Bark-in Software Configuration

Group 19:

Jacob Haldeman

Connor Kalina

Turner Halligan

Robert Corbett

Ian Lundberg

Table of Contents

SOFTWARE PLAN

1.0 SCOPE

Bark-in is a social networking application that allows dog owners to schedule visits to parks as well as view what other dogs will be at the park at that time. Owners can view various traits about the dogs such as temperament, size, vocalization, etc. The application will also allow park visitors to post reviews and ratings of the various dog parks that they visit, based on maintenance and cleanliness, as well as view how busy the parks are at various times and days of the week. The main purpose of the application is to allow dog owners to decide whether they want to visit a specified dog park at a given time based on what dogs are/will be there and user reviews about the park.

1.1 FUNCTIONS

- *Users will be able to create profiles for themselves and their dogs.*
- *The profiles will contain personality and physical trait information.*
- *Users will be able to link their profiles with their dogs' profiles.*
- *Users will be able to schedule a time slot when they will be at a dog park.*
- *Users will be able to see the profiles of other dogs that are also scheduled.*
- *Users will be able to view how busy a park is at a given time and day.*

- *Users will be able to read and leave reviews for a given dog park.*

1.2 Performance

- *Bark-in will be hosted on a server capable of storing a scalable amount of user and dog profiles.*
- *Bark-in will have database storage for dog park information and user/dog profiles.*
- *The server will have a starting capacity of up to a thousand concurrent users.*
- *The application will be able to perform queries to the server for updates on park information.*

1.3 Limitations

- *Bark-in will not monitor real-time user location.*
- *Bark-in will not be able to organize groups and meetups.*
- *Bark-in will only store information for dog parks based in the greater Pittsburgh area.*

2.0 Tasks

2.1 User Interface

- *The front-end programmer will be assigned to this task. They will work on the frontend of the application to create an effective and clear interface for the end user.*

2.2 Database Management

- *The back-end programmer will be assigned to this task. They will work on the backend of the application to create a database to collect and store user information, geo-data on parks, and other necessary data. The database must also be able to connect with the frontend through the application.*

2.3 Data Mining

- *The data-miner will be assigned to this task. They will work on finding all dog parks and entering them into the system.*

2.4 Testing

- *Everyone is assigned to test a certain part of the application. However, they will not test their own code or user experience.*

3.0 Resources

3.1 Hardware

- *Various personal desktop computers, laptops, tablets, and smartphones.*

3.2 Software

- *Discord for communication.*
- *GitHub for version control.*
- *Google Drive for document sharing.*
- *Visual Studio 2019*
- *Django*
- *Bootstrap*
- *Windows SQL Database*
- *HTML 5, CSS 3, JavaScript, and Python*

3.3 People

Names	Project Manager	Front-End Programmer	GUI Programmer	Application Programmer	Back-End Programmer	Tester	Sales Person	Data Miner	Presenter	Document Specialist
Jacob Haldeman	✔					✔	✔		✔	
Ian Lundberg						✔		✔		✔
Rob Corbett		✔	✔							
Turner Halligan					✔					
Connor Kalina				✔						

Group Member	Roles
Jacob Haldeman	Project Manager, Tester, Sales Person, Presenter
Ian Lundberg	Tester, Data Miner, Document Specialist
Rob Corbett	Front-End Programmer, GUI Programmer
Turner Halligan	Back-End Programmer
Connor Kalina	Application Programmer

Requirements specification

1. Product Overview and Summary

Bark-in is a social networking application which allows dog owners to schedule visits to parks by first viewing what other dogs will be there at that time. Users will be able to create profiles for their dogs which include pictures and traits about them, including breed, size, temperament, age, vocalization, etc. Once a user schedules a visit to the park, other users will be able to see their dog's profiles for that time and location. This will let users view how busy a park will be as well as allow they to see if they believe their dog will get along with the other dogs there at that time. The application will also allow users to rate and review parks on qualities such cleanliness and maintenance.

When users create an account, they will be able to create up to 16 profiles for their dogs. The application's back end will store a database for users and a separate database for dogs and the two will be linked in one to many relationships. Another database will store information and pictures for local dog parks. This parks database will be linked to another database storing the ratings and reviews for the parks. Lastly, a fifth database will be needed to store the users' scheduled visits. When a user views a park, they will be able to choose a date and time. The application will display the park's information as well as the profiles of the dogs scheduled to be there at the time. If the user finds a park and time that they find suitable, they will be able to schedule a visit and their dogs' profiles will be viewable by other users.

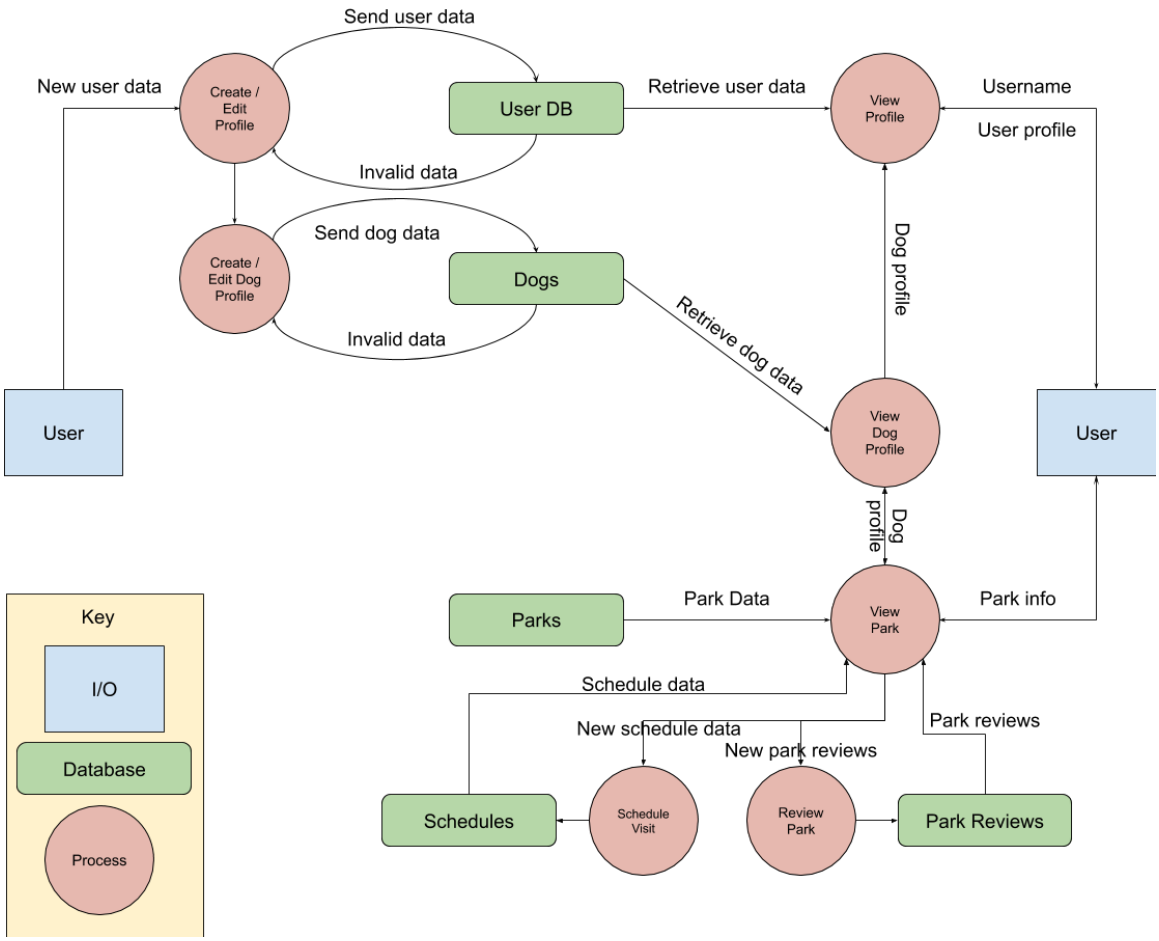
As well as this core functionality, in the future Bark-in will allow users to generate friends lists. These lists will allow users to message each other and plan visits and playdates. Users will also be able to enter information about other dogs at the park that are not part of the Bark-in community, giving other users a better picture of the parks. Also, using the data gathered from users, the application will be able to plot trends at parks allowing it to predict how busy parks will be in the future.

2. Information Description

2.1 User Manual

See attached "User Manual" document.

2.2 Data Flow Diagram



2.3 Data Structure

- user(UID, username, hashed password, owned dogs, bio, favorite dog park)
- dog(UID, name, profile picture, breed, size, temperament, activity level, volume, notes)
- park(UID, geolocation, name, info, address, cleanliness, fenced in, off leash, reviews)
- park_review(UID, used id, review, star rating, time stamp(time))
- schedule(UID, park id, dog id, time frame, date)

2.4 Data Dictionary

- UID : user
- Username : user
- Hashed Password : user
- Owned Dog 1-X : user
- Bio : user
- Favorite Dog Park : user

- UID : dog
- Name : dog
- Profile Picture : dog
- Breed : dog
- Size : dog
- Temperament : dog
- Activity Level : dog
- Volume : dog
- Notes : dog
- UID : park
- Geolocation : park
- Name : park
- Info : park
- Address : park
- Cleanliness Rating : park
- Fenced-In : park
- Off Lease : park
- Reviews List of IDs : park
- UID : park_review
- User UID : park_review
- Review : park_review
- Star Rating : park_review
- Time Stamp : park_review
- UID : Schedule
- Dog UID : Schedule
- Park UID : Schedule
- Time Frame : Schedule
- Date : Schedule

3. Functional Description

Functions, Process Narratives, Data Flow Diagrams, and Constraints

Create User

- IC Card

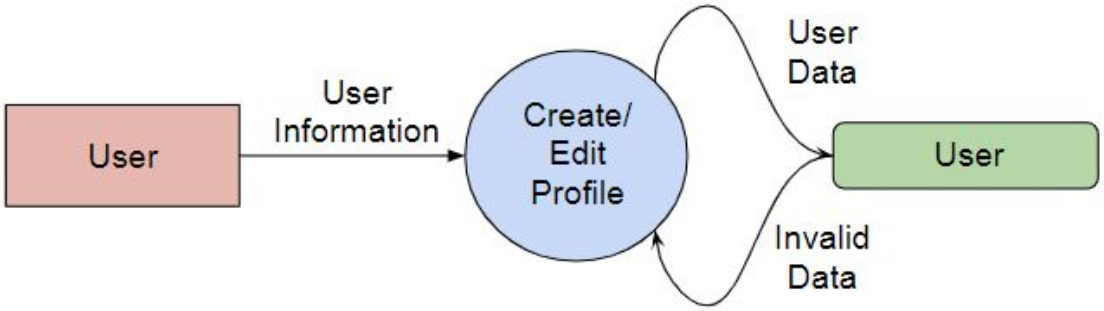
IC Card IC Name: create_user_profile

Description: a new user creates a profile
Interaction Pattern:

By Myself no Interacton

My Task: create profile, check against database, add to profile database
Time Critical Condition: none
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
  - This function will be available at the log-in screen. It will allow a user to create an account. This function will store the retrieved data in the user database
- Constraints
  - This function cannot create a guest account. This function will not verify the user has entered the correct information.



Create Dog

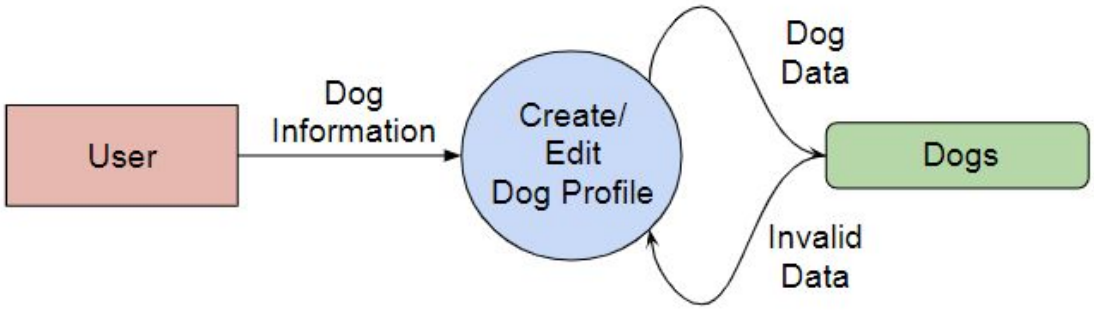
- IC Card

IC Card IC Name: create_dog_profile

Description: a user adds a dog to the dog database
Interaction Pattern:

By Myself with Interaction
My Task: create dog profile, add to database
Time Critical Condition:
Name of Other IC: edit_user_profile
Message to Other IC: add dog
Other IC's Task: edit profile with new dog
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
    - This function will occur inside the user's profile. It will allow them to create a profile for a dog they own. It will contain information other users may find useful to know before bringing their dog to the same park, such as the dog's name, temperament, size, etc.
- Constraints
    - This function will not be able to check for duplicate profiles and it will not be able to verify the given information about the dogs.



