrospective <= Finish half
    - Whiteboard in Sprint 1

Second half of class to start Sprint 2

- Sprint planning <= 15min
    - Sprint Goal: Developing admin privileges and interface
        - As an admin, we should be able to CRUD
        - Admin interface should be clear to follow (redo inconsistent button)
        - Renaming page titles
- Sprint standup <= 15min
    - 3 Questions (Document each person's answers and make them visible for all to see)
        - ~~What have you completed?~~
            - ~~Dylan:~~
            - ~~Will:~~
            - ~~Sydnie:~~
            - ~~Thomas:~~
            - ~~Jacob:~~
        - What do you plan to do?
            - Dylan:
                - 2.2.0_1
                    - Pair programming for admin interfaces with Will
            - Will:
                - 2.2.0_1
                - Deleted old branches to end Sprint 1
                - Making new Dev branch
            - Sydnie:
                - 2.2.0_1
            - Thomas:
                - 2.2.0_3 Admin page screenshots
            - Jacob:
                - 2.2.0_2
                - Privacy page
        - ~~What issues are you running into, if any?~~

  - ~~Dylan:~~
  - ~~Will:~~
  - ~~Sydnie:~~
  - ~~Thomas:~~
  - ~~Jacob:~~
- Backlog grooming <= 30min

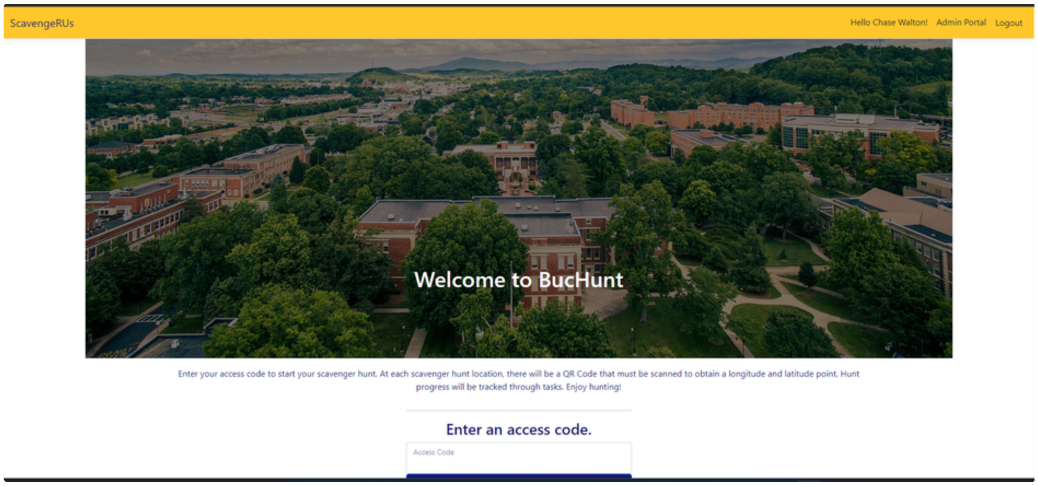
Retrospective - Sprint 1

Documentation - Sprint 1

Listed Below are Admin Pages you can access.

Title Page

1. BucHunt Home Page as Admin



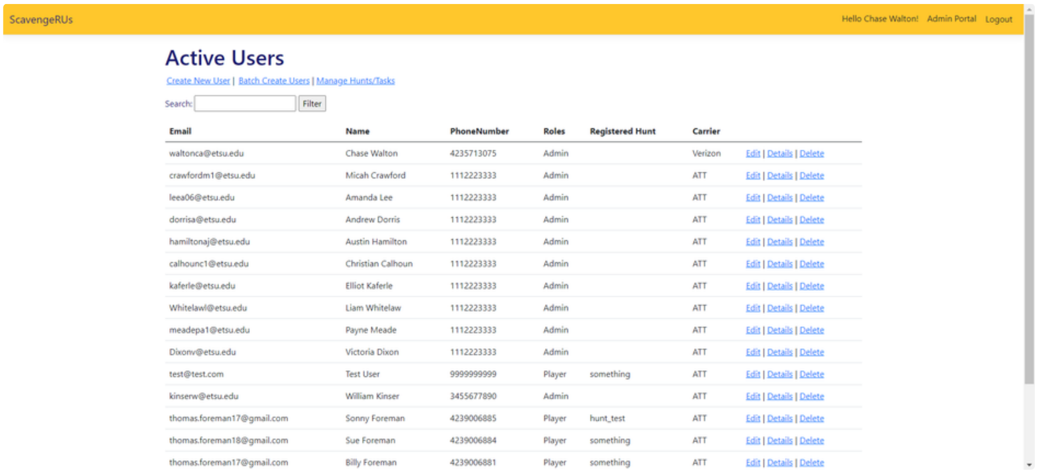
Welcome to BucHunt

Enter your access code to start your scavenger hunt. At each scavenger hunt location, there will be a QR Code that must be scanned to obtain a longitude and latitude point. Hunt progress will be tracked through tasks. Enjoy hunting!

Enter an access code.

Active Users Page

2. First page when pressing "Admin Portal" in the top right.



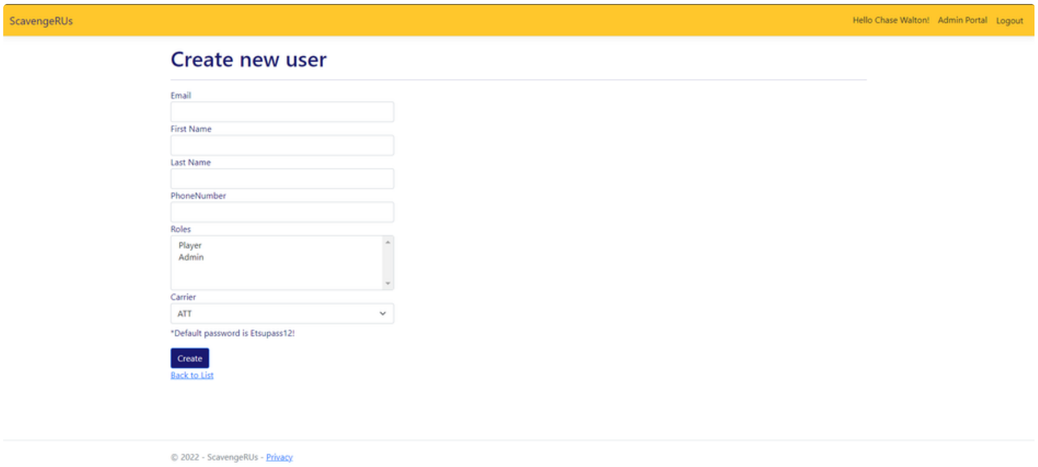
Active Users

Create New User | Batch Create Users | Manage Hunts/Tasks

Search:

Email	Name	PhoneNumber	Roles	Registered Hunt	Carrier	
waltonca@etsu.edu	Chase Walton	4235713075	Admin		Verizon	Edit | Details | Delete
crawfordm1@etsu.edu	Micah Crawford	1112223333	Admin		ATT	Edit | Details | Delete
leea06@etsu.edu	Amanda Lee	1112223333	Admin		ATT	Edit | Details | Delete
dorrisa@etsu.edu	Andrew Dorris	1112223333	Admin		ATT	Edit | Details | Delete
hamiltonaj@etsu.edu	Austin Hamilton	1112223333	Admin		ATT	Edit | Details | Delete
calhounc1@etsu.edu	Christian Calhoun	1112223333	Admin		ATT	Edit | Details | Delete
kaferle@etsu.edu	Elliot Kaferle	1112223333	Admin		ATT	Edit | Details | Delete
Whitelawl@etsu.edu	Liam Whitelaw	1112223333	Admin		ATT	Edit | Details | Delete
meadepa1@etsu.edu	Payne Meade	1112223333	Admin		ATT	Edit | Details | Delete
Dixonv@etsu.edu	Victoria Dixon	1112223333	Admin		ATT	Edit | Details | Delete
test@test.com	Test User	9999999999	Player	something	ATT	Edit | Details | Delete
kinserw@etsu.edu	William Kinser	3455677890	Admin		ATT	Edit | Details | Delete
thomas.foreman17@gmail.com	Sonny Foreman	4239006885	Player	hunt_test	ATT	Edit | Details | Delete
thomas.foreman18@gmail.com	Sue Foreman	4239006884	Player	something	ATT	Edit | Details | Delete
thomas.foreman17@gmail.com	Billy Foreman	4239006881	Player	something	ATT	Edit | Details | Delete

