SE 492 Biweekly Report
1/27/2024 - 2/10/2024

sdmay24-41
Griffith Buck Rose Website
Lindsey Smith: Client
Mai Zheng: Advisor
Amy Hartjen: Front-End Lead / Communications
Erik Sandberg: Database Lead
Devin Amdahl: Scrum Master / DevOps
Patrick Origer: Research / Third Party Manager
Greg Carter: Connections Master
Logan Schmidt: Backend Helper
Alex Reynolds: Testing Manager

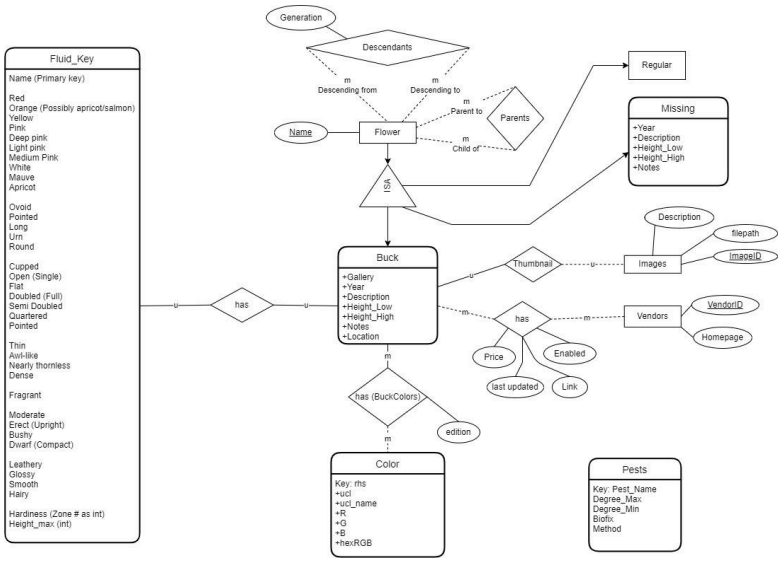
Report Period Summary:

In the last two weeks, our group met with our advisor and client to discuss ways that our project could be expanded to increase complexity, as well as how to implement the fluid key sorting feature that the client requested. For frontend, we implemented a given footer design and collected requirements and information for the “About” and “Griffith Buck” pages. For backend, we have made a lot of progress regarding our database. This progress was led by Erik.

Past Period Accomplishments

Collectively as a group, we met up for client & advisor meetings.

- Erik Sandberg
    - Working on initializing data in the database
    - Validating CSV data files for importing into the database
    - Finishing schema for all tables on the database
    - Creating SQL scripts to insert csv files into the database
    - Updated Design Schema for database tables



- Greg Carter
    - Due to several technical issues, I wasn't able to do anything for this project. As of 2/9/24 they are all resolved and I will resume work this week.
- Amy Hartjen
    - Completed footer as specified by client and received client approval on it
    - Received information needed from client to be put on the About Griffith Buck page and the glossary for the About Reiman Gardens Page so we can begin implementation of those
    - Updated the website design to be more intuitive and navigable
- Patrick Origer
    - I came across one decent profanity filter. The filter's library of inappropriate language has not been updated since about 2019, so I will likely continue looking for a more updated profanity filter, or will consider layering two text profanity filters using this one and maybe another newer one. 2019 profanity filter: https://github.com/vzhou842/profanity-check
    - I spent a good while reading through Weather.gov documentation for the API. Searching for API calls I need and choosing only those that are unlikely to be deprecated, as weather.gov has a system to denote specific calls and the likelihood they will be altered or deprecated, which could break some functionality of the Degree Day calculator down the line. I also wrote some API calls to receive some temperature data, which did return what was needed for that test.
    - I lastly searched for methods of mapping a sort of coordinate system onto an image, with our case being a map of Iowa and the coordinate system being latitude and longitude.
- Devin Amdahl
    - Cleaned up our issue board as it hadn't been updated over break. Added comments to old issues that hadn't been closed, closed relevant issues, and created new issues that prioritized the completion of our MVP by February 18th.
    - While we have since agreed upon implementing the fluid key sorting on frontend (more performant), I had researched Spring Data JPA's query by example functionality and other sorting/filtering functionalities and then created a short write-up explaining how we could utilize these functionalities to handle the fluid key sorting on backend.
    - I met with Greg to discuss the current state of the backend codebase, what we need to prioritize to meet our MVP deadline, and the primary areas of the codebase that will need to be refactored/replaced by Spring Data JPA and JPA Repository. This meeting lasted for approximately an hour. I met with Erik to go over the old create.sql and drop.sql scripts and we identified some portions that we thought should be changed. During this call I installed MySQL Server onto our team server, made it a service, created our 'seniorDesign' schema, and ensured that Erik could access the schema with his MySQL account. From there, Erik