Edit User

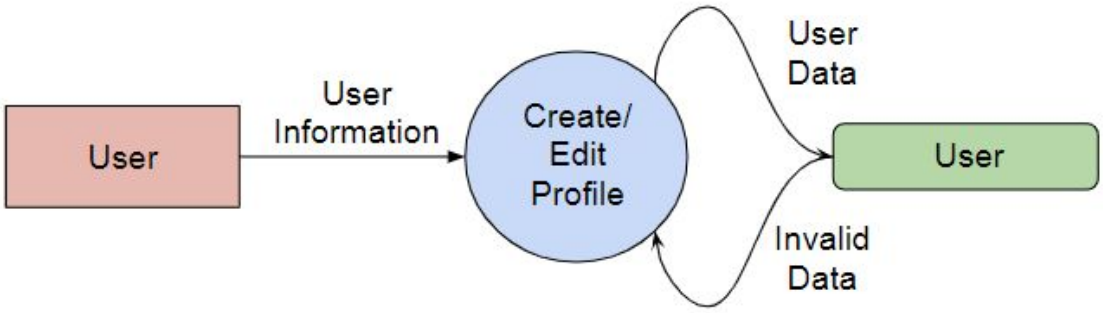
- IC Card

IC Card IC Name: edit_user_profile

Description: modifies a user profile
Interaction Pattern:

By Myself no Interacton
My Task: edit profile
Time Critical Condition: none
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
    - Users will be able to edit their profile. From their profile page, or when creating a new profile, they will be able to enter their name, change their password, change their profile picture, change the dogs associated with their profile and change their favorite dog park. This information will then be updated in the database, and the change will be reflected on the user's profile.
- Constraints
    - Current location and location history will not be stored, nor will contact information. Therefore, the users will have no ability to add or change this information. Users will also not be able to change the unique identifier (UID) associated with their account.



Edit Dog

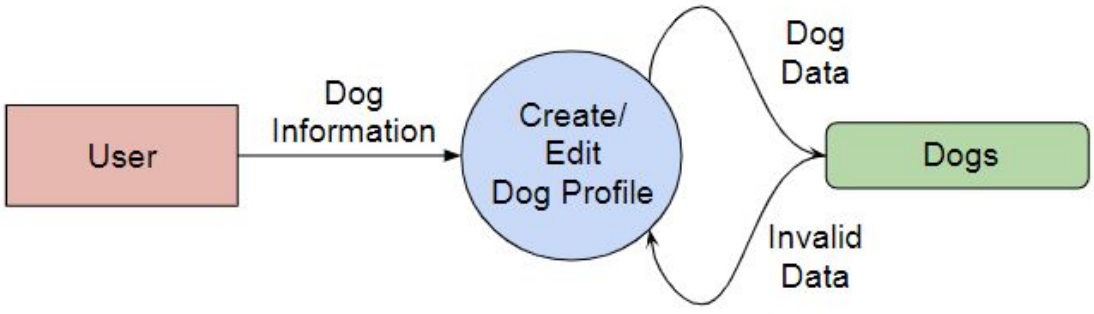
- IC Card

IC Card IC Name: edit_dog_profile

Description: modifies a dog profile
Interaction Pattern:

By Myself with Interaction
My Task: edit profile
Time Critical Condition: none
Name of Other IC: edit_user_profile
Message to Other IC: update with edited dog
Other IC's Task: edit profile with edited dog
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
    - Users will be able to edit their dogs' profiles. From the dog's profile page, or when adding a new dog profile, users will be able to edit the dog's name, profile picture, breed, size, age, temperament, activity level, volume level, and additional notes about the dog. This information will then be updated in the database, and the change will be reflected on the user's profile.
- Constraints
    - Dog temperament, activity level, and volume level will only be stored on a sliding scale of 1 to 10. No additional information will be stored for those categories, so users will not be able to describe their dog's temperament, unless they use the notes section. Users will not be able to change the unique identifier (UID) associated with each dog.



View User

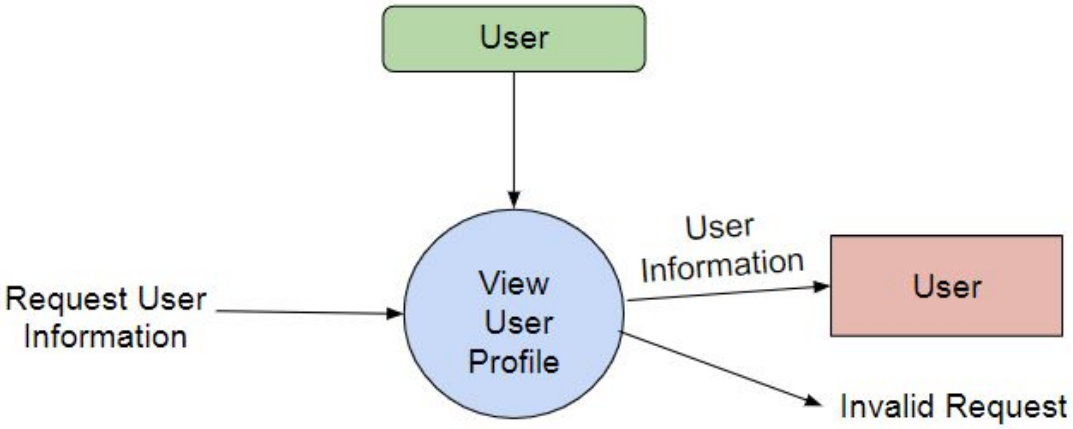
- IC Card

IC Card IC Name: view_user_profile

Description: displays user profile
Interaction Pattern:

By Myself no Interacton
My Task: display user info
Time Critical Condition: fails after 30 seconds
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
    - Users will be able to visit a page displaying their profile. The information on this page will include their username, links to dog profiles owned by them, and a list of scheduled park visits. From this page, the users will also be able to logout, change password, add/manage a dog profile, view parks, or schedule/remove a park visit.
- Constraints
    - This function will not be able to link to any other user profiles.



View Dog

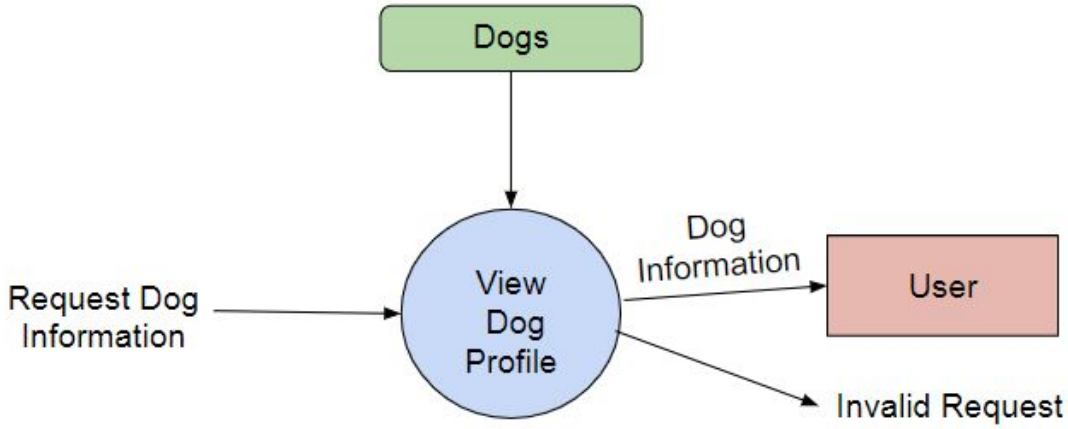
- IC Card

IC Card IC Name: view_dog_profile

Description: displays dog profile
Interaction Pattern:

By Myself no Interacton
My Task: display dog info
Time Critical Condition: timeout after 30 seconds
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
    - Users will be able to visit a page displaying profiles for each of the dogs they own. The information on this page will include the dog's name, picture, breed, size, age, temperament and other relevant data. This page will also display any upcoming park visits the owner has scheduled.
- Constraints
    - This function will only be linked to a single owner profile; multiple users cannot claim the same dog. Users will only be able to create 16 dog profiles.



View Park

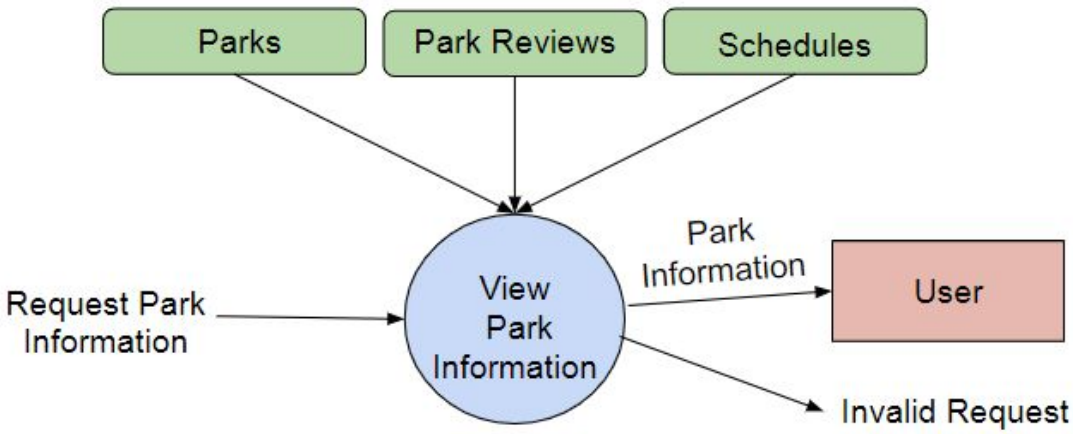
- IC Card

IC Card IC Name: view_park

Description: displays park information
Interaction Pattern:

By Myself no Interacton
My Task: display park info
Time Critical Condition: 30 second timeout
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process Narrative
    - Users will be able to access a page that displays information regarding the selected park. Information on this page will include the park's name, location, reviews, and all dogs scheduled to be at that park. The page will also allow the user to review the park.
- Constraints
    - This function will not be linked to any other parks.
    - The user must be logged in to view the form to review the park.



Review Park

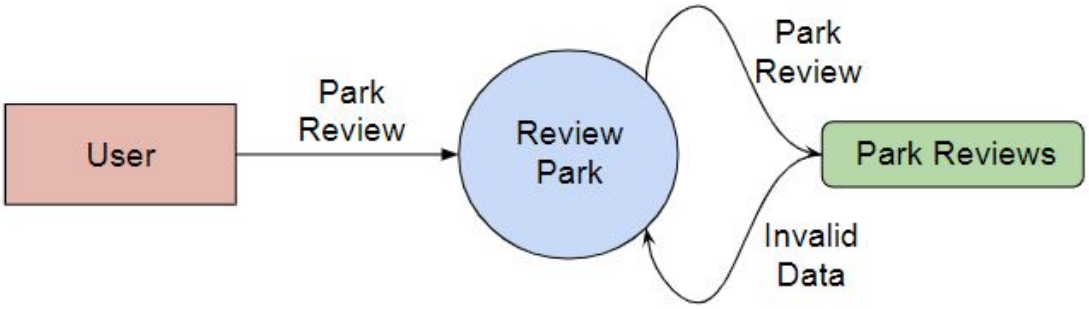
- IC Card

IC Card IC Name: review_park

Description: allows a user to review a park
Interaction Pattern:

By Myself no Interacton
My Task: updates review database with review information
Time Critical Condition: none
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process Narrative
    - This function will occur when the user views a park. Upon viewing the park's information window the user will have the option to leave a review of the park based on previous visits. It will ask for a text explanation as well as a numerical score.
- Constraints
    - The user must be logged in to review a park



Schedule Visit

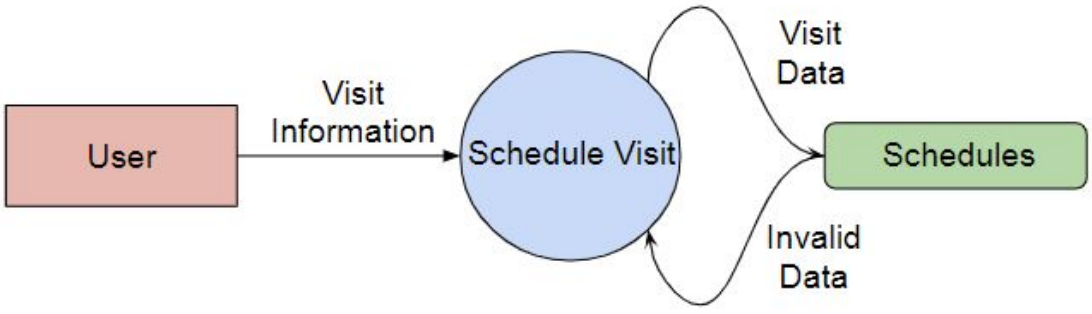
- IC Card

IC Card IC Name: schedule_visit

Description: allows the user to schedule a park visit
Interaction Pattern:

By Myself no Interacton
My Task: update schedule database with new schedule
Time Critical Condition: none
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
    - This function will occur once the user has viewed a park. Upon viewing the park's information window the user will have the option to schedule a visit at that park. It will then require the user to pick a date from a calendar, a time slot for that day, and the profiles of whatever dogs they plan on bringing with them.
- Constraints
    - The function will not check if the park is closed during a date or time they choose.