3. Create New User page.



Create new user

Email

First Name

Last Name

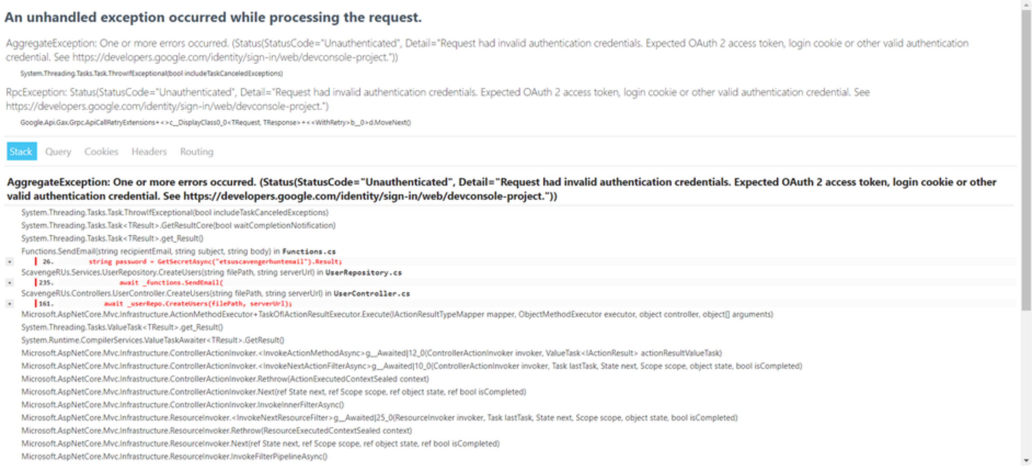
PhoneNumber

Roles
Player
Admin

Carrier
ATT

*Default password is Etsupass12!

4. “Batch Create Users” throws an exception.



An unhandled exception occurred while processing the request.

AggregateException: One or more errors occurred. (Status(StatusCode="Unauthenticated", Detail="Request had invalid authentication credentials. Expected OAuth 2 access token, login cookie or other valid authentication credential. See https://developers.google.com/identity/sign-in/web/devconsole-project."))

System.Threading.Tasks.Task.ThrowIfExceptional(bool includeTaskCanceledExceptions)

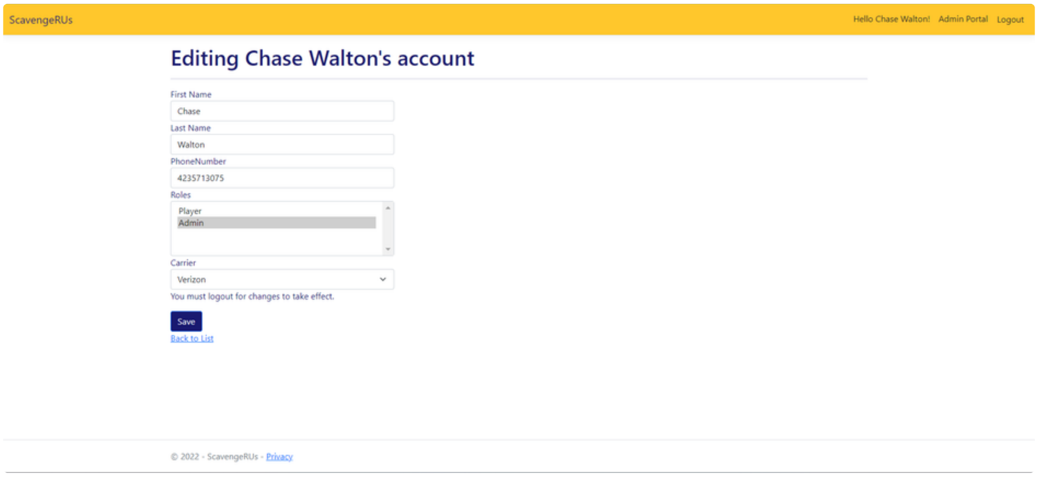
RpcException: Status(StatusCode="Unauthenticated", Detail="Request had invalid authentication credentials. Expected OAuth 2 access token, login cookie or other valid authentication credential. See https://developers.google.com/identity/sign-in/web/devconsole-project.")

Google.Api.Gax.Grpc.ApiCallRetryExtensions+<>c__DisplayClass0_0<TRequest, TResponse>+<<WithRetry>b__0>d.MoveNext()

AggregateException: One or more errors occurred. (Status(StatusCode="Unauthenticated", Detail="Request had invalid authentication credentials. Expected OAuth 2 access token, login cookie or other valid authentication credential. See https://developers.google.com/identity/sign-in/web/devconsole-project."))

```
System.Threading.Tasks.Task.ThrowIfExceptional(bool includeTaskCanceledExceptions)
System.Threading.Tasks.Task<TResult>.GetResultCore(bool waitCompletionNotification)
System.Threading.Tasks.Task<TResult>.get_Result()
Functions.SendEmail(string recipientEmail, string subject, string body) in Functions.cs
+    26.         string password = GetSecretAsync("etsuscavengerhuntemail").Result;
ScavengeRUs.Services.UserRepository.CreateUsers(string filePath, string serverUrl) in UserRepository.cs
+   235.                 await _functions.SendEmail(
ScavengeRUs.Controllers.UserController.CreateUsers(string filePath, string serverUrl) in UserController.cs
+   161.             await _userRepo.CreateUsers(filePath, serverUrl);
Microsoft.AspNetCore.Mvc.Infrastructure.ActionMethodExecutor+TaskOfIActionResultExecutor.Execute(IActionResultTypeMapper mapper, ObjectMethodExecutor executor, object controller, object[] arguments)
System.Threading.Tasks.ValueTask<TResult>.get_Result()
System.Runtime.CompilerServices.ValueTaskAwaiter<TResult>.GetResult()
Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.<InvokeActionMethodAsync>g__Awaited|12_0(ControllerActionInvoker invoker, ValueTask<IActionResult> actionResultValueTask)
Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.<InvokeNextActionFilterAsync>g__Awaited|10_0(ControllerActionInvoker invoker, Task lastTask, State next, Scope scope, object state, bool isCompleted)
Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.Rethrow(ActionExecutedContextSealed context)
Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.Next(ref State next, ref Scope scope, ref object state, ref bool isCompleted)
Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.InvokeInnerFilterAsync()
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeNextResourceFilter>g__Awaited|25_0(ResourceInvoker invoker, Task lastTask, State next, Scope scope, object state, bool isCompleted)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.Rethrow(ResourceExecutedContextSealed context)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.Next(ref State next, ref Scope scope, ref object state, ref bool isCompleted)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.InvokeFilterPipelineAsync()
```

5. Clicking the “Edit” button on the active users



Editing Chase Walton's account

First Name
Chase

Last Name
Walton

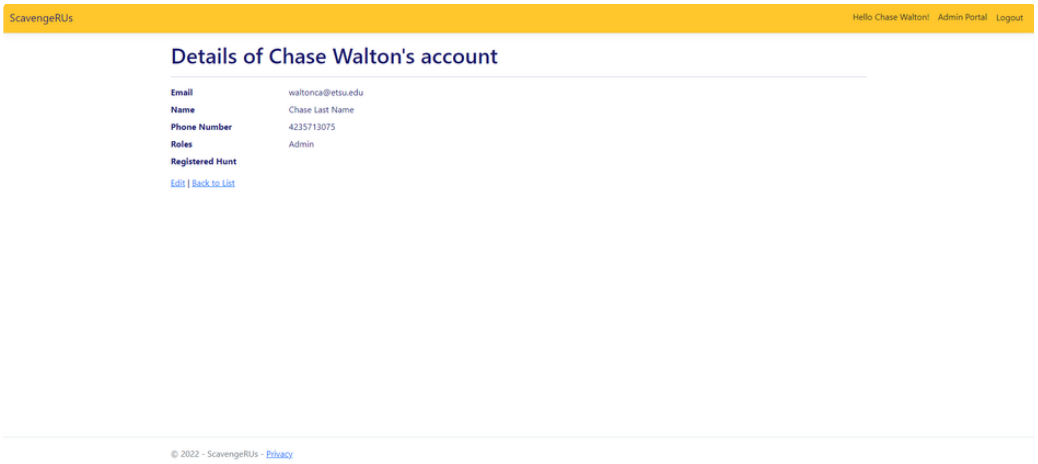
PhoneNumber
4235713075

Roles
Player
Admin

Carrier
Verizon

You must logout for changes to take effect.

6. Clicking the “Details” button on an active user. NOTE: Pressing “Edit on this page will take you to the Edit page for the user.