executed the create.sql script and created the schema's tables. This meeting lasted for approximately 1.5 hours.
    - Ensured that the MySQL Server would restart in response to random crashes (killing the program).
    - Created MySQL accounts for all team members, and provided two sets of directions for accessing the schema. One set of instructions for accessing the schema via CLI after connecting to the server using SSH, and another set of instructions for accessing the schema remotely using MySQL Workbench and TCP/IP over SSH.
- Alex Reynolds
    - Created a couple different potential roadmaps for mobile support and discussed them with teammates.
    - Gained a better understanding of current progress on backend and frontend. Started the process of refactoring the backend.
    - Created a better plan for FluidKey.
    - Made sure testing dependencies are installed, set up testing files, and finalized plans for testing.

Issues

- Erik Sandberg
    - Issue 36: Create SQL Script for Generating Schema
    - Issue 83: Modify Database Schema to Include Fluid Key Attributes for Buck Flowers
    - Issue 78: Finalize VBA Scripts and Validate All Data Needed to Initialize Our Server's Database
- Amy Hartjen
    - Issue 92: Create Website Footer Consistent with Figma Prototypes
- Patrick Origer
    - Issue 86: Research Text Moderation Services/Libraries for Community Page Moderation
    - Issue 87: Research Input Validation Services/Libraries for Community Page Moderation
    - Issue 88:Verify that Weather Results from Weather.gov API are Consistent and Accurate
    - Issue 90: Research Functionality for Displaying Map of Iowa with Underlying Coordinate System
- Devin Amdahl
    - Issue 74: Meet with Relevant Parties to Close "Waiting" Issues
    - Issue 79: Create Database Instance on Server
    - Issue 82: Research Libraries Needed to Implement Fluid Key Functionality on Backend and Create Rough Outline for Implementation
    - Issue 84: Create MySQL User Accounts for All Team Members

    - Issue 99: Create a Gitlab Runner for Our Project's Repository
        - Still working on this.
- Alex Reynolds
    - Issue 40: Added files to backend and frontend, began refactoring process, and ensured correct dependencies were installed for backend and frontend.
    - issue 101: Created a more detailed plan for my new idea that we should handle fluid key filtering on the frontend

Individual Contributions

Person	Issue	Contribution	Date	Hrs
Amy Hartjen	#92	Implemented through issue 98 but unrelated, fixing spacing and overall design of the website to better fit design standards	1/27-2/3	3
Erik Sandberg	#36	Validation changes, edit create and drop tables script. Changes are also reflected in new schema design	2/7	3
Erik Sandberg	#83, #36	Added filter key entity to ER diagram in issue #83, implementation is included in #36 create tables file	2/6 - 2/7	2
Erik Sandberg	#78	Manually validated data in excel spreadsheet, created csv files for each table for the given data we have	2/8	5
Amy Hartjen	#92	Updated footer, waiting for client confirmation	2/8	2
Amy Hartjen	#92	Received client feedback and finished footer	2/9	1
Patrick Origer	#86, #87	Looked into options for text filtering libraries	2/9	1
Patrick Origer	#88	Got a chain of APIs with Weather.gov to return temperature data on specific days, and worked on getting all hourly temperatures to be returned from time requested to the beginning of year.	2/7-2/9	4
Patrick Origer	#90	Found some methods of how to project a coordinate system onto an image	2/10	1
Logan Schmit	#81	Began mapping out the necessary attributes from the ER Diagram	2/10	2
Devin Amdahl	N/A	Cleaned up our issue board as it hadn't been updated over break. Added comments to old issues that hadn't been closed, closed relevant issues, and created new issues that prioritized the completion of our MVP by February 18th.	1/27-2/3	2
Devin Amdahl	#82	While we have since agreed upon implementing the fluid key sorting on frontend (more performant), I had researched	1/27-2/3	1