Log In

- IC Card

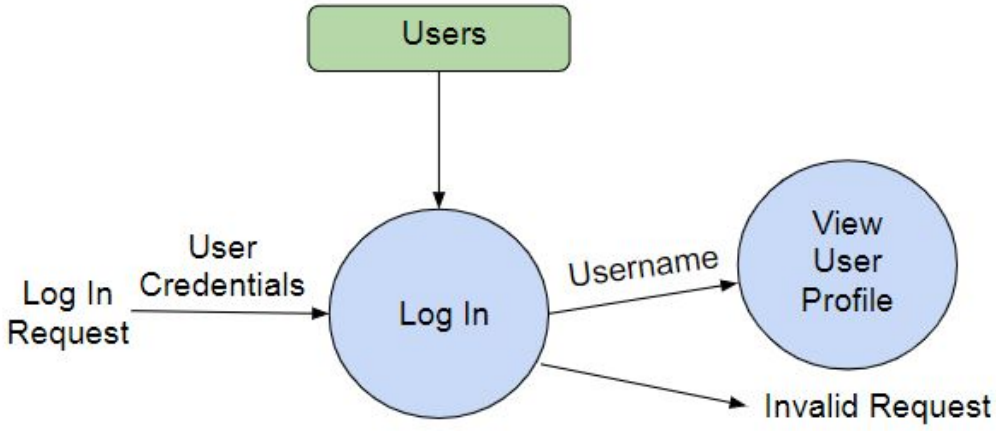
IC Card IC Name: log_in

Description: signs a user into their account
Interaction Pattern:

By Myself no Interacton

My Task: sign in
Time Critical Condition: 30 second timeout
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
    - This function will occur as the webpage is first loaded. If the user does not log in, it will be available via a button on screen. The function will allow the user to log in to their account. It will check entered username and password against a database, and allow login if they match
- Constraints
    - This program will not enforce a log in, users can use the site without being logged in.



Log Out

- IC Card

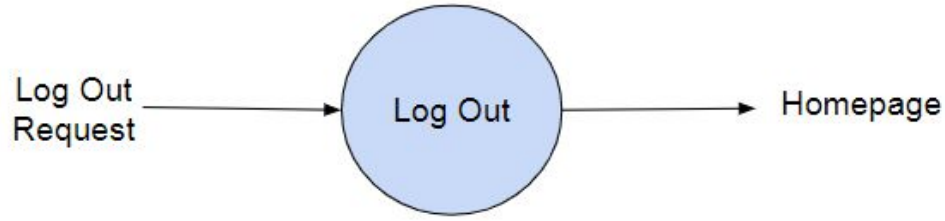
IC Card IC Name: log_out

Description: Sign user out of account
Interaction Pattern:

By Myself no Interacton

My Task: sign out
Time Critical Condition: none
Name of Other IC: none
Message to Other IC: none
Other IC's Task: none
Card 1of 1 (If necessary please use several IC cards to describe an IC)

- Process narrative
    - This function will log the user out of the application. The user will click on the logout button, with a location to be determined, and it will sign the user out of the application.
- Constraints
    - Must be logged in



4. Performance Requirements

Bark-in will be hosted on a server capable of storing 1,000 user profiles and 10,000 dog profiles in the databases. The park database will host information of 200 parks in the Pittsburgh area. The server will be able to handle 1,000 concurrent users, and the application will be able to perform queries to the server to update information. Queries from the client will not take more than 1000ms and will timeout if the deadline is missed.

5. Exception Conditions and Handling

For the storage of parks, users profiles and dog profiles, this will be limited by the hardware available to host the website. Therefore, this will be a hard limit until the hardware can be upgraded, and will be addressed as the site grows. Until additional capacity can be added, additional requests to add profiles will result in an error. Concurrent users will similarly be limited by hardware, but will only cause exceptions at high-traffic times, and will return to normal operation when traffic subsides.. If the limit is exceeded, users will be shown an error informing them that the server is busy, and to try again in a few minutes. Requests may also take longer to process during these times. If this becomes a pervasive issues, hardware can be upgraded for increased capacity.

6. Implementation Priorities

If time constraints are an issue, the following core functionality will be implemented first:

- Profiles for users and dogs with owners linked to their dogs.
    - Dog profiles will feature size, breed, activity level, and aggressiveness indicators.
    - Users will be able to view other's profiles and dogs
- Scheduling functionality for dogs at parks
- Information on parks such as business, cleanliness, and a list of dogs that are at the park.

When this core functionality is fulfilled, development will begin on some less important functions following an incremental development model. These include:

- Real time check in for dogs at a park. This functionality allows the user to not schedule a visit beforehand, simply check in to the park that they are currently at
- Incentive to use the app, a user profile badge or level that indicates activity level.

7. Foreseeable Modifications and Enhancements

Real-time check-in will realistically only be feasible once a mobile app for this product is created. People would not be bringing their laptops to a park to check in, even if there was an internet connection for them to use. Designing a mobile-friendly web client could serve as a stopgap, and user location could be accessed through the web browser. Incentives to participate with the app will be important in order to discourage people from simply looking at what parks are popular and what dogs are there, without contributing to the information themselves. This could be accomplished through things like badges on their account or a user level to indicate how much they have contributed to the app.

8. Acceptance Criteria

Software

( ) Software configuration is complete, documents include all necessary criteria: requirements, design, test plan, test results, source code, user manual, maintenance procedure.

( ) Databases are fully configured and ready to be used

( ) Web app is capable of running on any modern browser without additional add-ons, extensions, plug-ins, etc.

( ) Server software will be able to be executed on a Windows PC without additional plug-ins. Software can be installed from a flash stick or website via download of an executable "install.exe"

Document

( ) Documents are labelled with appropriate sections divided via labelled dividers

( ) Changes made after reviews are well indicated with yellow tabs, this document includes a revision history

( ) Cross references throughout the document. References are labelled with page numbers and sections

Presentation

( ) Powerpoint presentation highlighting functionality of the web app

( ) All project members are present and contributions are clearly explained

Demo

( ) Start

( ) Verify user login, for multi-user demo, users are logged in beforehand

( ) User can use all major functions

( ) User can go through a scenario

( ) Display results of scenario

( ) Store user edited info in database

( ) Verify admin login for a new or existing database

( ) Display all user information to admin in existing database, admin is capable of removing and adding users and performing basic maintenance

( ) Provide a learning for users to learn how to use the basics

( ) Show how to maintain the web app: find source code, make a small modification and recompile

9. Sources of Information

1. Application and the backend databases will be implemented using SQLite with Django framework; documentation can be found at:
    a. sqlite.org/docs.html
    b. docs.djangoproject.com
2. Front-end will be implemented with Bootstrap/HTML/Javascript; documentation can be found at:
    a. getbootstrap.com/docs/
    b. devdocs.io/html/
    c. devdocs.io/javascript/

Revision History

Date	Description	Person	Comments
10/15/2019	Requirement Spec Completed	Everyone	Initial version
10/2/2019	Added labor (man-hours) cost table	Everyone	Added after milestone 2 turn-in

11/13/2019	Roles changed into a matrix format	Ian Lundberg	Originally list of roles, changed to match project requirements
11/13/2019	Added administrator actor to use case diagram and enclosed use cases with box	Ian Lundberg	Was missing administrator actor and was not enclosed in a box
11/19/2019	Added test coverage and updated test input	Ian Lundberg	Was missing test coverage for WhiteBox tests and test table input was formatted incorrectly
12/2/2019	BlackBox Test #16 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters for a username
12/2/2019	BlackBox Test #18 resolved	Connor Kalina	accepting allowance of emojis in username as a feature
12/2/2019	BlackBox Test #22 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters for a username
12/2/2019	BlackBox Test #25 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters in a user's biography
12/2/2019	BlackBox Tests #26-28 resolved	Connor Kalina	Favorite park feature implemented
12/2/2019	BlackBox Test #31 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters for a dog name
12/2/2019	BlackBox Test #32-33 resolved	Connor Kalina	Dog profile photo feature implemented
12/2/2019	BlackBox Test #35	Connor Kalina	Fixed a bug not

	resolved		allowing the correct number of characters for dog breed
12/2/2019	BlockBox Test #36-37 resolved	Connor Kalina	Fixed a bug that was allowing illegal input into dog size
12/2/2019	BlackBox Test #44 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters in the notes section of a dog profile
12/2/2019	BlackBox Test #45-60 resolved	Connor Kalina	Edit dog profile feature was implemented
12/2/2019	BlackBox Test #62-64 resolved	Connor Kalina	Page was not showing an error message on illegal username
12/2/2019	BlackBox Test #74 75 resolved	Connor Kalina	Fixed a bug that was allowing park reviews with a star rating below 1 or above 5
12/2/2019	BlackBox Test #87 resolved	Connor Kalina	Handled an exception that was being thrown when a user would type illegal input into the date field when scheduling an event
12/2/2019	BlackBox Test #88 resolved	Connor Kalina	Handled an exception that was being thrown when a user would type illegal input into the time field when scheduling an event
12/2/2019	BlackBox Test #95 resolved	Connor Kalina	Implemented edit dog profile feature,

			integration test was failing.

User manual

Product overview

Bark-in is a social networking application which allows dog owners to schedule visits to parks by first viewing what other dogs will be there at that time. Users will be able to create profiles for their dogs which include pictures and traits about them, including breed, size, temperament, age, vocalization, etc. Once a user

schedules a visit to the park, other users will be able to see their dog's profiles for that time and location. This will let users view how busy a park will be as well as allow they to see if they believe their dog will get along with the other dogs there at that time. The application will also allow users to rate and review parks on qualities such cleanliness and maintenance.

When users create an account, they will be able to create up to 16 profiles for their dogs. The application's back end will store a database for users and a separate database for dogs and the two will be linked in one to many relationships. Another database will store information and pictures for local dog parks. This parks database will be linked to another database storing the ratings and reviews for the parks. Lastly, a fifth database will be needed to store the users' scheduled visits. When a user views a park, they will be able to choose a date and time. The application will display the park's information as well as the profiles of the dogs scheduled to be there at the time. If the user finds a park and time that they find suitable, they will be able to schedule a visit and their dogs' profiles will be viewable by other users.

As well as this core functionality, in the future Bark-in will allow users to generate friends lists. These lists will allow users to message each other and plan visits and playdates. Users will also be able to enter information about other dogs at the park that are not part of the Bark-in community, giving other users a better picture of the parks. Also, using the data gathered from users, the application will be able to plot trends at parks allowing it to predict how busy parks will be in the future.

GETTING STARTED

2.1 Log In

Upon reaching Bark-in's main page, users will immediately be prompted to log in with a username and password or create an account.

2.2 Help Mode

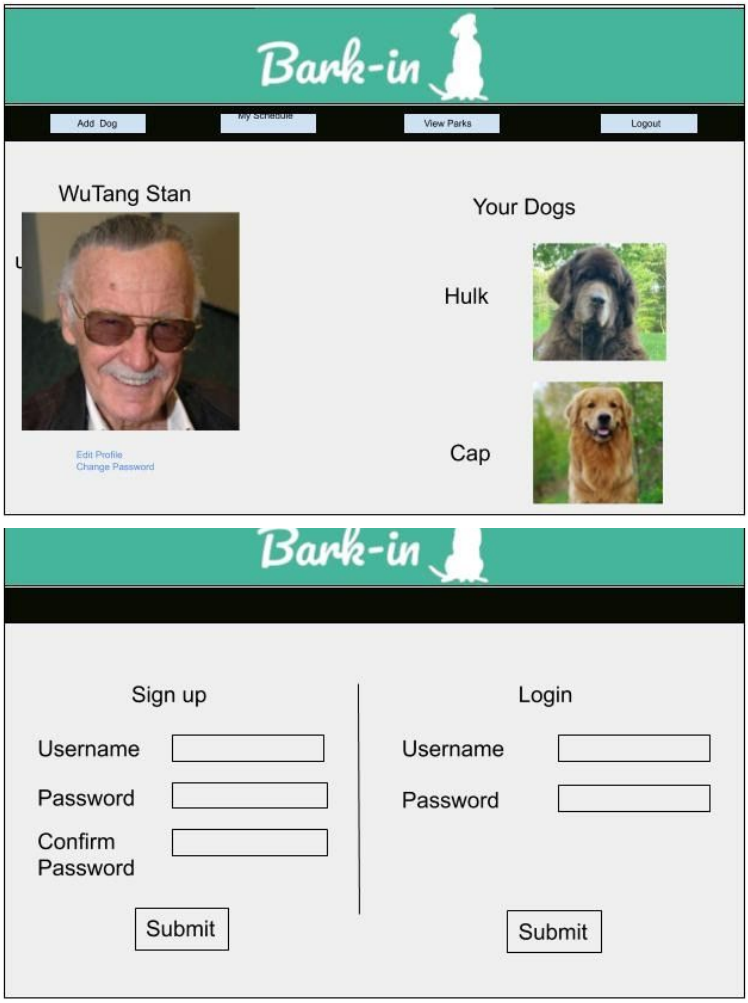
On the dashboard, there will be a *help* button that will redirect the user to a web page with information on website features and how to use them.