ScavengeRUs

Hello Chase Walton! Admin Portal Logout

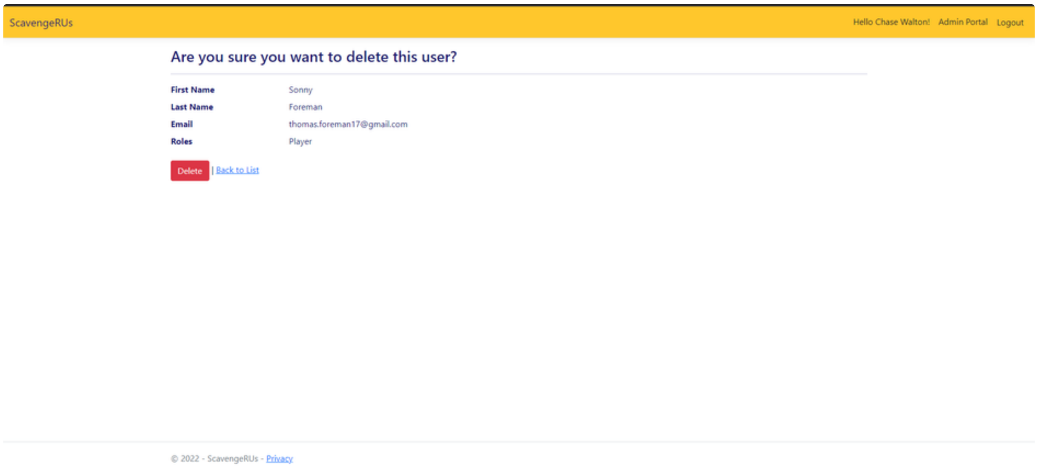
Details of Chase Walton's account

Email	waltonca@etsu.edu
Name	Chase Last Name
Phone Number	4235713075
Roles	Admin
Registered Hunt	

Edit | Back to List

© 2022 - ScavengeRUs - Privacy

7. Clicking the “Delete” button will take you to a confirmation page with user details.



ScavengeRUs

Hello Chase Walton! Admin Portal Logout

Are you sure you want to delete this user?

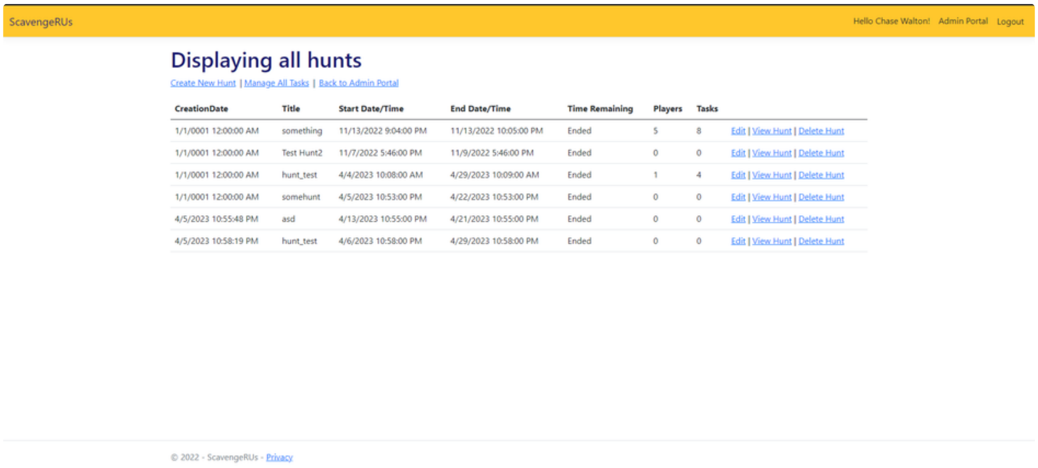
First Name	Sonny
Last Name	Foreman
Email	thomas.foreman17@gmail.com
Roles	Player

Delete | Back to List

© 2022 - ScavengeRUs - Privacy

Hunt Management Page

8. Clicking the “Manage Hunt/Tasks” button will take you to the “Displaying all hunts” section



ScavengeRUs

Hello Chase Walton! Admin Portal Logout

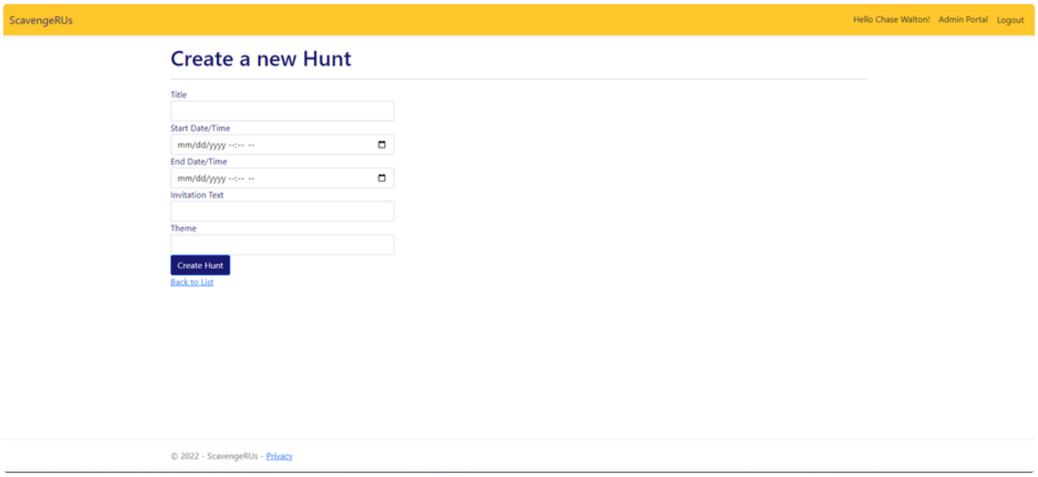
Displaying all hunts

Create New Hunt | Manage All Tasks | Back to Admin Portal

CreationDate	Title	Start Date/Time	End Date/Time	Time Remaining	Players	Tasks	
1/1/0001 12:00:00 AM	something	11/13/2022 9:04:00 PM	11/13/2022 10:05:00 PM	Ended	5	8	Edit | View Hunt | Delete Hunt
1/1/0001 12:00:00 AM	Test Hunt2	11/7/2022 5:46:00 PM	11/9/2022 5:46:00 PM	Ended	0	0	Edit | View Hunt | Delete Hunt
1/1/0001 12:00:00 AM	hunt_test	4/4/2023 10:08:00 AM	4/29/2023 10:09:00 AM	Ended	1	4	Edit | View Hunt | Delete Hunt
1/1/0001 12:00:00 AM	somehunt	4/5/2023 10:53:00 PM	4/22/2023 10:53:00 PM	Ended	0	0	Edit | View Hunt | Delete Hunt
4/5/2023 10:55:48 PM	asd	4/13/2023 10:55:00 PM	4/21/2023 10:55:00 PM	Ended	0	0	Edit | View Hunt | Delete Hunt
4/5/2023 10:58:19 PM	hunt_test	4/6/2023 10:58:00 PM	4/29/2023 10:58:00 PM	Ended	0	0	Edit | View Hunt | Delete Hunt

