Oregon State University

Corvallis, Oregon

CS 361 - Software Engineering I

Connecting Animal Shelters With People

Group 8 Members: Aimee Sullivan, Andrew Kim, James Le, Matthew Weakley, Ya Lien

Project Website**:** http://web.engr.oregonstate.edu/~lej/361/home.php

Table of Contents

XP Cycle 1:

User Stories

Create Shelter Account
As a shelter side user, I can create Shelter account that is unique from the general public's user account. Only users with verified shelter email addresses should be allowed to create such accounts. The account should be linked with the Staff Accounts that associate with the shelter.

Create Staff Account
As a staff of the shelter, I can create Staff account that allows access to the Shelter account and remove/update information on the Shelter Account. This account should be verified by the Shelter Account.

Create General User Account
As a general user, I can create an account by providing a username, password, age, gender, and location.

Shelter's Staff Provides Pet Info
After associating with the shelter's account, the staff member will be able to log into their account and write descriptions about the pets in a form. The form will have several categories, and a box option to give details about the animal. Such as the animals sex, its age, its weight and height, what its fears are, its activity, and its background story. It will also include the name of the shelter and a contact number.

General Public Retrieves Pet Info by Quick Search Bar
The general public who are searching for the pet will be able to search and view the pet's information provided by the staff member without requiring an account. There will be a quick search bar for the user to type in what type of pets they want.

General Public Retrieves Pet Info by Search Form
The general public who are searching for pets will be able to search and view the pet's information provided by the staff member without an account. There will be a search form that allows the user to fill in more details on the search. The form will include the type of pet, its age range, its gender, and its characteristic. A list of pets will then populate in another page with all of its information.

General Public Provides Info
After the general public creates an account, they can provides their info, such as gender, age, location, and etc. in a form, which is then input into the database, and the website will take this into consideration when the users uses the quick search bar, and matches them with the pet depending on compatibility and criteria.

Shelter Schedule

The staff member will be able to create a schedule, which will provide visibility to the general public about times and days that visitations are allowed. This schedule will also have events such as large volunteer days, park days, etc.

Individual Pet Schedule

The staff member will be able to create a schedule for each individual animal. This will allow the user to look through individual schedules or to search through a list of available animals for the time and day that they are available.

Tasks and Effort Estimates

** 10 effort points per week **

Create Shelter Account

Effort Estimation: 2 points

- Set up a database to store account information
- Create web page for user to enter required information
- Create web page to show the created account

Create Staff Account

Effort Estimation: 2 points

- Set up a database to store account information
- Create web page for user to enter required information
- Create web page to show the created account

Create General User Account

Effort Estimation: 2 points

- Set up a database to store account information
- Create web page for user to enter required information
- Create web page to show the created account

Shelter's Staff Provides Pet Info

Effort Estimation: 2 points

- Create a web page for staff to input pet data
- Information is inserted into database
- Create a web page to display the pet information

General Public Provides Info

Effort Estimation: 2 points

- Create a web page for general public to provide their information
- Input the information into the database
- Create a web page to display the information

General Public Retrieves Pet Info by Quick Search Bar

Effort Estimation: 1 point

- Create a quick search bar on a web page
- Make queries to the database
- Create a web page which returns the result

General Public Retrieves Pet Info by Search Form

Effort Estimation: 1 point

- Create a web page with an input form
- Make queries to the database
- Create web page to show search result

Individual Pet Schedule

Effort Estimation: 3 points

- Create a web page to display calendar information
    - Select among available pets in the database
    - Select month and year
    - Default to current month and year
    - Display all calendar appointments for selected pet in current time frame

Shelter Schedule

Effort Estimation: 3 points

- Create a web page to display calendar information
    - Select among available shelters in the database
    - Select month and year
    - Default to current month and year
    - Display all calendar appointments for selected shelter in current time frame

User Stories Prioritized by Customer

1. Create Shelter Account
2. Create Staff Account
3. Shelter’s Staff Provides Pet Info
4. Unknown user search
5. Create General User Account
6. General Public Retrieves Pet Info by Quick Search Bar
7. General Public Retrieves Pet Info by Search Form
8. General Public Provides Info
9. Individual Pet Schedule
10. Shelter Schedule
11. General Schedule

Collaboration

Splitting into pairs was interesting in a group of 5, but we managed to work in teams of 2 and 3. Before any work began, we held a meeting to discuss the vision statement and what work was ahead of us. From there we all came up with clarification questions to ask our customer about the vision statement. After our questions were answered and we were ready to begin working, we organized who was working in which group.