2.3 Sample Run

To schedule a dog park visit with Bark-in, the user must complete the following steps: First, the user will arrive on the homepage and will need to either log in or sign up by clicking the “Sign up/Log in” button on the top left of the page. Second, once the user has logged in for the first time they will be presented with their profile where they can add information about themselves by clicking the “Edit Profile” hyperlink under their profile picture, and create profiles for their dogs by clicking the “Add Dog” button on the top left of the page. Third, after the user is finished setting up their profile and profile(s) for their dogs they have to click the “View Parks” button at the top of the page. This will take them to another web page where they will see a map populated with nearby dog parks and a list of them on the left side of the page. Fourth, the user must pick a dog park from either the list on the left side or by clicking on a pin inside the map view. This will take them to another web page where they can schedule a visit. Finally, the user must pick a date from the calendar view on the left and input an arrival and departure time on the right. Once this is done all the user has to do is click “Submit” and their visit is scheduled. See the steps below for a more detailed description:

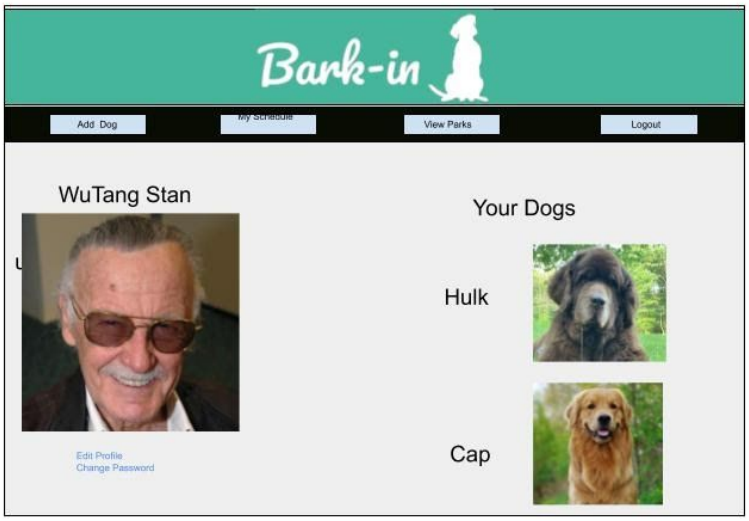
1. At the home page pictured below, click the “Sign up/Login” button.

2. Upon reaching the log in/sign up page, click into the textbox titled "Username" and type your username. Then click into the textbox titled "Password" and type your password. Complete the process by clicking "Submit".
3. You should now see your profile screen as pictured below. From here you can edit your profile. To create a profile for your dog click the "Add Dog" button in the top bar.



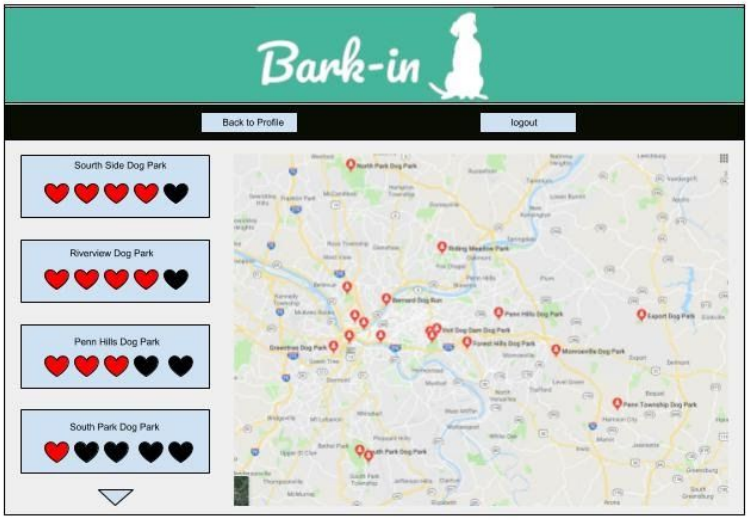
4. Upon reaching the page for adding a new dog you must enter the name of the dog, its age, breed, size, vocalization level, and a picture. Once this information has been filled out click the "Submit" button.

5. You will be returned to the user profile page pictured below. Now to schedule a visit press the “View Parks” button in the top bar of the page.



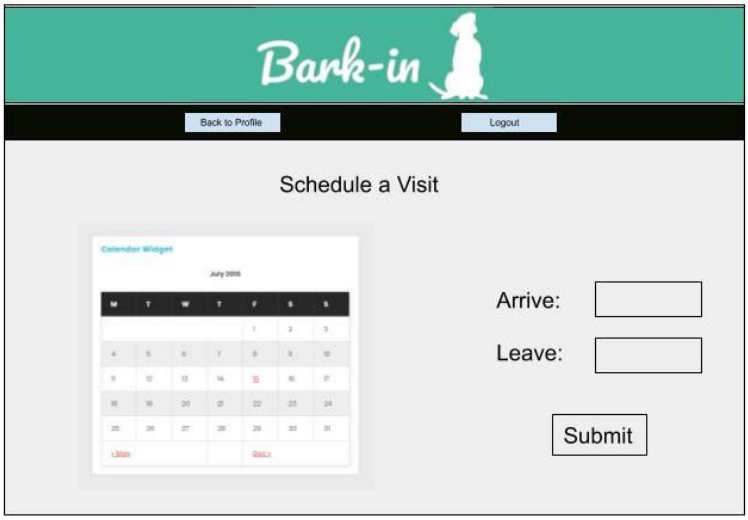
6. Upon reaching the view parks page you will see a map containing pins of the dog parks near your location and a list on the left side of the page. Click a park

from the list or a pin on the map. This will take you to the page where you can



schedule a visit.

7. To schedule a visit you must pick the date you plan on visiting from a calendar on the left and a time slot on the right: by entering the time you plan on arriving and the time you plan on leaving. Finally, all you have to do is click on the “Submit” button to confirm your scheduled time.



Modes of Operation

Upon reaching Bark-in, a user is required to log in or create an account. If the user is creating a profile they will need to enter information about themselves, such as a biography and contact information. The user will also have the option to create and add dog profiles during account creation or from viewing his/her profile. After creating an account or logging in, the user will see a map populated with nearby dog parks. The user can then search for a dog park using the address bar on the top of the web page by selecting one from a list of nearby parks on the left side of the map, or by clicking on a dog park on the map, represented by a pin. After selecting a dog park, a window will popup with information about the park and a button to schedule a visit. If the user decides to schedule a visit they will be asked to select a date from a calendar and the range of time they plan on spending there. After selecting a time frame the user will be shown a list of other dogs that are already scheduled to be there. The user can then view the profiles of the other dogs and/or confirm his/her scheduled visit.

Advanced Features

Real-time check-in will realistically only be feasible once a mobile app for this product is created. People would not be bringing their laptops to a park to check in, even if there was an internet connection for them to use. Designing a mobile-friendly web client could serve as a stopgap, and user location could be accessed through the web browser. Incentives to participate with the app will be important in order to discourage people from simply looking at what parks are popular and what dogs are there, without contributing to the information themselves. This could be accomplished through things like badges on their account or a user level to indicate how much they have contributed to the app.

Installation

Program is accessed through a hosted web page by navigating to https://bark-in.herokuapp.com in a web browser.

Maintenance Procedure

The maintenance procedure for the application will be accomplished via the following categories:

1. Corrective Maintenance
    a. Reactive modification of the application to correct discovered problems.
2. Adaptive Maintenance
    a. Modification of the application to keep the application usable in a changed or changing environment.
3. Perfective Maintenance
    a. Modification of the application to improve performance or maintainability.
4. Preventative Maintenance
    a. Modification of the application to detect and correct latent faults in the software product before they become effective faults.

Version control of the application will be used via a git repository on github. By using separate branches and cloning the repository, the maintenance or development team

will be able to work on the following maintenance categories above independently from each other and the live version of the application. Once the changes to the application are made the application will be put into maintenance mode, disabling use for the general user base, while the maintenance branch is merged with the live branch. integration and system testing will then be performed by the SQA team, and assuming no problems, the application will be brought out of maintenance mode and made available again to the general user base and all changes will be documented in the revision history.

Revision History

Date	Description	Person	Comments
10/15/2019	Requirement Spec Completed	Everyone	Initial version
10/2/2019	Added labor (man-hours) cost table	Everyone	Added after milestone 2 turn-in
11/13/2019	Roles changed into a matrix format	Ian Lundberg	Originally list of roles, changed to match project requirements
11/13/2019	Added administrator actor to use case diagram and enclosed use cases with box	Ian Lundberg	Was missing administrator actor and was not enclosed in a box
11/19/2019	Added test coverage and updated test input	Ian Lundberg	Was missing test coverage for WhiteBox tests and test table input was formatted incorrectly

12/2/2019	BlackBox Test #16 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters for a username
12/2/2019	BlackBox Test #18 resolved	Connor Kalina	accepting allowance of emojis in username as a feature
12/2/2019	BlackBox Test #22 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters for a username
12/2/2019	BlackBox Test #25 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters in a user's biography
12/2/2019	BlackBox Tests #26-28 resolved	Connor Kalina	Favorite park feature implemented
12/2/2019	BlackBox Test #31 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters for a dog name
12/2/2019	BlackBox Test #32-33 resolved	Connor Kalina	Dog profile photo feature implemented
12/2/2019	BlackBox Test #35 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters for dog breed
12/2/2019	BlockBox Test #36-37 resolved	Connor Kalina	Fixed a bug that was allowing illegal input into dog size
12/2/2019	BlackBox Test #44 resolved	Connor Kalina	Fixed a bug not allowing the correct number of characters in the notes section of a dog profile
12/2/2019	BlackBox Test #45-60	Connor Kalina	Edit dog profile

	resolved		feature was implemented
12/2/2019	BlackBox Test #62-64 resolved	Connor Kalina	Page was not showing an error message on illegal username
12/2/2019	BlackBox Test #74 75 resolved	Connor Kalina	Fixed a bug that was allowing park reviews with a star rating below 1 or above 5
12/2/2019	BlackBox Test #87 resolved	Connor Kalina	Handled an exception that was being thrown when a user would type illegal input into the date field when scheduling an event
12/2/2019	BlackBox Test #88 resolved	Connor Kalina	Handled an exception that was being thrown when a user would type illegal input into the time field when scheduling an event
12/2/2019	BlackBox Test #95 resolved	Connor Kalina	Implemented edit dog profile feature, integration test was failing.

Software Process Generator

Based upon your project's needs, please select appropriate value for each parameter.
If you do not use a parameter, you need not select any value for that parameter.
In the calculation of optimal software process model, that parameter will be ignored.

Parameter	Value
Early Functionality	○ Iteratively introduce features ◉ Only produce final product
Feature Adaptation	○ Impossible ◉ Flexible
User Involvement	(Initially) ○ 0% ○ 10% ○ 20% ○ 30% ○ 40% ○ 50% ○ 60% ◉ 70% ○ 80% ○ 90% ○ 100% (Frequent feedback)
Documentation	○ Not produced ◉ Produced
Experienced Team	○ Requested ◉ Not Required
Model Type	(Linear) ○ 0% ○ 10% ○ 20% ○ 30% ◉ 40% ○ 50% ○ 60% ○ 70% ○ 80% ○ 90% ○ 100% (Iterative)
Planning and Scheduling	○ Upfront ◉ Continuous
Risk Mitigation	○ Yes ◉ No
Project Size	(Small) ○ 0% ○ 10% ◉ 20% ○ 30% ○ 40% ○ 50% ○ 60% ○ 70% ○ 80% ○ 90% ○ 100% (Large)
Prototype	○ Used ◉ Not Used
CrossPlatform	◉ No ○ Yes

SubmitParameters

waterfall : 15

incremental : 11 (optimal)