© 2022 - ScavengeRUs - Privacy

9. Clicking the “Create New Hunt” button will take you to the Create Hunt page.



Create a new Hunt

Title

Start Date/Time

mm/dd/yyyy --:-- --

End Date/Time

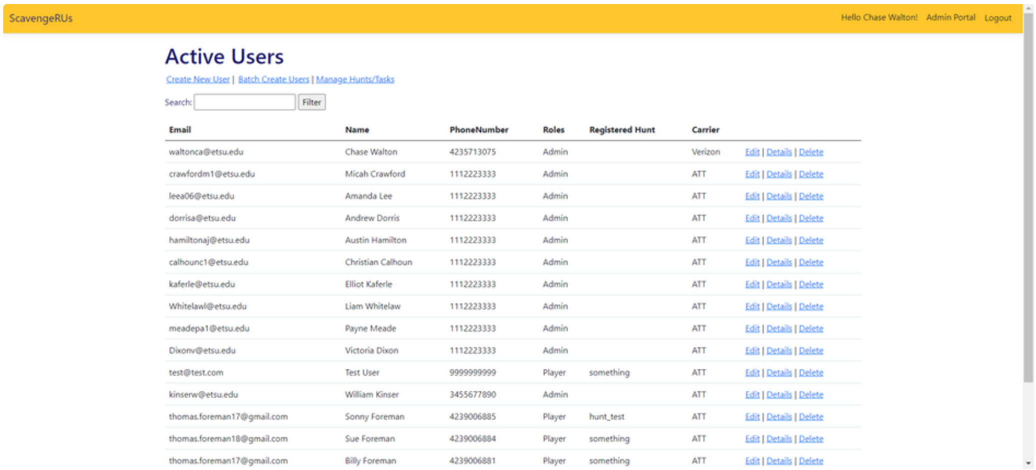
mm/dd/yyyy --:-- --

Invitation Text

Theme

© 2022 - ScavengeRUs - Privacy

10. Clicking the “Back to Admin Portal” button will take you back to the Admin Home Page.



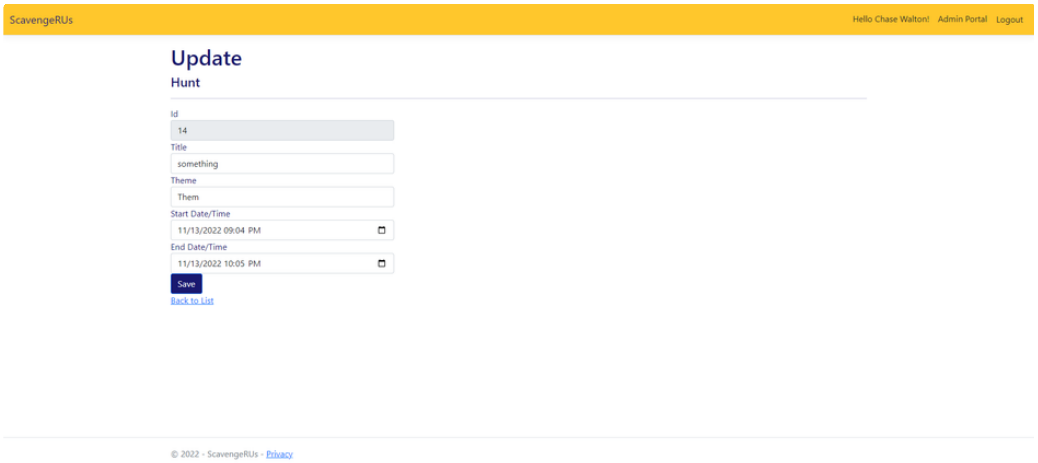
Active Users

Create New User | Batch Create Users | Manage Hunts/Tasks

Search: Filter

Email	Name	PhoneNumber	Roles	Registered Hunt	Carrier	
waltonca@etsu.edu	Chase Walton	4235713075	Admin		Verizon	Edit | Details | Delete
crawfordm1@etsu.edu	Micah Crawford	1112223333	Admin		ATT	Edit | Details | Delete
leea06@etsu.edu	Amanda Lee	1112223333	Admin		ATT	Edit | Details | Delete
dorrisa@etsu.edu	Andrew Dorris	1112223333	Admin		ATT	Edit | Details | Delete
hamiltonaj@etsu.edu	Austin Hamilton	1112223333	Admin		ATT	Edit | Details | Delete
calhounc1@etsu.edu	Christian Calhoun	1112223333	Admin		ATT	Edit | Details | Delete
kaferle@etsu.edu	Elliot Kaferle	1112223333	Admin		ATT	Edit | Details | Delete
Whitelawl@etsu.edu	Liam Whitelaw	1112223333	Admin		ATT	Edit | Details | Delete
meadepa1@etsu.edu	Payne Meade	1112223333	Admin		ATT	Edit | Details | Delete
Dixonv@etsu.edu	Victoria Dixon	1112223333	Admin		ATT	Edit | Details | Delete
test@test.com	Test User	9999999999	Player	something	ATT	Edit | Details | Delete
kinserw@etsu.edu	William Kinser	3455677890	Admin		ATT	Edit | Details | Delete
thomas.foreman17@gmail.com	Sonny Foreman	4239006885	Player	hunt_test	ATT	Edit | Details | Delete
thomas.foreman18@gmail.com	Sue Foreman	4239006884	Player	something	ATT	Edit | Details | Delete
thomas.foreman17@gmail.com	Billy Foreman	4239006881	Player	something	ATT	Edit | Details | Delete

11. Clicking the “Edit” button on a hunt will display this.



Update

Hunt

Id

14

Title

something

Theme

Them

Start Date/Time

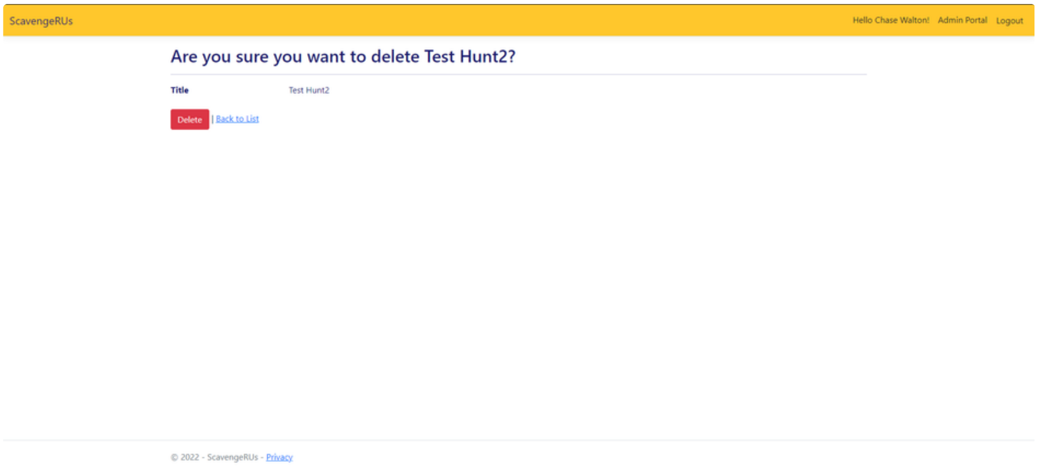
11/13/2022 09:04 PM

End Date/Time

11/13/2022 10:05 PM

© 2022 - ScavengeRUs - Privacy

11. Clicking the “Delete Hunt” button will take you to the confirmation page

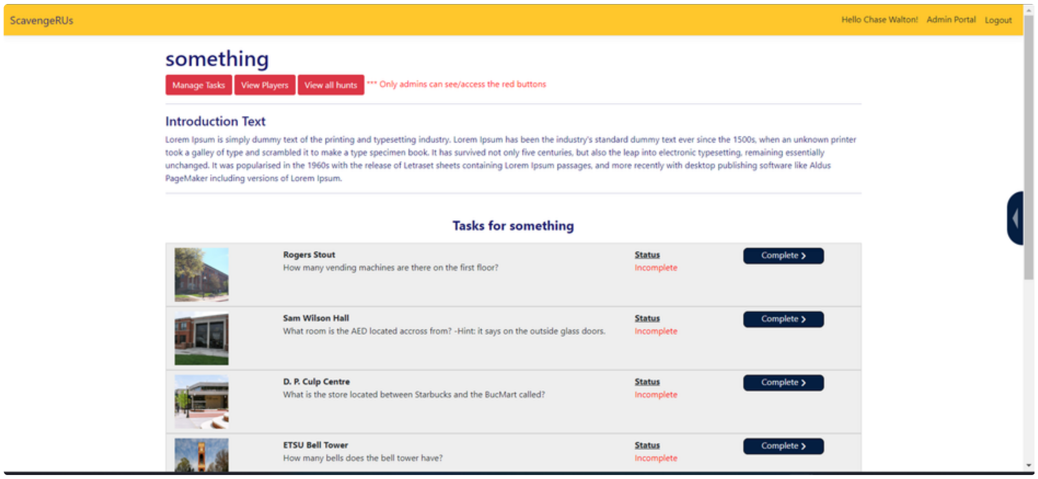


Are you sure you want to delete Test Hunt2?

Title Test Hunt2

© 2022 - ScavengeRUs - Privacy

12. Clicking the “View Hunt” button will take you to a Hunt Page that a user would see. Admin buttons are in red.



something

*** Only admins can see/access the red buttons

Introduction Text

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Tasks for something

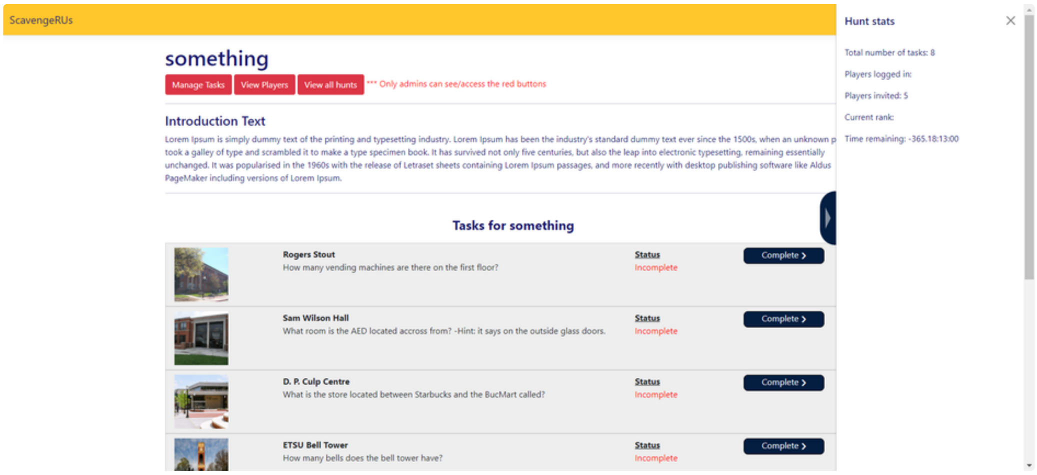
Rogers Stout
How many vending machines are there on the first floor?
Status
Incomplete

Sam Wilson Hall
What room is the AED located accross from? -Hint: it says on the outside glass doors.
Status
Incomplete