The team of 3, which we will call group A (James, Matthew, and Ya), started the project off by doing the database design. The work started with creating a visual representation of the database. This included an entity relation (ER) diagram and then a database schema. After the design was finalized, then came the programming. The database was written and hosted on the OSU MS SQL Server due to familiarity and simplicity. Group B, after the database was intact, took on the PHP frontend to interact with the created database.

The team of 2, Group B (Aimee and Andrew), met on a regular basis to code together, set objectives for coding on our own time, and coordinate tasks. We primarily used c9.io as our working coding platform, with the latest version synced to github. We were able to utilize our knowledge of php and databases from a previous class to get up to speed quickly but found that we needed to collaborate with Group A quite a bit due to our deliverables' dependencies on the database structure. Primarily, we used Google Hangouts chat to coordinate activities and used our unit testing document to keep track of bugs, including those that may be a result of the database structure.

The main method of collaboration between the groups, and even the team as a whole, has been Google Drive. Google Drive is where all of our project documents, including code, are stored. Each team member can upload, view, and edit all documents simultaneously, making pair work a breeze. We chose Google Drive because it was a reliable, free service we were all familiar with. Google services also allow us to communicate very easily and effectively.

The biggest issue we faced as a team was finding times to collaborate. We have members in several different time zones, from Pacific to Eastern, so for us to all find a time to either have meetings or collaborate on the project is difficult. To further complicate the issue, we all have different work schedules, with some being full time students and others having full time jobs. Even through our complicated schedules, we have been able to stay in touch consistently through Google Hangouts. One benefit for us has been the retention of chat history. If someone wasn't active in the conversation at the time, they can read it later and make some contribution after.

Unit Tests

Manual unit tests run on test site/database at
http://web.engr.oregonstate.edu/~lej/361/home.php

Unit Test	Pass/Fail	Comments
Able to create user_account table	PASS	-
Able to create pet_account table	PASS	-
Able to create shelter_account table	PASS	-
Able to create calendar table	PASS	-
Able to create schedule_appointment table	PASS	-
Able to create play_with table	PASS	-
Shelter information can be input into database	PASS	Springfield record is in database Springfield Animal Control record in db
Shelter form should have the right type for each input into the form	PASS	Shelter name, Permission code, First name, Last name input type = text Password, Confirm password input type = password Database fields for shelterName, firstName, lastName, password,email, permissionCode are all varchar(255)
Shelter account must be able to be associated with zero to many staff accounts	FAIL	No foreign key for shelter id in user_account table. Multiple shelter accounts representing user make it infeasible to associate pets with a shelter upon entry by shelter staff user.
Staff account form has the right information	PASS	User information is in member profile
Staff information can be input into database	PASS	Record for jsmith added to database
Staff form should have the right type for each input into the form	PASS	Username, email, First name, Last name, Permission code input type = text Password, Confirm password input type = password Database fields for userName, firstName, lastName, password,email, permissionCode are all varchar(255)
Staff account can be associated to one shelter	FAIL	Working on fix
Pet is associated with one shelter	PASS	Grab current session's shelter name and then get the id
Pet has multiple categories	FAIL	Database needs to be updated to include additional categories, per the user story
Updated User Account Creation	PASS	User account created within shelter account with appropriate permission code and access level

Add event to the calendar table	PASS	Event successfully added to the calendar table with appropriate shelter, user, and pet account references
Add to “play with” database table	PASS	Successfully added to “play with” database table with appropriate user and pet account references

Bugs Found	Comments
Typo found in play_with query	
Typo found in schedule_appointment query	
Syntax error found in pet_account	Removed comma after ON DELETE CASCADE
Lack of shelter id in user_account	Fixed added sid foreign key to user_account
Shelter accounts should be removeable	
No database fields for shelter location information	Allow users or general public to search for shelters/pets by location
No way to associate staff with shelter.	Fixed added sid foreign key to user_account
Shelter account needs to have a phone number so that it can be associated to the pet, per the user story	
Pet database also needs a open form text so that shelter can add info like fears, background, and activities, per the user story	

Acceptance Tests

Manual unit tests run on test site/database at
http://web.engr.oregonstate.edu/~lej/361/home.php