spiral : 18

xp : 14

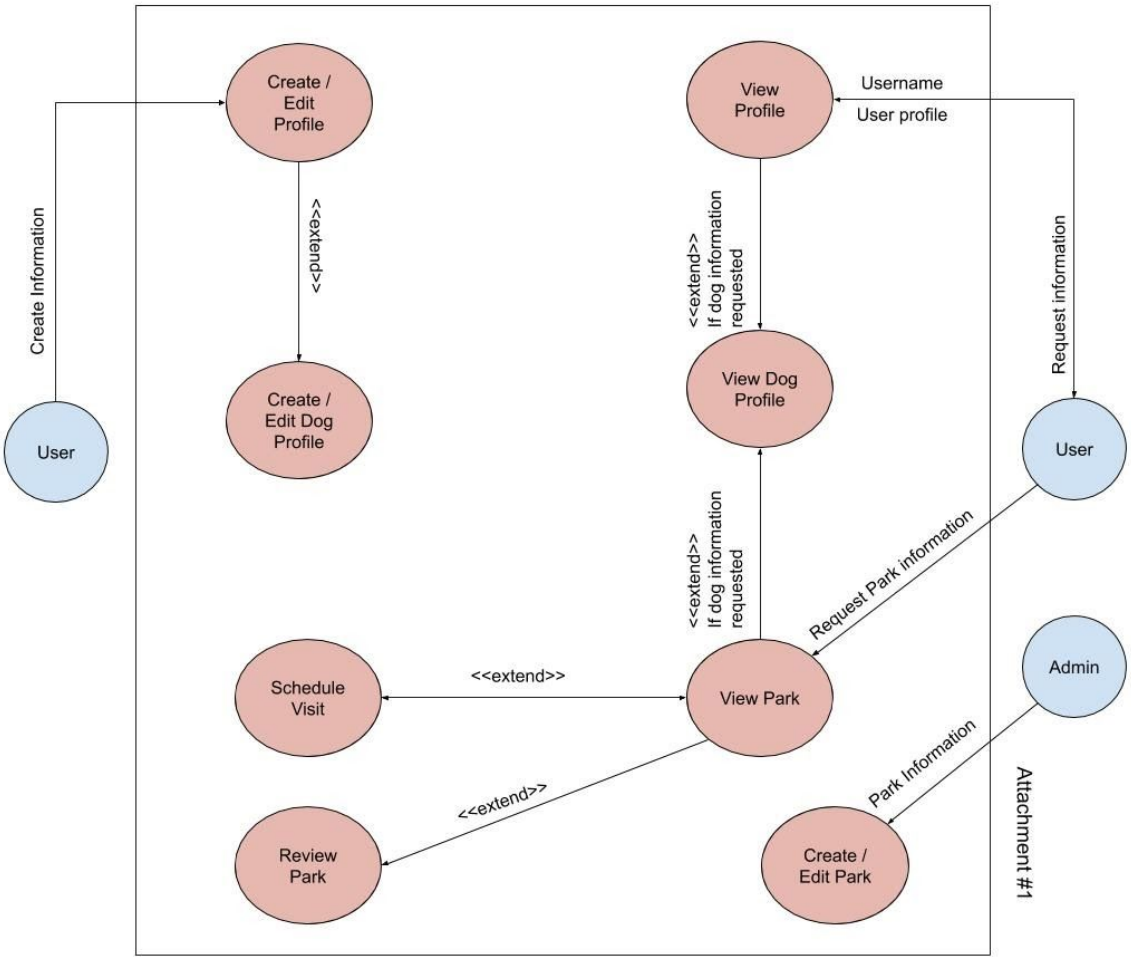
scrum : 14

crossplatform : 15

Object-Oriented Analysis

1.0System Overview

1.1 Use Case Diagram



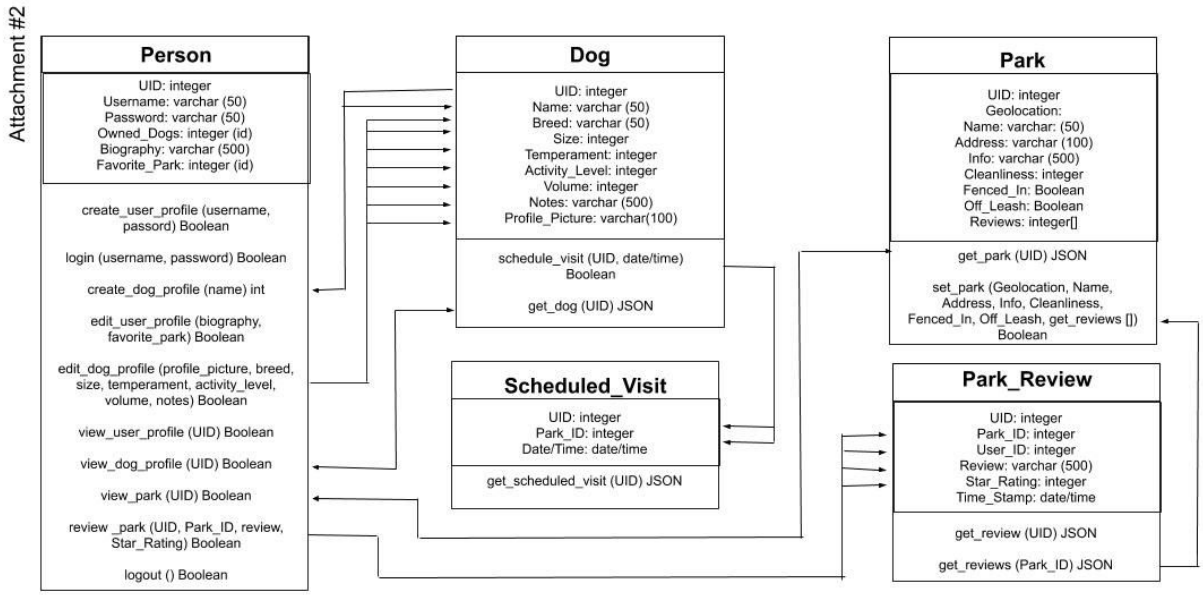
2.0 Class Model

2.1 The Classes

1. Person(UID, username, password, owned dogs, bio, favorite dog park)

2. Dog(UID, name, profile picture, breed, size, temperament, activity level, volume, notes
3. Park(UID, geolocation, name, info, address, cleanliness, fenced in, off leash, reviews)
4. Park_Review(UID, user id, review, star rating, time stamp(time))
5. Scheduled_Event(UID, park id, dog id, time frame, date)

2.2 The Class Diagram

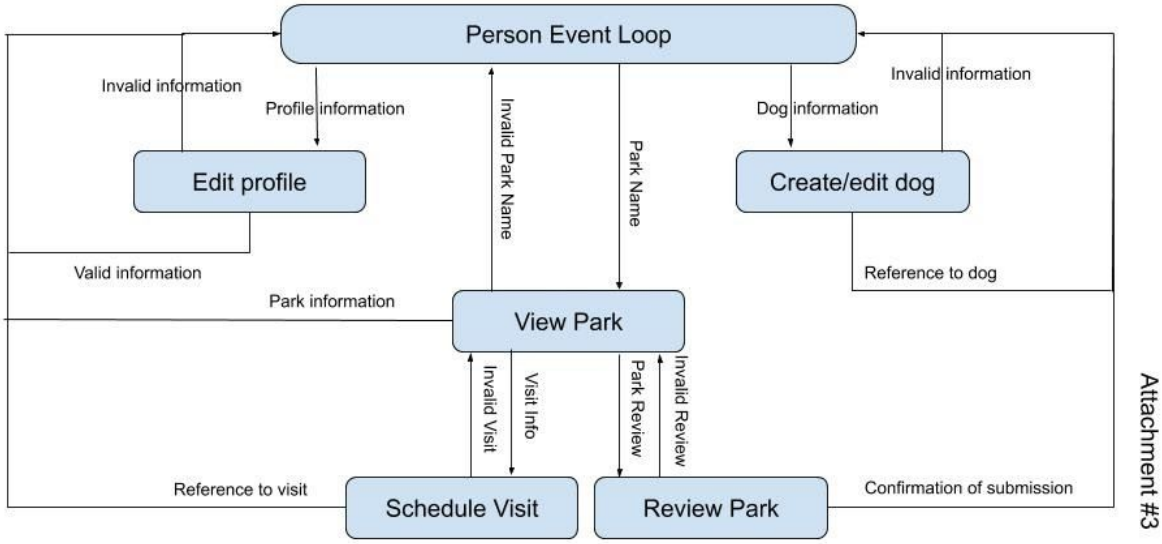


3.0 The Dynamic Model

3.1 The Scenarios

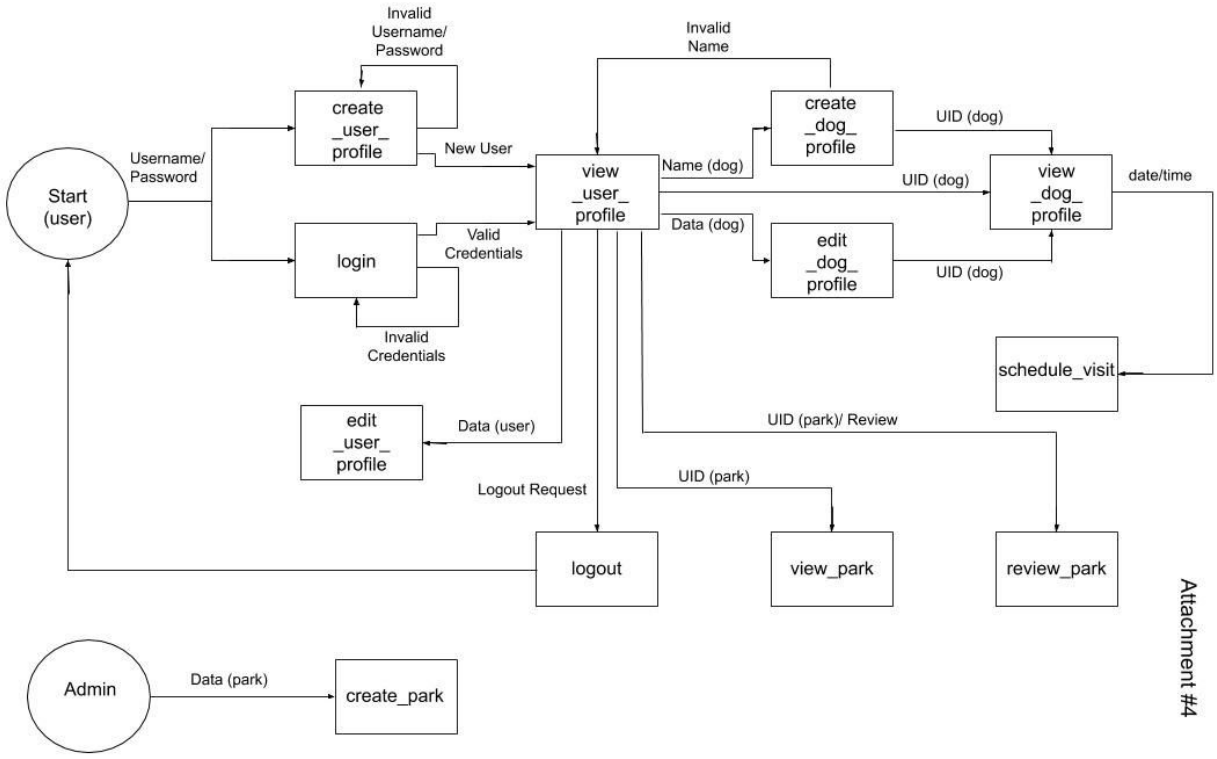
1. Person
    a. A person edits their profile.
    b. A person adds a new dog.
    c. A person edits a dog profile.
    d. A person views a park.
    e. A person schedules their dog to visit a park
    f. A person leaves a review of the park.

3.2 The State Diagrams



Dynamic Model

4.0 The Functional Model



Functional Model

Object-Oriented Design

Modules

Main::log_in

- ➔ Module Type: Method
- ➔ Return Type: Person
- ➔ Input Arguments: username, password
- ➔ Output Arguments: none
- ➔ Error Messages: if not a valid username or password
- ➔ Files accessed: user database
- ➔ Files changed: none
- ➔ Modules called: none
- ➔ Narrative: This function will occur as the webpage is first loaded. If the user does not log in, it will be available via a button on screen. The function will allow the user to log in to their account. It will check entered username and password against a database, and allow login if they match.

- **Main::log_out**

- Module Type: Method
- Return Type: void
- Input Arguments: none
- Output Arguments: none
- Error Messages: if not currently logged in.
- Files accessed: none
- Files changed: none
- Modules called: none
- Narrative: This function will log the user out of the application. The user will click on the logout button, with a location to be determined, and it will sign the user out of the application.

Main::create_profile

- Module Type: Method
- Return Type: Person
- Input Arguments: user data
- Output Arguments: none
- Error Messages: if invalid user data entered
- Files accessed: user database
- Files changed: user database
- Modules called: none
- Narrative: This function will be available at the log-in screen. It will allow a user to create an account. This function will store the retrieved data in the user database

Main::view_user_profile

- ➔ Module Type: Method
- ➔ Return Type: void
- ➔ Input Arguments: username
- ➔ Output Arguments: none
- ➔ Error Messages: if username is invalid
- ➔ Files accessed: user database
- ➔ Files changed: none
- ➔ Modules called: none
- ➔ Narrative: Users will be able to visit a page displaying their profile. The information on this page will include their username, links to dog profiles owned by them, and a list of scheduled park visits. From this page, the users will also be able to logout, change password, add/manage a dog profile, view parks, or schedule/remove a park visit.

Main::view_park

- ➔ Module Type: Method
- ➔ Return Type: none
- ➔ Input Arguments: park data
- ➔ Output Arguments: none
- ➔ Error Messages: if invalid park data entered
- ➔ Files accessed: parks database
- ➔ Files changed: none
- ➔ Modules called: Park::review_park, Park::schedule_visit
- ➔ Narrative: Users will be able to access a page that displays information regarding the selected park. Information on this page will include the park's name, location, reviews, and all dogs scheduled to be at that park. The page will also allow the user to review the park.

Person::edit_profile

- ➔ Module Type: Method
- ➔ Return Type: void
- ➔ Input Arguments: data to be changed
- ➔ Output Arguments: none
- ➔ Error Messages: if invalid user data entered
- ➔ Files accessed: user database
- ➔ Files changed: user database
- ➔ Modules called: none
- ➔ Narrative: Users will be able to edit their profile. From their profile page, or when creating a new profile, they will be able to enter their name, change their password, change their profile picture, change the dogs associated with their profile and change their favorite dog park. This information will then be updated in the database, and the change will be reflected on the user's profile.

Person::create_dog

- ➔ Module Type: Method
- ➔ Return Type: Dog
- ➔ Input Arguments: dog data
- ➔ Output Arguments: none
- ➔ Error Messages: if invalid dog data entered
- ➔ Files accessed: dog database
- ➔ Files changed: dog database
- ➔ Modules called: none
- ➔ Narrative: This function will occur inside the user's profile. It will allow them to create a profile for a dog they own. It will contain information other users may find useful to know before bringing their dog to the same park, such as the dog's name, temperament, size, etc.

Dog::view_dog

- ➔ Module Type: Method
- ➔ Return Type: void
- ➔ Input Arguments: data to be changed
- ➔ Output Arguments: none
- ➔ Error Messages: if no dog profiles
- ➔ Files accessed: dog database
- ➔ Files changed: none
- ➔ Modules called: none
- ➔ Narrative: Users will be able to visit a page displaying profiles for each of the dogs they own. The information on this page will include the dog's name, picture, breed, size, age, temperament and other relevant data. This page will also display any upcoming park visits the owner has scheduled.