D. P. Culp Centre
What is the store located between Starbucks and the BucMart called?
Status
Incomplete

ETSU Bell Tower
How many bells does the bell tower have?
Status
Incomplete

Side Panel:



something

*** Only admins can see/access the red buttons

Introduction Text

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown p took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Tasks for something

Rogers Stout
How many vending machines are there on the first floor?
Status
Incomplete

Sam Wilson Hall
What room is the AED located accross from? -Hint: it says on the outside glass doors.
Status
Incomplete

D. P. Culp Centre
What is the store located between Starbucks and the BucMart called?
Status
Incomplete

ETSU Bell Tower
How many bells does the bell tower have?
Status
Incomplete

Hunt stats

Total number of tasks: 8

Players logged in:

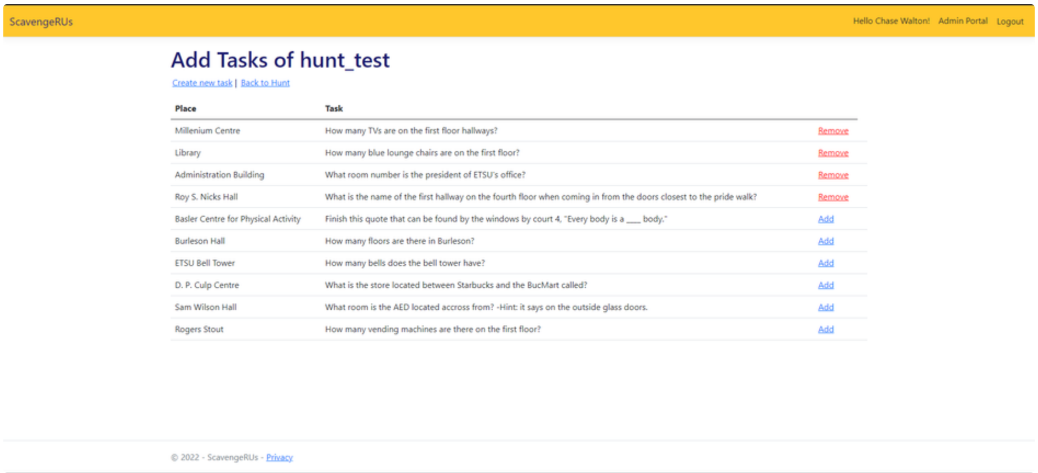
Players invited: 5

Current rank:

Time remaining: -365.18:13:00

Managing Task of Hunt

13. Clicking the “Manage Tasks” will take you to this page



Add Tasks of hunt_test

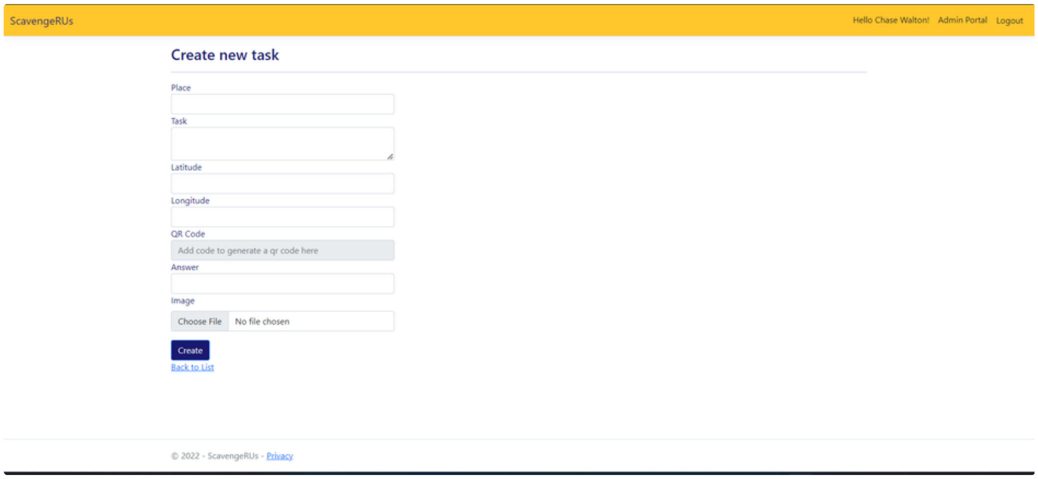
Create new task | Back to Hunt

Place	Task	
Millenium Centre	How many TVs are on the first floor hallways?	Remove
Library	How many blue lounge chairs are on the first floor?	Remove
Administration Building	What room number is the president of ETSU's office?	Remove
Roy S. Nicks Hall	What is the name of the first hallway on the fourth floor when coming in from the doors closest to the pride walk?	Remove
Basler Centre for Physical Activity	Finish this quote that can be found by the windows by court 4, "Every body is a ___ body."	Add
Burleson Hall	How many floors are there in Burleson?	Add
ETSU Bell Tower	How many bells does the bell tower have?	Add
D. P. Culp Centre	What is the store located between Starbucks and the BucMart called?	Add
Sam Wilson Hall	What room is the AED located accross from? -Hint: it says on the outside glass doors.	Add
Rogers Stout	How many vending machines are there on the first floor?	Add

© 2022 - ScavengeRUs - Privacy

NOTE: Clicking “Add” or “Remove” will not take you to another page. Functionality is on this page.

14. Clicking “Create new task” button will take you to Create Task page



Create new task

Place

Task

Latitude

Longitude

QR Code

Add code to generate a qr code here

Answer

Image

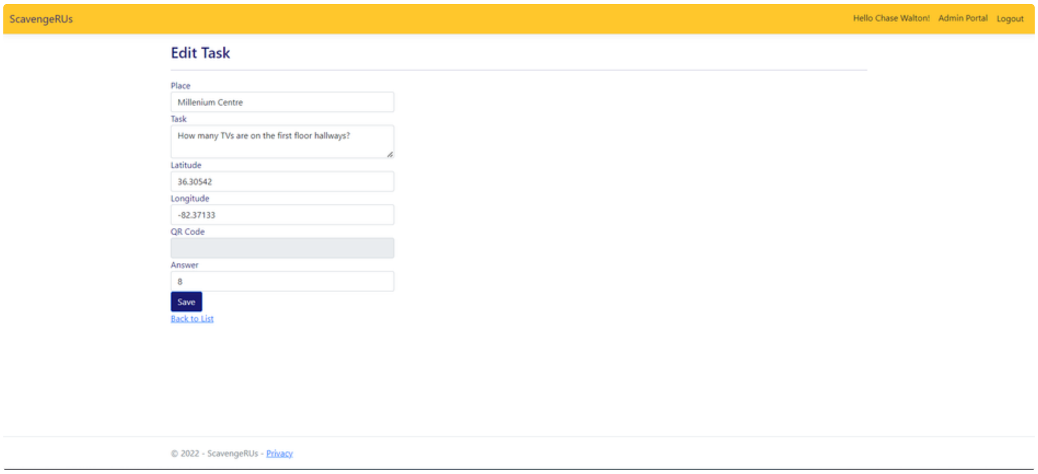
No file chosen

© 2022 - ScavengeRUs - Privacy

Going back to page 8.

Managing All Tasks

15. Clicking the “Edit” button on a task will display the edit page for a task.



Edit Task

Place: Millenium Centre

Task: How many TVs are on the first floor hallways?

Latitude: 36.30542

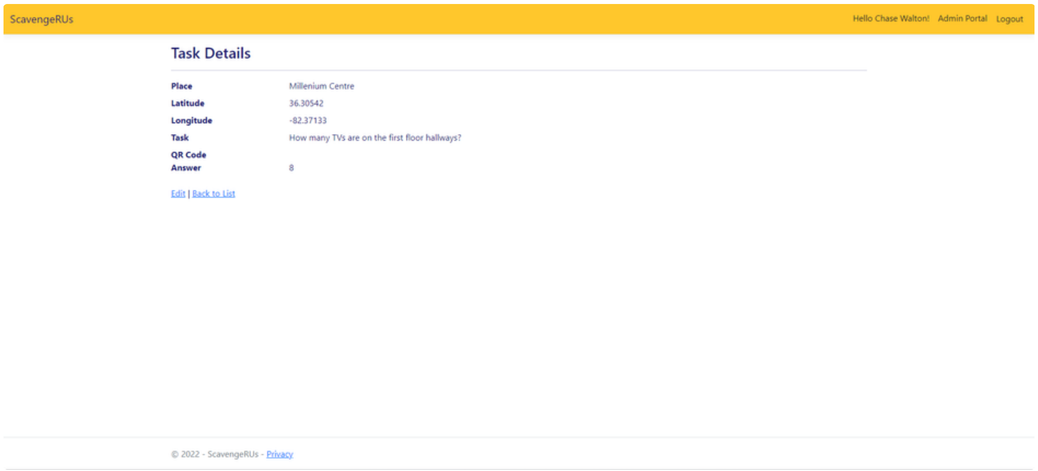
Longitude: -82.37133

QR Code:

Answer: 8

© 2022 - ScavengeRUs - Privacy

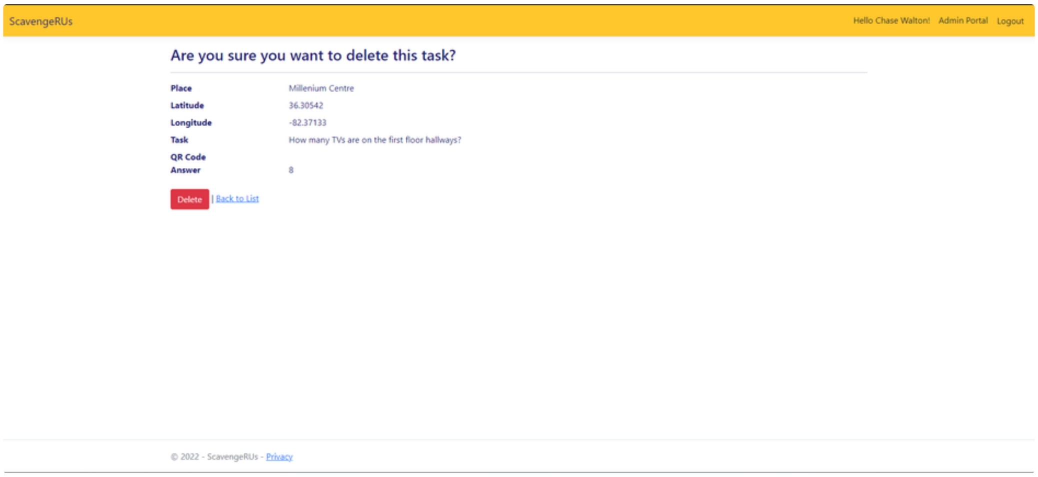
16. Clicking the details button on a task will display the details of a task



Task Details

Place	Millenium Centre
Latitude	36.30542
Longitude	-82.37133
Task	How many TVs are on the first floor hallways?
QR Code	
Answer	8

17. Clicking the “Delete” button on a task will take you to the Delete Confirmation page

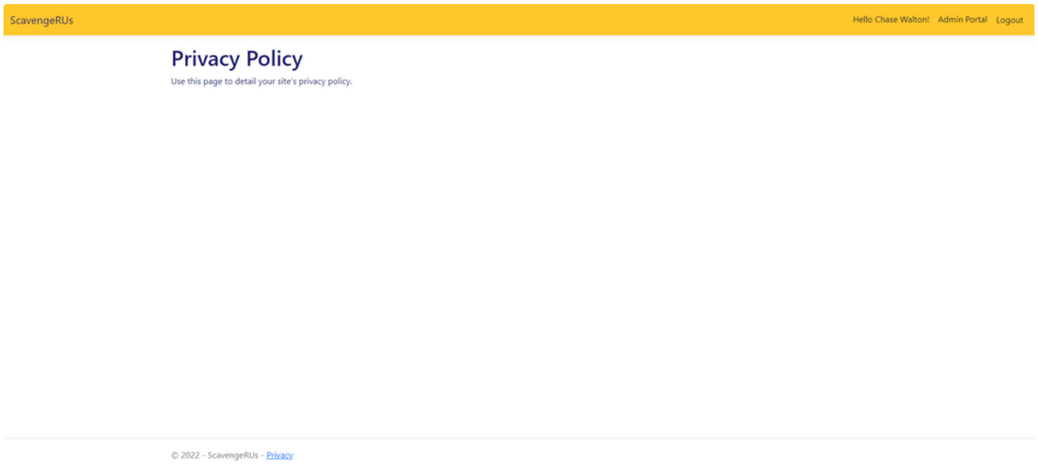


Are you sure you want to delete this task?

Place	Millenium Centre
Latitude	36.30542
Longitude	-82.37133
Task	How many TVs are on the first floor hallways?
QR Code	
Answer	8

Extra

18. Clicking the “Privacy” button at the bottom of any page will display this for admins



Privacy Policy

Use this page to detail your site's privacy policy.

Sprint 2 (11/15 - 11/20)

11/15/2023 Sprint 2 In Class Meeting

- Sprint planning <= 15min
    - Sprint Goal: Developing admin privileges and interface
        - As an admin, we should be able to CRUD
        - Admin interface should be clear to follow (redo inconsistent button)
        - Renaming page titles
- Sprint standup <= 15min
    - 3 Questions (Document each person's answers and make them visible for all to see)
        - What have you completed?
            - Dylan:
                - Review over CRUD
            - ~~Will:~~ Out
            - Sydnie:
                - Nothing yet
            - Thomas:
                - Documentation for Admin pages
            - Jacob:
                - Privacy policy changes
        - What do you plan to do?
            - Dylan:
                - "Complete" button → "Edit" button
            - ~~Will:~~ Out
            - Sydnie:
                - Add "Delete" button
                - "Edit Hunt" button
            - Thomas:
                - Add "Delete" button
                - "Edit Hunt" button
            - Jacob:
                - Add task at the bottom of the lists of tasks
        - What issues are you running into, if any?
            - None yet
- Backlog grooming (Optional) <= 30min

11/20/2023 - End of Sprint 2

First half of class to finish out Sprint 2

- Sprint Review <= 15min
  - Dev team presents the code to the PO (only shows the new and working changes)
  - PO’s idea of our progress: A good direction and good to release
  - Completed
    - Changed the “complete” button
      - Now it is “Edit task” and “Delete task” for admins
        - Edit task button does work
          - Does not go back to the correct page
        - Delete task button does work
    - Privacy page updated and viewable for everyone
    - Edit hunt button
      - Just front end
- Sprint Retrospective <= Finish half
  - Whiteboard in Sprint 2

Second half of class to start Sprint 3

- Sprint planning <= 15min
  - Sprint Goal: Cleaning up task menus for admins to circumvent master task list. Hide the “complete” button once a task is completed.
- Sprint standup <= 15min
  - 3 Questions (Document each person’s answers and make them visible for all to see)
    - What have you completed?
      - Dylan:
      - Will:
      - Sydnie:
      - Thomas:
      - Jacob:
    - What do you plan to do?
      - Dylan:
      - Will:
      - Sydnie:
      - Thomas:
      - Jacob:
    - What issues are you running into, if any?
      - Dylan:
      - Will:
      - Sydnie:
      - Thomas:
      - Jacob:
- Backlog grooming <= 30min

Retrospective - Sprint 2

Sprint 2 Review

Manage Task Page - Admin View

Buttons

The complete Button is now the edit task and delete task button.

- Edit Task:
    - The button links back to the old page when you leave the edit task page.
- Delete Task:
    - It has a very nice confirmation page that asks the user if they want to delete the task.
    - Deletes the task but links back to the old page. It should come back to the manage task page after deleting.
- Edit Hunt:
    - The UI is nice; backend functionality will need to be added later.
        - Added invitation text for the hunt.
        - Title
        - Theme
        - Start Date/Time
        - End Date/Time

Create a Task

- The Create button links to the page we want to remove.
- There is an additional create task button that you have to hit on a second page; this is confusing to an admin.
- The page links back to the old page we want to remove once a task is created.
- The create task button is visible to users of a hunt; it should be admin view only.

User Task View

- The user task view shows a complete button next to each task for a user to attempt to complete the task.
    - The button still needs attention for the UI; it is wonky.
- The page title should display “Tasks for [HUNT NAME].”
    - Will be added to the product backlog.

Sprint 3 (11/27-12/4)

11/27/2023 Sprint 3 In Class Meeting

- Sprint planning (Just to go back over it after coming back from break) <= 15min
  - Sprint Goal: Cleaning up task menus for admins to circumvent master task list. Hide the “complete” button once a task is completed.
- Sprint standup <= 15min
  - 3 Questions (Document each person’s answers and make them visible for all to see)
    - What have you completed?
      - Thanksgiving break
    - What do you plan to do?
      - Dylan: 2.3.0_1, deleting master task list page
      - Will: 2.3.0_1, deleting master task list page
      - Sydnie: 2.3.0_5, clean up the UI on the complete button for user view on the task page
      - Thomas: 2.3.0_5, clean up the UI on the complete button for user view on the task page
      - Jacob: 2.3.0_4, make add task button only viewable to admin and then 2.3.0_3, title for the task page view for users if it can be done
    - What issues are you running into, if any?
      - Short timeline
      - Hunt ID is in a for loop
        - This is messing with circumventing the master list page
- Backlog grooming <= 30min

11/29/2023 - End of Sprint 3

First half of class to finish out Sprint 3

- Sprint Review <= 15min
  - Dev team presents the code to the PO (only shows the new and working changes)
  - PO’s idea of our progress:
  - Completed tasks:
    -
- Sprint Retrospective <= Finish half
  - Whiteboard in Sprint 3

Second half of class to start Sprint 4 (Final sprint)

- Sprint planning <= 15min
  - Sprint Goal:
- Sprint standup <= 15min
  - 3 Questions (Document each person’s answers and make them visible for all to see)
    - What have you completed?
      - Dylan:
      - Will:
      - Sydnie:
      - Thomas:
      - Jacob:
    - What do you plan to do?
      - Dylan:
      - Will:
      - Sydnie:
      - Thomas:
      - Jacob:
    - What issues are you running into, if any?
      - Dylan:
      - Will:
      - Sydnie:
      - Thomas:
      - Jacob:
- Backlog grooming <= 30min

Retrospective - Sprint 3

Sprint 3 Review

2.3.0_1 Delete the master task list page.