Acceptance Test	Pass/Fail	Comments
Shelter account can be created	PASS	Shelter name: **Springfield** Password: **sac** Confirm password: **sac** Email: **sullivai@oregonstate.edu** First Name: **Aimee** Last Name: **Sullivan** RESULT: **Account created. Verification email sent.** Shelter name: **Springfield Animal Control** Password: **sac** Confirm password: **sac** Email: **sullivai@oregonstate.edu** RESULT: **Account created. Verification email sent.**
Shelter account email must be verified	FAIL	Verification link in email clicked. Shelter name without spaces successful. Shelter name with spaces link is incorrect.
Shelter account form has the right information	PASS	Once logged in, profile view information is correct. Items that were left blank are blank; items that were provided are correct. Creation date is correct date.
Same account cannot be created	PASS	Shelter name: **Springfield** Password: **sac** Confirm password: **sac** Email: **sullivai@oregonstate.edu** RESULT: **Shelter name unavailable. Please try again.**
Account creation should fail if required inputs are missing	PASS	Shelter name: **Cat Town** Password: **Password** Confirm password: **Password** Email: RESULT: **Form does not submit. No entry in database.** Shelter name: **Cat Town** Password: **Password** Confirm password: Email: **a@b.com** RESULT: **Form does not submit. No entry in database.** Shelter name: **Cat Town** Password: Confirm password: **Password** Email: **a@b.com** RESULT: **Form does not submit. No entry in database.** Shelter name: Password: **Password** Confirm password: **Password** Email: **a@b.com** RESULT: **Form does not submit. No entry in database.**

Account creation should fail if passwords do not match	PASS	Shelter name: **Cat Town** Password: **password** Confirm password: **Password** Email: **a@b.com** RESULT: **Passwords do not match. Try again. No entry in database.**
Registration works in common browsers	PASS	Functions work in IE11, Firefox 53, Chrome 58 on Windows 10
Staff account can be created	PASS	Username: **jsmith** Password: **thedr** Confirm password: **thedr** Shelter: **Springfield** Permission Code: **cats** RESULT: **Account created.**
Same staff account cannot be created	PASS	Username: **jsmith** Password: **kelad** Confirm password: **kelad** Shelter: **Springfield** Permission Code: **cats** RESULT: **Username unavailable. Please try again.**
Staff account creation should fail if certain required inputs are not used	PASS	Shelter name: **jimmy** Password: **thefish** Confirm password: **thefish** Shelter: Permission Code: RESULT: **Incorrect shelter information. Please contact administrator for permission code. No entry in database.** Shelter name: **jimmy** Password: Confirm password: Shelter: **Springfield** Permission Code: **cats** RESULT: **Form does not submit. No entry in database.**
Staff account can edit shelter information	FAIL	Logged in as jsmith. Add new animal form visible... PASS Shelter information (address/phone) not available to edit…. FAIL
Able to create a pet	PASS	
Staff member is able to input pet info	PASS	
Duplicate pets cannot be created	PASS	
Pet account creation should fail if certain required inputs are not used	PASS	
Shelter accounts can remove other accounts	PASS	

Bugs Found	Comments
Problem with redirection in Chrome	Fixed <meta http-equiv> syntax

Shelter name with spaces in it breaks email verification link	Fixed urlencode() function added to registershelter.php
Shelter information not available for staff to edit	
Staff accounts should be removable	Fixed in second cycle
Pets need to be removable	Fixed in second cycle

XP Cycle 2:

Updated User Stories, Effort Estimates, and Priorities

** 10 effort points per week **

Create Shelter Account: Modifications

Effort Estimation: 1 points

- Add location as a database option
- Add phone number as a database option
- Make account removable

Create Staff Account: Modifications

Effort Estimation: 1 points

- Staff needs to be able to edit shelter information
- Make account removable

Shelter's Staff Provides Pet Info: Modifications

Effort Estimation: 1.5 points

- Associate shelter phone number to the pet
- Add background and activities as a database option
- Create open text form to add misc info to the database about the pet
- Make account removable

General Public Provides Info

Effort Estimation: 2 points

- Create a web page for general public to provide their information
- Input the information into the database
- Create a web page to display the information

General Public Retrieves Pet Info by Quick Search Bar

Effort Estimation: 2 point

- Create a quick search bar on a web page
- Make queries to the database
- Create a web page which returns the result

General Public Retrieves Pet Info by Search Form

Effort Estimation: 2 point

- Create a web page with an input form
- Make queries to the database
- Create web page to show search result

Individual Pet Schedule

Effort Estimation: 3 points

- Create a web page to display calendar information
    - Select among available pets in the database
    - Select month and year

    - Default to current month and year
    - Display all calendar appointments for selected pet in current time frame