Dog::edit_dog

- ➔ Module Type: Method
- ➔ Return Type: void
- ➔ Input Arguments: data to be changed
- ➔ Output Arguments: none
- ➔ Error Messages: if invalid dog data entered
- ➔ Files accessed: dog database
- ➔ Files changed: dog database
- ➔ Modules called: none
- ➔ Narrative: Users will be able to edit their dogs' profiles. From the dog's profile page, or when adding a new dog profile, users will be able to edit the dog's name, profile picture, breed, size, age, temperament, activity level, volume level, and additional notes about the dog. This information will then be updated in the database, and the change will be reflected on the user's profile.

Park::review_park

- ➔ Module Type: Method
- ➔ Return Type: void
- ➔ Input Arguments: park data
- ➔ Output Arguments: none
- ➔ Error Messages: if invalid park data entered
- ➔ Files accessed: park database
- ➔ Files changed: park database
- ➔ Modules called: none
- ➔ Narrative: This function will occur when the user views a park. Upon viewing the park's information window the user will have the option to leave a review of the park based on previous visits. It will ask for a text explanation as well as a numerical score.

Park::schedule_visit

- ➔ Module Type: Method
- ➔ Return Type: Scheduled_Event
- ➔ Input Arguments: schedule_event data
- ➔ Output Arguments: none
- ➔ Error Messages: if invalid schedule_visit data entered
- ➔ Files accessed: schedules database
- ➔ Files changed: schedules database
- ➔ Modules called: none
- ➔ Narrative: This function will occur once the user has viewed a park. Upon viewing the park's information window the user will have the option to schedule a visit at that park. It will then require the user to pick a date from a calendar, a time slot for that day, and the profiles of whatever dogs they plan on bringing with them.

Testing

1.0 Purpose

This test plan will test each unit of the Bark-In application, backend database structure, and front-end user interface, from both the user and administrative perspective. It will ensure all inputs work as planned. It will test for as many corner cases and illegal parameters as can reasonably be covered and handle them appropriately.

2.0 Unit Testing

2.1 Create Profile

Allows a user to create a profile that will allow him/her to login and access the application.

2.1.1 Equivalence Classes

2.1.1.1 username

- Legal
  - Not found in database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Already in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.01.1.1 password

- Legal
  - Alphanumeric characters
  - Symbols
  - Arbitrary length
- Illegal
  - N/A

2.1.2 Checklist

- ❏ Attempt to add user with legal name/legal password
- ❏ Attempt to add username already in database (legal password)
- ❏ Attempt to add user with name too long (legal password)
- ❏ Attempt to add user illegal name (illegal characters)
- ❏ Attempt to add user with illegal password (illegal characters)

2.1.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.2 Edit Profile

Allows a user to edit their username, password, bio, and favorite park.

2.2.1 Equivalence Classes

2.2.1.1 username

- Legal
  - Not found in database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Already in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.2.1.2 password

- Legal
  - Alphanumeric characters
  - Symbols
  - Arbitrary length
- Illegal
  - N/A

2.2.1.3 bio

- Legal
  - Less than 513 characters
  - Alphanumeric characters
  - Symbols
- Illegal
  - More than 512 characters

2.2.1.4 favorite park

- Legal
  - Arbitrary length
  - Alphanumeric characters
  - Already in park database
- Illegal
  - Non-alphanumeric characters
  - Not in park database

2.2.2 Checklist

- ❏ Attempt to change username to a new legal username
- ❏ Attempt to change username to one already in the database
- ❏ Attempt to change username to one with non-alphanumeric characters

- ❏ Attempt to change username to one longer than 150 characters
- ❏ Attempt to change password to a new legal password
- ❏ Attempt to change bio to a new legal bio
- ❏ Attempt to change bio to one with more than 512 characters
- ❏ Attempt to change favorite park to a new legal favorite park
- ❏ Attempt to change favorite park to one with non-alphanumeric characters
- ❏ Attempt to change favorite park to one not in the database

2.2.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.3 Login

Allows a person to log in as a specific user. Users must be logged into to edit their profile, add or change dogs, schedule time at parks, and review parks.

2.3.1 Equivalence Classes

2.3.1.1 username

- Legal
  - In database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Not found in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters in length
  - User already logged in
- Unregistered
  - Not found in database

2.3.1.1 password

- Legal
  - In database (with username)
  - Alphanumeric characters
  - Symbols
  - Arbitrary length
- Illegal
  - Not found in database

2.3.2 Checklist

- ❏ Attempt to login user with valid username/password pair
- ❏ Attempt to login user with valid username and valid password that don't match
- ❏ Attempt to login user that is already logged in

- ❏ Attempt to login user with username not in database
- ❏ Attempt to login user with password not in database
- ❏ Attempt to login user with username including invalid characters
- ❏ Attempt to login user with username longer than 150 characters

2.3.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.4 Logout

Allows a person to log out from their session.

2.4.1 Equivalence Classes

2.4.1.1 username

- Legal
  - In database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
  - Currently logged in
- Illegal
  - Not found in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters in length
  - Not currently logged in

2.4.2 Checklist

- ❏ Attempt to logout user with valid username who is logged in
- ❏ Attempt to logout user who is not logged in
- ❏ Attempt to logout username not in database
- ❏ Attempt to logout username with invalid characters
- ❏ Attempt to logout username with more than 150 characters

2.4.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.5 Create Dog

Allows a user to create a dog profile that will be attached to their profile and can be scheduled at a park.

2.5.1 Equivalence Classes

2.5.1.1 name

- Legal
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length

- Illegal
    - Non-alphanumeric characters excluding [_,@,+,., -]
    - Longer than 150 characters

2.5.1.2 picture

- Legal
    - Less than 10MB
    - PNG or JPEG format
- Illegal
    - More than 10MB
    - Formats other than PNG or JPEG

2.5.1.3 breed

- Legal
    - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
    - Less than 151 characters in length
- Illegal
    - Non-alphanumeric characters excluding [_,@,+,., -]
    - Longer than 150 characters

2.5.1.4 size

- Legal
    - Character
    - ‘S’, ‘M’, or ‘L’
- Illegal
    - Not a character
    - Character other than ‘S’, ‘M’, or ‘L’

2.5.1.5 temperament

- Legal
    - Integer greater than 0 and less than 11
- Illegal
    - Not an integer
    - Integer less than 1
    - Integer more than 10

2.5.1.6 activity level

- Legal
    - Integer greater than 0 and less than 11
- Illegal
    - Not an integer
    - Integer less than 1
    - Integer more than 10

2.5.1.7 volume

- Legal
    - Integer greater than 0 and less than 11
- Illegal

- Not an integer
- Integer less than 1
- Integer more than 10

2.5.1.8 notes

- Legal
  - Less than 513 characters
  - Alphanumeric characters
  - Symbols
- Illegal
  - More than 512 characters

2.5.2 Checklist

- ❏ Attempt to add dog with valid parameters
- ❏ Attempt to add dog with name including illegal characters
- ❏ Attempt to add dog with name longer than 150 characters
- ❏ Attempt to add dog with picture larger than 10MB
- ❏ Attempt to add dog with picture format other than PNG or JPEG
- ❏ Attempt to add dog with breed including illegal characters
- ❏ Attempt to add dog with breed longer than 150 characters
- ❏ Attempt to add dog with size not a single character
- ❏ Attempt to add dog with size using characters other than S, M, or L
- ❏ Attempt to add dog with temperament less than 1
- ❏ Attempt to add dog with temperament greater than 10
- ❏ Attempt to add dog with activity level less than 1
- ❏ Attempt to add dog with activity level greater than 10
- ❏ Attempt to add dog with volume less than 1
- ❏ Attempt to add dog with volume greater than 10
- ❏ Attempt to add dog with notes more than 512 characters

2.5.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.6 Edit Dog

Allows a user to edit one of the dogs attached to their profile.

2.6.1 Equivalence Classes

2.6.1.1 name

- Legal
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal

- Non-alphanumeric characters excluding [_,@,+,., -]
- Longer than 150 characters

2.6.1.2 picture

- Legal
  - Less than 10MB
  - PNG or JPEG format
- Illegal
  - More than 10MB
  - Formats other than PNG or JPEG

2.6.1.3 breed

- Legal
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.6.1.4 size

- Legal
  - Character
  - ‘S’, ‘M’, or ‘L’
- Illegal
  - Not a character
  - Character other than ‘S’, ‘M’, or ‘L’

2.6.1.5 temperament

- Legal
  - Integer greater than 0 and less than 11
- Illegal
  - Not an integer
  - Integer less than 1
  - Integer more than 10

2.6.1.6 activity level

- Legal
  - Integer greater than 0 and less than 11
- Illegal
  - Not an integer
  - Integer less than 1
  - Integer more than 10

2.6.1.7 volume

- Legal
  - Integer greater than 0 and less than 11
- Illegal
  - Not an integer

- Integer less than 1
- Integer more than 10

2.6.1.8 notes

- Legal
  - Less than 513 characters
  - Alphanumeric characters
  - Symbols
- Illegal
  - More than 512 characters

2.6.2 Checklist

- ❏ Attempt to edit dog with new valid parameters
- ❏ Attempt to edit dog with new name including illegal characters
- ❏ Attempt to edit dog with new name longer than 150 characters
- ❏ Attempt to edit dog with new picture larger than 10MB
- ❏ Attempt to edit dog with new picture format other than PNG or JPEG
- ❏ Attempt to edit dog with new breed including illegal characters
- ❏ Attempt to edit dog with new breed longer than 150 characters
- ❏ Attempt to edit dog with new size not a single character
- ❏ Attempt to edit dog with new size using characters other than S, M, or L
- ❏ Attempt to edit dog with new temperament less than 1
- ❏ Attempt to edit dog with new temperament greater than 10
- ❏ Attempt to edit dog with new activity level less than 1
- ❏ Attempt to edit dog with new activity level greater than 10
- ❏ Attempt to edit dog with new volume less than 1
- ❏ Attempt to edit dog with new volume greater than 10
- ❏ Attempt to edit dog with new notes more than 512 characters

2.6.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.7 View user profile

Allows a user to view another user's profile (or their own profile)

2.7.1 Equivalence Classes

2.7.1.1 username

- Legal
  - Already in database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length

- Illegal
  - Not in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.7.2 Checklist

- ❏ Attempt to view user with valid username and in the database
- ❏ Attempt to view user with valid username and not in the database
- ❏ Attempt to view user with username including invalid characters
- ❏ Attempt to view user with username longer than 150 characters

2.7.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.8 View dog profile

Allows a user to view a dog's profile.

2.8.1 Equivalence Classes

2.8.1.1 dog name

- Legal
  - Already in database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Not in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.8.2 Checklist

- ❏ Attempt to view dog with valid name and in the database
- ❏ Attempt to view dog with valid name and not in the database
- ❏ Attempt to view dog with name including invalid characters
- ❏ Attempt to view dog with name longer than 150 characters

2.8.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.9 View park

Allows a user to view the page about a park.

2.9.1 Equivalence Classes

2.9.1.1 park name

- Legal
  - Already in database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Not in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.9.2 Checklist

- ❏ Attempt to view park with valid name and in the database
- ❏ Attempt to view park with valid name and not in the database
- ❏ Attempt to view park with name including invalid characters
- ❏ Attempt to view park with name longer than 150 characters

2.9.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.10 Review park

Allows a user to write a review for a park.

2.10.1 Equivalence Classes

2.10.1.1 park name

- Legal
  - Already in database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Not in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.10.1.1 park review

- Legal
  - Alphanumeric characters
  - Symbols
  - Less than 1025 characters in length
- Illegal
  - Longer than 1024 characters

2.10.1.1 star rating

- Legal
  - Integer greater than 0 and less than 6

- Illegal
  - Not an integer
  - Integer less than 1 or more than 5

2.10.2 Checklist

- ❏ Attempt to review park with valid name, review, and star rating
- ❏ Attempt to review park with name containing invalid characters
- ❏ Attempt to review park with name longer than 150 characters
- ❏ Attempt to review park with review longer than 1024 characters
- ❏ Attempt to review park with star rating not an integer
- ❏ Attempt to review park with star rating less than 1
- ❏ Attempt to review park with star rating more than 5

2.10.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

2.11 Schedule event

Allows a user to schedule a dog for time at a park.

2.11.1 Equivalence Classes

2.11.1.1 park name

- Legal
  - Already in database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Not in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.11.1.2 dog name

- Legal
  - Already in database
  - Alphanumeric characters and [_,@,+,., -] (commas and brackets not included)
  - Less than 151 characters in length
- Illegal
  - Not in database
  - Non-alphanumeric characters excluding [_,@,+,., -]
  - Longer than 150 characters

2.11.1.2 date

- Legal
  - mm/dd/yyyy

- Illegal
    - Any other input

2.11.1.2 time

- Legal
    - hh:mm (24-hour format)
- Illegal
    - Any other input

2.11.2 Checklist

- ❏ Attempt to schedule event with valid park name, dog name, date, and time
- ❏ Attempt to schedule event with park name containing invalid characters
- ❏ Attempt to schedule event with park name longer than 150 characters
- ❏ Attempt to schedule event with park name not in database
- ❏ Attempt to schedule event with dog name containing invalid characters
- ❏ Attempt to schedule event with dog name longer than 150 characters
- ❏ Attempt to schedule event with dog name not in database
- ❏ Attempt to schedule event with date in wrong format
- ❏ Attempt to schedule event with time in wrong format

2.11.3 Who will perform the tests

The automated tests will be created by Jacob Haldeman

3.0 Integration Testing

3.1 Purpose

Provides a systematic approach for assembling the software in an incremental fashion.