Issues with merging: the page wasn’t hidden and deleted after the merge.

The save button under the edit task page goes back to the master task list page.

Moving back to the Backlog.

2.3.0_2 Complete Button Hide after Task Completed

Works perfectly!

2.3.0_3 Title for the Task page view for users.

Looks great!

The user site says

2.3.0_4 Make Add Task button only viewable to admin

Complete! Not visible to users and visible to admins.

2.3.0_5 Clean up the UI on the Complete button for user view on the task page.

Looks beautiful!

Sprint 4 (11/29 - 12/6)

Sprint Goal: Fix the last item not completed from sprint 3 and create a shippable product for the next team that includes artifacts for their discovery.

11/29/2023 Daily Standup Meeting

Accomplished

- Review of Sprint 3
- Merging of Sprint 3 code
- Release of Sprint 3
- Retrospective for Sprint 3
- Made the complete button on the task page hide for a user after they complete a task.
- Changed the title of the user task page.
- Changed all of the links (except for two) to avoid the old master task page.
- The create task button was hidden for the user view on the task page.

Obstacles

- The merge didn’t go exactly as planned:
    - The merge caused some of the changes in branch 2.3.0_1 to not appear in the 2.3.0_dev branch once merged.
- Some buttons were discovered at the last minute that took us to the master task page, which we didn’t want to exist.

Plans

- Correct the mistakes from sprint 3.
- Write developer stories for the next team.
- Create shippable artifacts.

12/4/2023 Sprint 4 In Class Meeting

- Sprint planning (Just to go back over it after coming back from break) <= 15min
  - Sprint Goal: Wrapping up. Developers are writing up user stories for documentation for the next semester.
- Sprint standup <= 15min
  - 3 Questions (Document each person's answers and make them visible for all to see)
    - What have you completed?
      - Dylan: Buttons are fixed and master list page is now gone
      - Will: Nothing
      - Sydnie: Nothing
      - Thomas: Developer stories
      - Jacob: Nothing
    - What do you plan to do?
      - Dylan: Write up user stories
      - Will: Write up user stories
      - Sydnie: Write up user stories
      - Thomas: Write up user stories
      - Jacob: Write up user stories
    - What issues are you running into, if any?
      - Merging conflicts
- Backlog grooming <= 30min

12/6/2023 End of Sprint 4

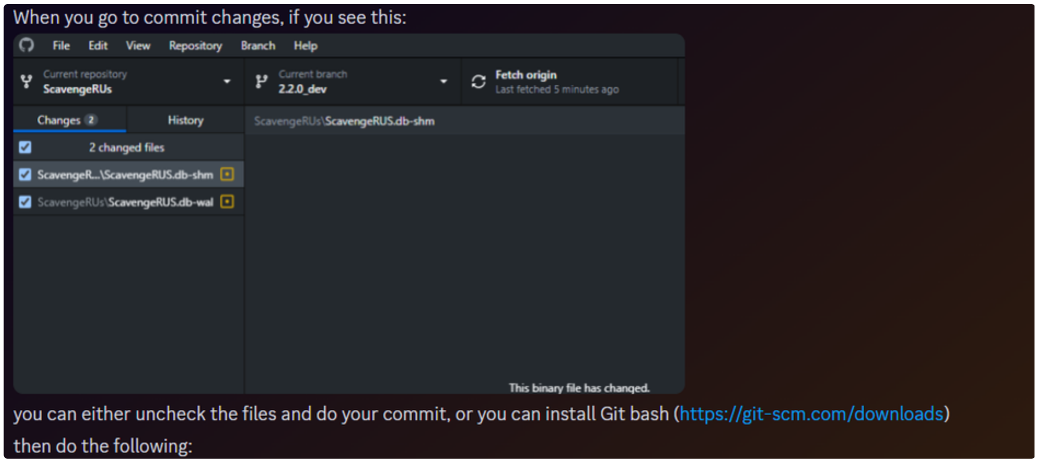
- Sprint standup <= 15min
    - 3 Questions (Document each person’s answers and make them visible for all to see)
        - What have you completed?
            - Dylan: User story
            - Will: User story
            - Sydnie: User story
            - Thomas: User story
            - Jacob: User story
- Sprint Review <= 15min
    - Dev team presents the code to the PO (only shows the new and working changes)
    - PO’s idea of our progress: The user stories meet the criteria perfectly
    - Completed tasks:
        - All 5 developers have complete their user stories

Sprint 4 Review

Bug Page

- Tasks in hunts will be marked as complete or incomplete across all hunts if they share the same tasks
- When editing a task, the picture of the task cannot be edited
- The current database design has gone beyond the scope of ScavengeRUS Tier 2, (overcomplicated and incorrectly configured for the purpose of Starting one hunt and allowing multiple people to participate on a team to complete a hunt.) Currently, to be able to have competing teams you must create separate teams and track them completely separately.
- The Locations/Edit and Locations/Delete pages are set to redirect back to the Locations index page. We are unable to make the redirect go back to Hunt/ViewTasks that we are editing in because the Hunt.id functionality is wrapped inside of a for loop that will not allow it to be used outside of that loop. Therefore instead of going back to the Hunt/ViewTasks it will go back to the Hunt Index page.

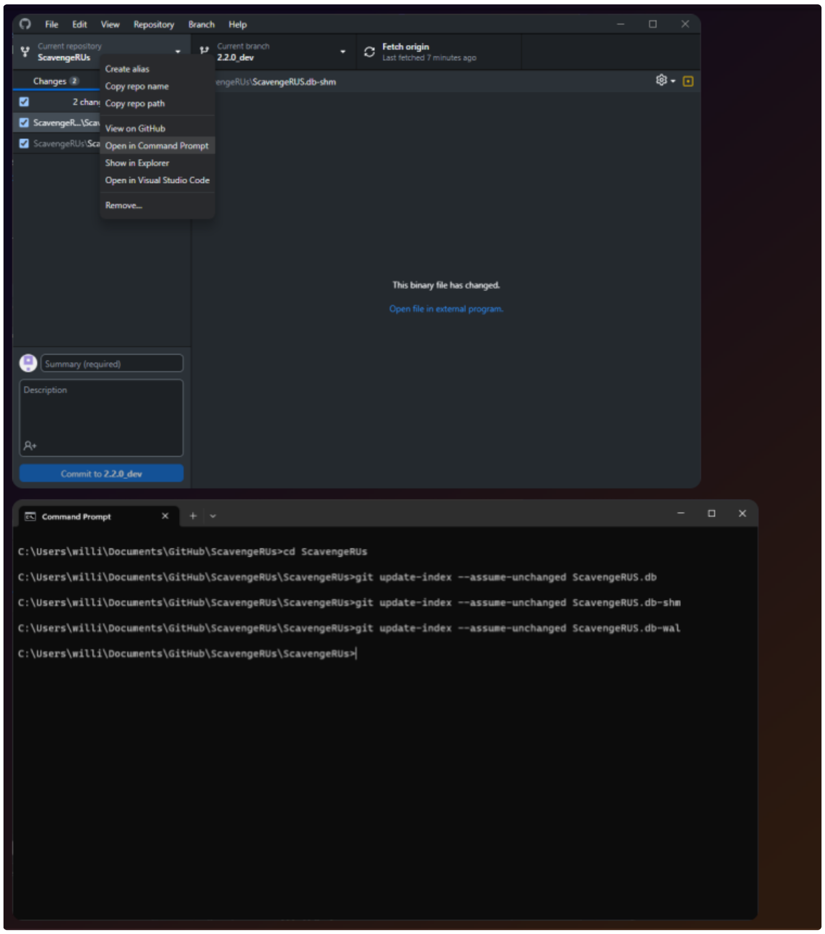
Database source control solution



When you go to commit changes, if you see this:

This binary file has changed.

you can either uncheck the files and do your commit, or you can install Git bash (https://git-scm.com/downloads) then do the following:



This binary file has changed.

Open file in external program.