Shelter Schedule

Effort Estimation: 3 points

- Create a web page to display calendar information
    - Select among available shelters in the database
    - Select month and year
    - Default to current month and year
    - Display all calendar appointments for selected shelter in current time frame

Summary

For the second XP cycle we refined a number of our user story features that we had implemented in a basic form during the first cycle, and added some additional ones. For example, we added several entities to the database tables for the shelter account, including shelter phone number and location. Additionally, we added shelter information editing capabilities under the staff account, and modified the pet table attributes and refined the data entry form for pets under the shelter staff account. We added deletion functionality (individual users, staff, and pet information) under the shelter admin account. All of these changes added capabilities in line with the customer's requirements. We also implemented several additional user stories which we had not completed during the first XP cycle, including the pet quick search feature for the general public, and basic pet schedule display. We also improved the look of the front-end interface to make it more logical, user-friendly, and aesthetically pleasing.

Unlike the first XP cycle, the groups were a little less structured and our overall interaction was more asynchronous for the second cycle. Once we got our direction from the customer, it felt like we all got very busy with other work so were less available to meet and chat as a full group. Nevertheless, group members would contribute to the chat regularly, and although responses were not usually immediate, someone from the group would manage to eventually address all questions and issues. We still had our database and PHP teams to make changes and fixes to code from the first cycle, but responsibilities branched out from there and the teams worked more independently, with different groups focusing on different parts of the system. While any new features that were implemented by a pair generally involved modifications to the database, the PHP, and documentation of the test results, we were able to keep track of what changes were needed and when they were made via the history in Google docs, chats, and hangouts. Finally, we migrated our code from the test site and database that we had used in the first cycle to our production site.

As a team, we felt the second cycle was easier than the first, as we had put a substantial amount of effort into laying the groundwork for our project while implementing our user stories during the first cycle. This was not so much through analyzing requirements in advance as with a waterfall methodology, but rather the structure was informed completely by the features we were planning to implement. With the second cycle we were able to take that foundation and transform it into a fully functioning program by expanding our initial set of features and making small additions to the database and SQL queries to implement new ones. Even though our two teams from cycle 1 each had a different focus, by necessity, we all had to understand the workings of the project and of each other's design and code. That understanding of the overall database structure and how our php functioned made it possible for different, fluid pairs to easily implement slight adjustments, additions, and improvements during the second XP cycle with little need for major design modifications. This allowed us to focus more attention on improving the project and finalizing our report.

Reflection

Both XP, a method of the agile workflow, and waterfall have their unique benefits and drawbacks. After working with each method for a few weeks, we have been able to experience them first hand and can decide what methods we like in particular situations.

When using waterfall, the big key was to have a very thorough and detailed code plan from the get go. From inception of the project, the whole timeline needs to be in place. This includes defining requirements, system design, prototyping if needed, implementation, testing, operation, ongoing maintenance, and detailed documentation. When done right, waterfall method makes for a straightforward, organized project. The extremely detailed documentation also allows for new team members to jump in on the project with little downtime. Most importantly, the customer will be able to read the plan and have a clear idea of where the project is headed, with all the milestones laid out very explicitly.

A major drawback to waterfall is the amount of forethought that must go into it. If one little piece of the pipeline was done incorrectly, or something unforeseen causes problems, changes must be made and the project will need to start all over. This sounds like a very risky approach, especially if a particular project has some unknown elements from the start. It is often difficult to anticipate all possible outcomes and problem points, so it would be easy to run into major problems using waterfall.

Agile, on the other hand, is a completely different world. Where waterfall requires meticulous thought and planning, agile is more of a “go with the flow” kind of process. The initial design is thought up and a quick meeting is thrown together to discuss the initial steps and priorities. Afterward, repeating meetings are held at regular intervals as status updates, with the customer being able to adjust priorities as problems come up and need addressed. This is a major advantage of agile, especially when projects will have lots of churn in the design process. Not only that, but agile allows for initial iterations of a project to be completed quickly.

Even though agile seems to solve all the problems waterfall introduces to a project, it brings upon it’s own. For one, the planning and process of the whole project is very unstructured. The project could start with a certain vision and evolve into something completely different over time. On top of that, documentation is often left at the wayside in favor of quick code iterations. This makes staff changes much more difficult to successfully achieve in a reasonable time.

For the projects we worked on in this class, the XP agile process was the easier method to implement. The limited time frame allotted for the projects would make waterfall very difficult to implement successfully, no matter the simplicity. Us students as sponsors are also not likely to have a 100% clear picture of our final product of our vision statements. This, again, is a better task for the agile method.