3.2 Integration Testing Checklist

- ❏ Attempt to create user profile
- ❏ Attempt to add dog to profile
- ❏ Attempt to edit user profile
- ❏ Attempt to edit dog profile
- ❏ Attempt to create park
- ❏ Attempt to schedule dog at park
- ❏ Attempt to create a review of the park

3.3 Who will perform the tests

The automated tests will be created by Ian Lundberg.

4.0 System Testing

4.1 Purpose

Provides an approach to determine whether the application satisfies all requirements.

4.2 System Testing Checklist

- ❏ Program starts
- ❏ Verify user login, for multi-user demo, users are logged in beforehand
- ❏ User can use all major functions
- ❏ User can go through a scenario
- ❏ Display results of scenario
- ❏ Store user edited info in database
- ❏ Verify admin login for a new or existing database
- ❏ Display all user information to admin in existing database, admin is capable of removing and adding users and performing basic maintenance
- ❏ Provide a learning experience for users to learn how to use the basics
- ❏ Show how to maintain the web app: find source code, make a small modification and recompile

4.3 Who will perform the tests

The automated tests will be created by Ian Lundberg.

5.0 Acceptance Testing

5.1 Purpose

Provides an approach to determine whether the application is ready for delivery.

5.2 Acceptance Testing Checklist

- ❏ Software configuration is complete, documents include all necessary criteria: requirements, design, test plan, test results, source code, user manual, maintenance procedure.
- ❏ Databases are fully configured and ready to be used
- ❏ Web app is capable of running on any modern browser without additional add-ons, extensions, plug-ins, etc.
- ❏ Server software will be able to be executed on a Windows PC without additional plug-ins. Software can be installed from a flash stick or website via download of an executable “install.exe”
- ❏ Documents are labelled with appropriate sections divided via labelled dividers
- ❏ Changes made after reviews are well indicated with yellow tabs, this document includes a revision history

- ❏ Cross references throughout the document. References are labelled with page numbers and sections
- ❏ Powerpoint presentation highlighting functionality of the web app
- ❏ All project members are present and contributions are clearly explained
- ❏ Program starts
- ❏ Verify user login, for multi-user demo, users are logged in beforehand
- ❏ User can use all major functions
- ❏ User can go through a scenario
- ❏ Display results of scenario
- ❏ Store user edited info in database
- ❏ Verify admin login for a new or existing database
- ❏ Display all user information to admin in existing database, admin is capable of removing and adding users and performing basic maintenance
- ❏ Provide a learning experience for users to learn how to use the basics
- ❏ Show how to maintain the web app: find source code, make a small modification and recompile

5.3 Who will perform the tests

The automated tests will be created by Ian Lundberg.

6.0 Testing Table

Test No.	Description	Input	Expected Output	Date Tested	Problem	Solution
Test 1 WhiteBox	Unit Test Login (PC)	username= ”user” Password= ”pass”	Navigate to user profile	12/3/2019	No Problems Found	No Solution Needed
Test 2 WhiteBox	Unit Test Login (BC)	username= ”user” password= ”notpass”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 3 WhiteBox	Unit Test Login (BC)	username/ password pair that is currently logged in	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

Test 4 WhiteBox	Unit Test Login (BC)	username= ”nulluser” password= ”pass”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 5 WhiteBox	Unit Test Login (BC)	username= ”user” password= ”nullpass”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 6 WhiteBox	Unit Test Login (SC)	username= ”user*” password= ”pass”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 7 WhiteBox	Unit Test Login (SC)	username= ”fZlu4cXuRldk774Sy621t0Zpc2101X0A04HGLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPYlM1vv1Kf49P97727hCeZo6UVxJX1p8UvJioKKM1uS3DWOA2YG14Jz12S35XdJruRNe56XW3L0pGO1CLp” password= ”pass”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 8 BlackBox	Unit Test Logout	username= ”user” (user logged in)	Navigate to login page	12/3/2019	No Problems Found	No Solution Needed
Test 9 BlackBox	Unit Test Logout	username= ”user” (user not	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		logged in)				
Test 10 BlackBox	Unit Test Logout	username= ”nulluser”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 11 BlackBox	Unit Test Logout	username= ”user*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 12 BlackBox	Unit Test Logout	username= ”fZlu4cXuRldk774Sy621t0Zpc2101X0A04HGLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPYlM1vv1Kf49P97727hCeZo6UVxJX1p8UvJioKKM1uS3DWOA2YG14Jz12S35XdJruRNe56XW3L0pGO1CLp”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 13 BlackBox	Unit Test Create Profile	username= ”user” password= ”pass”	Navigate to view user profile page, update database	12/3/2019	No Problems Found	No Solution Needed
Test 14 BlackBox	Unit Test Create Profile	username= ”existinguser” password= ”pass”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
~~Test 15 BlackBox~~	~~Unit Test Create Profile~~	~~Username too short~~	~~Error message displayed~~			
Test 16	Unit Test	username=	Error message	12/3/2019	No	No

BlackBox	Create Profile	”fZlu4cXuRldk774Sy621t0Zpc2101X0A04HGLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPYlM1vv1Kf49P97727hCeZo6UVxJX1p8UvJioKKM1uS3DWOA2YG14Jz12S35XdJruRNe56XW3L0pGO1CLp”	displayed		Problems Found	Solution Needed
Test 17 BlackBox	Unit Test Create Profile	username= ”user*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 18 BlackBox	Unit Test Create Profile	password= ”pass😀”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 19 BlackBox	Unit Test Edit Profile	username= ”user2”	Navigate to view profile, database updated	12/3/2019	No Problems Found	No Solution Needed
Test 20 BlackBox	Unit Test Edit Profile	username= ”existinguser”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 21 BlackBox	Unit Test Edit Profile	username= ”user*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 22 BlackBox	Unit Test Edit Profile	username= ”fZlu4cXuRldk774Sy621t0Zpc2101X0A04H	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		GLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPYlM1vv1Kf49P97727hCeZo6UVxJX1p8UvJioKKM1uS3DWOA2YG14Jz12S35XdJruRNe56XW3L0pGO1CLp”				
Test 23 BlackBox	Unit Test Edit Profile	password=”pass”	Navigate to view user profile, update database	12/3/2019	No Problems Found	No Solution Needed
Test 24 BlackBox	Unit Test Edit Profile	bio=”bio”	Navigate to view user profile, update database	12/3/2019	No Problems Found	No Solution Needed
Test 25 BlackBox	Unit Test Edit Profile	bio=”mggoqQU4ZCbFHWKEn5dQAaokeyf1e5pmxNnFEwnUtaEaftoiYezKqiwdU9FhPn703neW6vzVCceY1G1NxDxLeMV6QR51zvA8iiDOnHVjV16Y5ZacDGerth79Em2ULDce8UXlAVEh3D3pXjAZE4shXNqNXgrNpbuzvF	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		Tk6S9beA yyKhOyf4B nVHRrofc5 FI0SnDcvd 7yTlW689y cbi9Rh60Q P4dxczIQv qFWo6Jht mpMntO62 NZvTNvwY X2tmM0Eg YxxIQKEIU 9bnww6r1L n5rNKN5jV vzDoSLafP CNr46yZC zwB689uF cvXgi3php edv4656w4 XF7mJ0ge gQYQdooF UbFhdhc0 gU8hjjr3AC IIP27CFdJ h056SOqyt leaiG2nmG ZIm3gUpUj IISSX9sLA e1pegwgw w0GBaTX9 zBoMgu9i G6odaB3n QcogS7YE A67JsCRc nfDbFNMv 9OAQfhsm 2BI9SxaFc E01TNTm WuALo4Bq IUI9ObqrLi R09v84hK 9GPuD49z Z”				
Test 26 BlackBox	Unit Test Edit Profile	parkname= ”park”	Navigate to view user profile, update	12/3/2019	No Problems Found	No Solution Needed

			database			
Test 27 BlackBox	Unit Test Edit Profile	parkname= ”park*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 28 BlackBox	Unit Test Edit Profile	parkname= ”nullpark”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 29 BlackBox	Unit Test Create Dog	name=”dog ” breed=”bre ed” size=”s” temperame nt=”1 activity level=”1” volume=”1” notes=”not es”	Navigate to view user profile, update database	12/3/2019	No Problems Found	No Solution Needed
Test 30 BlackBox	Unit Test Create Dog	name=”dog *”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 31 BlackBox	Unit Test Create Dog	name=”fZlu 4cXuRldk7 74Sy621t0 Zpc2101X0 A04HGLu6 4iHVSb2bu nV4Qf1HX xxdm3nzfH 0UAZvPZg CDa5XPYl M1vv1Kf49 P97727hC eZo6UVxJ X1p8UvJio KKM1uS3 DWOA2YG 14Jz12S35 XdJruRNe 56XW3L0p GO1CLp”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

Test 32 BlackBox	Unit Test Create Dog	Photo with size > 10MB	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 33 BlackBox	Unit Test Create Dog	Photo that is not .PNG or .JPEG	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 34 BlackBox	Unit Test Create Dog	breed=”bree d*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 35 BlackBox	Unit Test Create Dog	breed=”fZl u4cXuRldk 774Sy621t 0Zpc2101X 0A04HGLu 64iHVSb2b unV4Qf1H Xxxdm3nzf H0UAZvPZ gCDa5XPY lM1vv1Kf4 9P97727h CeZo6UVx JX1p8UvJi oKKM1uS3 DWOA2YG 14Jz12S35 XdJruRNe 56XW3L0p GO1CLp”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 36 BlackBox	Unit Test Create Dog	size=”sml””	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 37 BlackBox	Unit Test Create Dog	size=”a”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 38 BlackBox	Unit Test Create Dog	temperame nt=”0”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 39 BlackBox	Unit Test Create Dog	temperame nt=”11”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

Test 40 BlackBox	Unit Test Create Dog	Activity level=”0”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 41 BlackBox	Unit Test Create Dog	Activity level=”11”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 42 BlackBox	Unit Test Create Dog	volume=”0”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 43 BlackBox	Unit Test Create Dog	volume=”11”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 44 BlackBox	Unit Test Create Dog	notes=”mggoqQU4ZCbFHWKEn5dQAaokeyf1e5pmxNnFEwnUtaEaftoiYezKqiwdU9FhPn703neW6vzVCceY1G1NxDxLeMV6QR51zvA8iiDOnHVjV16Y5ZacDGerth79Em2ULDce8UXIAVEh3D3pXjAZE4shXNqNXgrNpbuzvFTk6S9beAyyKhOyf4BnVHRrofc5FI0SnDcvd7yTlW689ycbi9Rh60QP4dxczIQvqFWo6JhtmpMntO62NZvTNvwYX2	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		tmM0EgYxxIQKEIU9bnww6r1Ln5rNKN5jVvzDoSLafPCNr46yZCzwB689uFcvXgi3phpedv4656w4XF7mJ0gegQYQdooFUbFhdhc0gU8hjjr3ACIIP27CFdJh056SOqytleaiG2nmGZIm3gUpUjIISSX9sLAe1pegwgww0GBaTX9zBoMgu9iG6odaB3nQcogS7YEA67JsCRcnfDbFNMv9OAQfhsm2BI9SxaFcE01TNTmWuALo4BqIUI9ObqrLiR09v84hK9GPuD49zZ”				
Test 45 BlackBox	Unit Test Edit Dog	name=”dog” breed=”breed” size=”s” temperament=”1 activity level=”1” volume=”1” notes=”notes”	Navigate to view user profile, update database	12/3/2019	No Problems Found	No Solution Needed