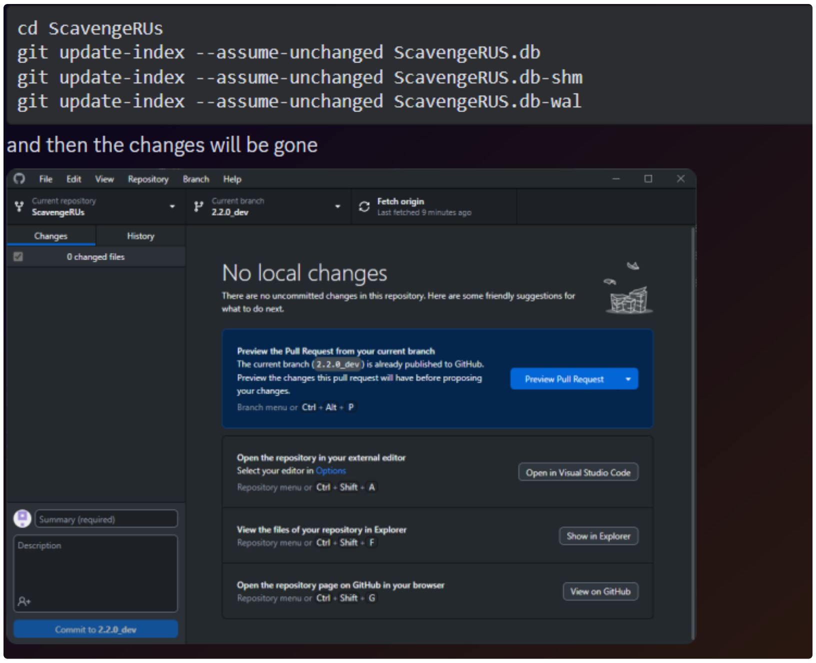
```
C:\Users\willi\Documents\GitHub\ScavengeRUs>cd ScavengeRUs

C:\Users\willi\Documents\GitHub\ScavengeRUs\ScavengeRUs>git update-index --assume-unchanged ScavengeRUS.db

C:\Users\willi\Documents\GitHub\ScavengeRUs\ScavengeRUs>git update-index --assume-unchanged ScavengeRUS.db-shm

C:\Users\willi\Documents\GitHub\ScavengeRUs\ScavengeRUs>git update-index --assume-unchanged ScavengeRUS.db-wal

C:\Users\willi\Documents\GitHub\ScavengeRUs\ScavengeRUs>
```



```
cd ScavengeRUs
git update-index --assume-unchanged ScavengeRUS.db
git update-index --assume-unchanged ScavengeRUS.db-shm
git update-index --assume-unchanged ScavengeRUS.db-wal
```

and then the changes will be gone

Current repository
ScavengeRUs

Current branch
2.2.0_dev

Fetch origin
Last fetched 9 minutes ago

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Preview the Pull Request from your current branch
The current branch (`2.2.0_dev`) is already published to GitHub. Preview the changes this pull request will have before proposing your changes.
Branch menu or Ctrl + Alt + P

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl + Shift + A

View the files of your repository in Explorer
Repository menu or Ctrl + Shift + F

Open the repository page on GitHub in your browser
Repository menu or Ctrl + Shift + G

PO Collaboration

Meeting 11/27

Issues

- Code Naming is inconsistent for tasks.
- When one task is marked complete, it is marked complete for users.
- If you interact with the admin profile and view a hunt, it will change all the completed tasks to incomplete for the players.
- First and Last names are required when adding players to a hunt as an admin.
- Removing null from the database.
- Lack of organization in the documentation.

Progress

- Built a timer
- Filter hunts and associated data
  - Filter by hunt, player, status, etc.
  - Viewable only on the admin side
- Made the incomplete hunts appear at the top, auto-sorted - Chris' team
- Source Control
  - All teams created development branches and left the previous version’s main as their archive of the old production code.
  - Once the review took place, the teams

Planned

- Sprint 4 will be for task assignment - Expectation Exceeders
  - In the tier two rollout, there are 3 different modes.
    - Implementing those different modes.
- Have a user login with 2 factor authentication - Chris' team

3 Major Accomplishments

Productivity Pirates

- Collaboration
- Establishing a clear definition of “Done”
- Creating a positive admin experience with the site.

Expectation Exceeders

- Pick accurate workloads
- Communication
  - Anytime there was something of use, it went to their Discord channel.
- 2 Separate Backlogs - One for each role
  - Admins
  - Players

Scrum-Diddly-Umptious

- Creating a clear roadmap.
- Clear standards.
- Enhancing user and admin experience.

Product Backlog Items

- Description invitation box functionality within edit hunt page
- Create a way to store tasks specific to a hunt
  - User Story: Eventually, I would like tasks to be hunt specific and move away from having a “bank” of hunts
- Remove the Master List of tasks from the database
  - User Story: When a user completes tasks in a hunt, we want them to come back and continue doing more hunts over time. They should have different tasks for each hunt
  - Acceptance Criteria: The database should have tables for each hunt rather than a master list for all hunts
- Create the ability for users to change their password
- Implement password reset functionality for user accounts
  - Description: As a user, I want to reset my password using either my provided phone number or email address to gain access to my account in case of forgetting my current password.

    Acceptance Criteria:

    i. Scenario - Initiation of Password Reset:
      - User can initiate the password reset process via the "Forgot Password" link on the login page.
      - User is prompted to input either their phone number or email address to proceed.

    ii. Scenario - Verification Process:
      - Upon submission of the phone number or email address, the system sends a verification code/link to the provided contact method.
      - The verification code/link should be delivered within 1 minute.

    iii. Scenario - Code/Link Validity and Security:
      - The verification code/link remains valid for 10 minutes after generation.
      - If the code/link is not used within this time frame, it should expire and prompt the user to request a new one.

    iv. Scenario - Changing Password:
      - After successful verification, the user is directed to a password reset page.
      - User should be able to set a new password that meets the defined password strength criteria.
      - The entire process from initiation to setting a new password should not exceed 5 minutes.

    v. Scenario - User Notification:
      - User receives a confirmation message upon successful password reset, confirming the action has been completed.

    Notes:
      - The reset password functionality should be developed adhering to best security practices to prevent unauthorized access.
      - Ensure the user interface is intuitive and user-friendly throughout the password reset process.

    Estimation:
      - Implementation of the password reset feature is estimated at X story points, considering frontend and backend development, testing, and potential security reviews.
- The admin interface should be clear to understand and follow
  - Acceptance Criteria: Admin view buttons should be consistent, match the navigation buttons for all other admin pages to those found at the top of the hunt page.
- Enhance website responsiveness and mobile experience
  - Description: As a user, I expect the website to seamlessly adapt to various screen sizes on iPhones, providing an optimal viewing experience and functionality across iPhones. The target audience of the hunt uses iPhones more than any other brand of smartphones.

    Acceptance Criteria:

    i. Viewport Responsiveness:

  - The website layout adjusts fluidly to different screen sizes and resolutions, optimizing the viewing experience.
  - Navigation menus collapse into a mobile-friendly icon or button on smaller screens.
  - No horizontal scrolling is required to access content.

ii. Navigation and Touch Accessibility:

  - Navigation menus and links are easily accessible and usable on touch screens with clear labels.
  - Interactive elements (buttons, links) have touch-friendly sizes to prevent accidental taps.

iii. Text Legibility and Forms:

  - Text is legible without zooming, with font sizes responsive to various screen sizes.
  - Forms and input fields are designed with touch-friendly targets for easy interaction.
  - Error messages are clearly visible, and form validation functions effectively.

iv. Touch Interactions and Orientation:

  - Interactive elements respond appropriately to touch gestures.

v. Consistency, Accessibility, and Compatibility:

  - Ensure consistency in the website's look and feel across different devices for a cohesive brand experience.
  - Verify compliance with accessibility standards (e.g., WCAG) using screen readers and assistive technologies.
  - Test the website on various mobile browsers for consistent functionality and appearance.

Notes:

i. Consider implementing mobile-specific features that could enhance user engagement and functionality on mobile devices.

ii. Continuous testing and optimization are crucial to maintain optimal performance and user experience across diverse devices.

- As an Admin, I want to be able to send notifications to a subset of players in a hunt so that I can keep them updated on any changes or alerts
  - Notifications can be sent via email or text (admin chooses).
  - Admin can select all or a subset of players in the hunt.
  - Admin manually enters the notification text.
- As an Admin, I want to be able to invite one or more people to play in a specific hunt so that I can control and manage who is involved
- As an Admin, I want to be able to quickly create accounts for a list of people so that I don’t have to enter each person manually
  - Create a new account using a person’s email address and phone number.
  - Assign an access code unique to that player for that hunt (can be duplicated for that player across hunts).
  - Find an existing account using a person’s email address and phone number.
  - Find an existing account using a person’s access code.
  - Invite someone to participate in a hunt by sending the hunt’s url and a invitation message to their email, then to send their unique access code to their phone via text message.
  - Access codes can be active or disabled or pending invite
- Fix the bug for databases where when one participant completes a task it complete for it all
- Create a view account page for users
  - Acceptance Criteria: A view account page should pop up whenever I click on my name in the top right corner. It should include the ability to edit my name, reset password, and delete my account. Selecting the “Delete Account” button should prompt me to confirm before deleting the account.