		Spring Data JPA's query by example functionality and other sorting/filtering functionalities and then created a short write-up explaining how we could utilize these functionalities to handle the fluid key sorting on backend.		
Devin Amdahl	#74, #79	I met with Greg to discuss the current state of the backend codebase, what we need to prioritize to meet our MVP deadline, and the primary areas of the codebase that will need to be refactored/replaced by Spring Data JPA and JPA Repository. This meeting lasted for approximately an hour. I met with Erik to go over the old create.sql and drop.sql scripts and we identified some portions that we thought should be changed. During this call I installed MySQL Server onto our team server, made it a service, created our 'seniorDesign' schema, and ensured that Erik could access the schema with his MySQL account. From there, Erik executed the create.sql script and created the schema's tables. This meeting lasted for approximately 1.5 hours.	1/27-2/3	2.5
Devin Amdahl	#79	Ensured that the MySQL Server would restart in response to random crashes (killing the program).	1/27-2/3	0.5
Devin Amdahl	#84	Created MySQL accounts for all team members, and provided two sets of directions for accessing the schema. One set of instructions for accessing the schema via CLI after connecting to the server using SSH, and another set of instructions for accessing the schema remotely using MySQL Workbench and TCP/IP over SSH.	1/27-2/3	1
Devin Amdahl	#99	Installed JDK, Maven, Docker, and GitLab Runner on our team's server. Created and registered two different GitLab runners (one for backend, one for frontend). More progress is required.	2/3-2/10	0.5
Alex Reynolds	#101	Did research into changing our approach for fluid keys. Brought it up to team that this needs to be changed to a frontend issue because it will heavily simplify our code and map->reduce->filter is traditionally a frontend js problem anyway	2/4	.5
Alex Reynolds	N/A	Did research into some technologies that could be helpful (swagger codegen, and progressive mobile webapps)	1/28-2/1	1
Alex Reynolds	#40	Added placeholder tests files for backend and frontend. These can definitely help guide the refactoring on backend and future development on frontend. Created a report as well. Created the files using research on standard springboot design practices, or ER diagram, communicating with team members, and based on what has been said in meetings	2/10	3

Person	WT	Total
Erik Sandberg	10	10
Amy Hartjen	6	6
Alex Reynolds	4.5	4.5
Greg Carter	0	0
Logan Schmit	2	2
Devin Amdahl	7.5	7.5
Patrick Origer	6	6

Plans For Upcoming Report Period

- Erik Sandberg
    - Work on finishing importing data onto the database
    - Generate fluid key CSV file using attributes received from the spreadsheet Lindsey provided
    - Search for pest data
    - Get information on available vendors
        - Find which buck flowers each vendor has
    - Get buck flower's gallery file path
    - Compile data to fill 'Image' entity
- Greg Carter
    - Issue 95: Create Page Layout for Individual Buck Rose Pages
    - Issue 80: Placeholder for Backend Community Page Functionality
        - This feature will be broken down into smaller stories
- Amy Hartjen
    - Issue 93: Create "About Dr. Griffith Buck" Informational Page
    - Issue 94: Create "About Reiman Gardens Page" Informational Page
- Patrick Origer
    - Finish:
    - Issue 86: Research Text Moderation Services/Libraries for Community Page Moderation
    - Issue 87: Research Input Validation Services/Libraries for Community Page Moderation
    - Issue 88:Verify that Weather Results from Weather.gov API are Consistent and Accurate
    - Work on:

        - Issue 85: Research Image Detection Services/Libraries for Community Page Moderation
        - Issue 90: Research Functionality for Displaying Map of Iowa with Underlying Coordinate System
        - Issue 91: Reach Out to Weather.gov for API Usage Limits
- Devin Amdahl
    - Issue 99: Create a Gitlab Runner for Our Project's Repository
        - Necessary tools/software has been installed on our team's server, and two GitLab runners have been created and registered. Still need to implement and test the .gitlab-ci.yml configuration file.
    - Issue 100: Generate Controller Documentation Using Swagger UI and Host the Documentation On Our Server
    - Assist in refactoring backend to utilize Spring Data JPA and JPA Repository
    - Assist in backend development to reach our MVP deadline of February 18th

Summary of Advisor Meeting

- We discussed with our advisor on how our approach to implementing fluid keys could be improved. Our current implementation primarily relies on boolean values to indicate if a Buck flower has the specified attributes. Our advisor stated that we could also store the fluid key attributes based on their associated data types.
- We discussed different ways to go about authenticating administrator users to manage the website through the admin portal. While we had thought about authenticating administrator users of the website through Iowa State University's Okta, we were worried about the complexity of the implementation. After speaking with our advisor, he reassured us that the implementation would not be overly complex. By authenticating administrator users of the website through Okta, we can mitigate the majority of our project's security concerns.