Test 46 BlackBox	Unit Test Edit Dog	name=”dog*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 47 BlackBox	Unit Test Edit Dog	name=”fZlu4cXuRldk774Sy621t0Zpc2101X0A04HGLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPYlM1vv1Kf49P97727hCeZo6UVxJX1p8UvJioKKM1uS3DWOA2YG14Jz12S35XdJruRNe56XW3L0pGO1CLp”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 48 BlackBox	Unit Test Edit Dog	Photo with size > 10MB	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 49 BlackBox	Unit Test Edit Dog	Photo that is not .PNG or .JPEG	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 50 BlackBox	Unit Test Edit Dog	breed=”breed*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 51 BlackBox	Unit Test Edit Dog	breed=”fZlu4cXuRldk774Sy621t0Zpc2101X0A04HGLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPY	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		lM1vv1Kf4 9P97727h CeZo6UVx JX1p8UvJi oKKM1uS3 DWOA2YG 14Jz12S35 XdJruRNe 56XW3L0p GO1CLp”				
Test 52 BlackBox	Unit Test Edit Dog	size=”sml”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 53 BlackBox	Unit Test Edit Dog	size=”a”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 54 BlackBox	Unit Test Edit Dog	temperame nt=”0”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 55 BlackBox	Unit Test Edit Dog	temperame nt=”11”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 56 BlackBox	Unit Test Edit Dog	Activity level=”0”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 57 BlackBox	Unit Test Edit Dog	Activity level=”11”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 58 BlackBox	Unit Test Edit Dog	volume=”0”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 59 BlackBox	Unit Test Edit Dog	volume=”1 1”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 60 BlackBox	Unit Test Edit Dog	notes=”mg goqQU4ZC bFHWKEn 5dQAaoke yf1e5pmxN	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		nFEwnUta EaftoiYezK qiwdU9Fh Pn703neW 6vzVCceY 1G1NxDxL eMV6QR5 1zvA8iiDO nHVjV16Y 5ZacDGert h79Em2UL Dce8UXIA VEh3D3pX jAZE4shX NqNXgrNp buzvFTk6S 9beAyyKh Oyf4BnVH Rrofc5FI0S nDcvd7yTl W689ycbi9 Rh60QP4d xczIQvqF Wo6Jhtmp MntO62NZ vTNvwYX2 tmM0EgYx xIQKEIU9b nww6r1Ln5 rNKN5jVvz DoSLafPC Nr46yZCz wB689uFc vXgi3phpe dv4656w4 XF7mJ0ge gQYQdooF UbFhdhc0 gU8hjjr3AC lIP27CFdJ h056SOqyt leaiG2nmG ZIm3gUpUj lISSX9sLA e1pegwgw w0GBaTX9				

		zBoMgu9iG6odaB3nQcogS7YEA67JsCRcnfDbFNMv9OAQfhsm2BI9SxaFcE01TNTmWuALo4BqIUI9ObqrLiR09v84hK9GPuD49zZ”				
Test 61 BlackBox	Unit Test View User Profile	username=”Ian”	Navigate to view user profile	12/3/2019	No Problems Found	No Solution Needed
Test 62 BlackBox	Unit Test View User Profile	username=”nulluser”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 63 BlackBox	Unit Test View User Profile	username=”user*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 64 BlackBox	Unit Test View User Profile	username=”fZlu4cXuRldk774Sy621t0Zpc2101X0A04HGLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPYlM1vv1Kf49P97727hCeZo6UVxJX1p8UvJioKKM1uS3DWOA2YG14Jz12S35XdJruRNe56X	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		W3L0pGO1CLp”				
Test 65 BlackBox	Unit Test View Dog Profile	name=”dog”	Navigate to view dog profile	12/3/2019	No Problems Found	No Solution Needed
Test 66 BlackBox	Unit Test View Dog Profile	name=”null dog”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 67 BlackBox	Unit Test View Dog Profile	name=”dog*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 68 BlackBox	Unit Test View Dog Profile	name=”fZlu4cXuRldk774Sy621t0Zpc2101X0A04HGLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPYlM1vv1Kf49P97727hCeZo6UVxJX1p8UvJioKKM1uS3DWOA2YG14Jz12S35XdJruRNe56XW3L0pGO1CLp”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 69 BlackBox	Unit Test Review Park	park_name=”Schenley”	Navigate to view park, update database	12/3/2019	No Problems Found	No Solution Needed
Test 70 BlackBox	Unit Test Review Park	park_name=”Schenley*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 71 BlackBox	Unit Test Review	park_name=”7CZtN5R	Error message displayed	12/3/2019	No Problems	No Solution

	Park	ixBsnu0mmup7ztWVt5sqJov9z75xB0lwux7VjiSigfW1pvAQHvqrtAMGWkbdVMRAl1BIqQ1NdZy6EoHiYoMISOK5LvFbIxMrdUUHwA358ncd3FPoIKO8lZkZYM2lkudgR69wpyudhghnCKIj840xRe3go”			Found	Needed
Test 72 BlackBox	Unit Test Review Park	park_review =”KskBpbuNorjCnzrEFwLcLyowFRHm5dNRya8Tf1EzM4ERriYYlYGuyESvDbPnCvLoqBSaV4gnxDptwUcLcz9h17RRnwGycWz2MBHhjEOUPlosw1wMhhjXAAJXKI4XArrvXIKvMFu1DVMmTNk8k1SHzdhuhsK1ZTKaYPdMGm9pIfKHX5jsIkxinLCNhhqueJxhZK9gFDh34bMBtW5h2twIsC	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		uO8hWRji3 vMskaEZm TwlvRPs98 6FHCx03M sbH2OSM 50niDCQdI 9PFPyyqrz hIOlJhFMd 1Pf7DAwj9 mG4oJKlX FFWrjytzac t0qEarGigT lUrSsPXfL 0Z0BascN TOAnJi1us UR1O4E0 DXPAMYry AOwTp0Al PEBJlA8iH l2zLVD6yQ PGfhcXpA T98Bqwggi ATPwp2d9 4BBLQEly FPedfXJjT nTSCiYCb 4LKnxNxfY ckbZ76nB Xi4OT1Yq nLOuqXlE HxZFWl0qj oLq9gwntb uhWOhVT 3EtAOUEU gZGpNvqU 1MDXV7p0 OO2LxkLg RnEN77oO 4tJNOh5sK EsFUbCKb iXqbiaPdR QDPeV3B dlyX4JYzf CZ3cQ28g CY0zsK17 oR71aT51 K4PeYwe0				

		HagYOxsMrfyf5exjeLMs7vMag6P8kkqIUC1ky3hankpFkpMVQwsylrdZEpp6lO0Masg3yS1ChDwQxt3ihKAQGmeWVGLFniWViPWwZmIOyF5InJpQRf9dvZv4acyknEMzWrMfptZFxnQZpd6sEYXjayn9XHRDWh1YxHAaFL9of82uruDBecUeVPbdspeguwZXB6EjeVQB9vhrdvtUjD36lxkUHkvbK0M6jzmyzv6yRZxVUA80p4yQ5wGqIgoZtkCOTDg2WPnFTZIdMj6CSP4sM2zKrVqnJphkV7u4qDLTB1WCt48yOUkhrsC4rtTZzTnrDlDtl76Xq37A4aoHj2dLLlOltRhRfOgLg7vyBxCB9eB6ABnvtZE0PwdLsilHVX6zrLYJtAFbPoJTsq3wrntn				

		kop7K7HUOPBcV”				
Test 73 BlackBox	Unit Test Review Park	star_rating =”five”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 74 BlackBox	Unit Test Review Park	star_rating =0	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 75 BlackBox	Unit Test Review Park	star_rating =6	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 76 BlackBox	Unit Test View park	park_name =”Schenley”	Navigate to park page	12/3/2019	No Problems Found	No Solution Needed
Test 77 BlackBox	Unit Test View park	park_name =”Central Park”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 78 BlackBox	Unit Test View park	park_name =”Schenley*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 79 BlackBox	Unit Test View park	park_name =”7CZtN5RixBsnu0mmup7ztWVt5sqJov9z75xB0lwux7VjiSigfW1pvAQHvqrtAMGWkbdVMRAl1BIqQ1NdZy6EoHiYoMISOK5LvFbIxMrdUUHwA358ncd3FPoIKO8lZkZYM2lkudgR69wpyudhghnCKIj840xRe3go”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

Test 80 BlackBox	Unit Test Schedule event	Park_name =”Schenley”, dog_name =”dog”, data=”12/1/2019”, time=”18:00”	New event scheduled	12/3/2019	No Problems Found	No Solution Needed
Test 81 BlackBox	Unit Test Schedule event	park_name =”Schenley*”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 82 BlackBox	Unit Test Schedule event	park_name =”7CZtN5RixBsnu0mmup7ztWVt5sqJov9z75xB0lwux7VjiSigfW1pvAQHvqrtAMGWkbdVMRAl1BIqQ1NdZy6EoHiYoMISOK5LvFbIxMrdUUHwA358ncd3FPoIKO8lZkZYM2lkudgR69wpyudhghnCKIj840xRe3go”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 83 BlackBox	Unit Test Schedule event	park_name =”Central Park”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 84 BlackBox	Unit Test Schedule event	dog_name =”nulldog”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 85 BlackBox	Unit Test Schedule event	name=”fZlu4cXuRldk774Sy621t0Zpc2101X0	Error message displayed	12/3/2019	No Problems Found	No Solution Needed

		A04HGLu64iHVSb2bunV4Qf1HXxxdm3nzfH0UAZvPZgCDa5XPYlM1vv1Kf49P97727hCeZo6UVxJX1p8UvJioKKM1uS3DWOA2YG14Jz12S35XdJruRNe56XW3L0pGO1CLp”				
Test 86 BlackBox	Unit Test Schedule event	dog_name =”nulldog”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 87 BlackBox	Unit Test Schedule event	data=”12012019	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 88 BlackBox	Unit Test Schedule event	time=”1pm”	Error message displayed	12/3/2019	No Problems Found	No Solution Needed
Test 89 BlackBox	Integration Test Create profile	Attempt to create user profile	User profile created	12/3/2019	No Problems Found	No Solution Needed
Test 90 BlackBox	Integration Test Add dog to profile	Attempt to create new dog on user profile	Dog profile created and linked to user profile	12/3/2019	No Problems Found	No Solution Needed
Test 91 BlackBox	Integration Test Edit user profile	Attempt to edit user profile	User profile changed	12/3/2019	No Problems Found	No Solution Needed
Test 92 BlackBox	Integration test Create park	Attempt to create new park	New park created	12/3/2019	No Problems Found	No Solution Needed

Test 93 BlackBox	Integration Test Schedule event	Attempt to schedule a dog at a park	Dog scheduled at park	12/3/2019	No Problems Found	No Solution Needed
Test 94 BlackBox	Integration Test Review park	Attempt to create new park review	Park review created	12/3/2019	No Problems Found	No Solution Needed
Test 95 BlackBox	Integration Test Edit dog profile	Attempt to edit dog profile	Dog profile changed	12/3/2019	No Problems Found	No Solution Needed
Test 96 BlackBox	System Test	Start program	Navigates to login page	12/3/2019	No Problems Found	No Solution Needed
Test 97 BlackBox	System Test	User logs in	Navigates to main page	12/3/2019	No Problems Found	No Solution Needed
Test 98 BlackBox	System Test	User clicks view profile	Navigates to user's profile page	12/3/2019	No Problems Found	No Solution Needed
Test 99 BlackBox	System Test	User clicks add dog	Navigates to create dog page	12/3/2019	No Problems Found	No Solution Needed
Test 100 BlackBox	System Test	User enters dog information and clicks submit	Navigates to view user profile, update database	12/3/2019	No Problems Found	No Solution Needed
Test 101 BlackBox	System Test	User clicks edit profile	Navigates to edit user profile	12/3/2019	No Problems Found	No Solution Needed
Test 102 BlackBox	System Test	User enters new profile information , click submit	Navigates to view user profile, updates database	12/3/2019	No Problems Found	No Solution Needed

Test 103 BlackBox	System Test	User clicks on a dog	Navigates to view dog profile	12/3/2019	No Problems Found	No Solution Needed
Test 104 BlackBox	System Test	User clicks edit dog profile	Navigates to edit dog profile	12/3/2019	No Problems Found	No Solution Needed
Test 105 BlackBox	System Test	User enters new dog information and clicks submit	Navigates to view dog profile, updates database	12/3/2019	No Problems Found	No Solution Needed
Test 106 BlackBox	System Test	User clicks home button	Navigates to home page with map and parks listed	12/3/2019	No Problems Found	No Solution Needed
Test 107 BlackBox	System Test	User clicks on a park	Navigate to view park	12/3/2019	No Problems Found	No Solution Needed
Test 108 BlackBox	System Test	User clicks schedule visit	Open schedule visit window	12/3/2012/3 /201919	No Problems Found	No Solution Needed
Test 109 BlackBox	System Test	User inputs a date and time and clicks schedule	Navigate to view park, update database with reservation	12/3/2019	No Problems Found	No Solution Needed
Test 110 BlackBox	System Test	User clicks leave review	Open review window	12/3/2019	No Problems Found	No Solution Needed
Test 111 BlackBox	System Test	User enters review information and clicks submit	Navigates to view park, update database with review	12/3/2019	No Problems Found	No Solution Needed

Test 112 BlackBox	System Test	User clicks logout button	Navigates to login page	12/3/2019	No Problems Found	No Solution Needed
Test 113 BlackBox	Acceptance Test Databases Configured	Initial database setup	Working databases	12/3/2019	No Problems Found	No Solution Needed
Test 114 BlackBox	Acceptance Test Browser compatibility	Run on modern browser	Website displays and works correctly	12/3/2019	No Problems Found	No Solution Needed
Test 115 BlackBox	Acceptance Test Server Install	Install on Window PC	Server software installs and runs correctly	12/3/2019	No Problems Found	No Solution Needed
Test 116 BlackBox	Acceptance Test Program starts	Run program	Program starts and runs correctly	12/3/2019	No Problems Found	No Solution Needed
Test 117 BlackBox	Acceptance Test User login	Log in user	User is logged in	12/3/2019	No Problems Found	No Solution Needed
Test 118 BlackBox	Acceptance Test Major Functions	User attempts major functions	Functions work correctly	12/3/2019	No Problems Found	No Solution Needed
Test 119 BlackBox	Acceptance Test Scenario	User goes through scenario	Scenario completed correctly	12/3/2019	No Problems Found	No Solution Needed
Test 120 BlackBox	Acceptance Test Scenario Results	Attempt to view results of scenario	Scenario results displayed	12/3/2019	No Problems Found	No Solution Needed
Test 121 BlackBox	Acceptance Test Database	Attempt to store new user data	Data stored correctly	12/3/2019	No Problems Found	No Solution Needed

	Storage	in database				
Test 122 BlackBox	Acceptance Test Admin login	Attempt to log in as admin	Admin logged in	12/3/2019	No Problems Found	No Solution Needed
Test 123 BlackBox	Acceptance Test Admin maintenance	Attempt to perform maintenance functions as admin	Maintenance functions performed correctly	12/3/2019	No Problems Found	No Solution Needed
Test 124 BlackBox	Acceptance Test Help info	User requests help information on website	Help information displayed to user	12/3/2019	No Problems Found	No Solution Needed