Team: Jorge Benavides, Jianyi Chen, Jwalin Maniar, Ryan Kelley.

Title: Rover&Friends

Summary: Rover&Friends is a website to help manage a user's pet. The user can create a page for themselves and their pet. Once the user has created their page, they'll be able to search for pet sitters or people with similar pets to become friends. The user will be able to add walk appointments or pet sitters to a calendar. The user will also be able to view a map of local pet resources within a mile range of his location. The user will also be able to buy company products from the site by adding things to their shopping cart.

Requirement Tables:

There is no Business requirement.

User requirements:

ID	Requirement	Topic area	Actor	Priority
US-01	As a user, I want to be able to sign up.	Authentication	User	High
US-02	As a user, I want to be able to login to my account.	Authentication	User	High
US-03	As a user, I want to be able to log out.	Authentication	User	High
US-04	As a user, I want to be able to add photos of me and my pet.	Authentication	User	High
US-05	As a user, I want to be able to search for my friends.	Search	User	Medium
US-06	As a user, I want to be able to view other people's' profile and add them to my friend list.	Friends List	User	Medium
US-07	As a user, I want to be able to delete my friend from friend list.	Friends List	User	Low
US-08	As a user, I want to be able to search for pet sitters around me.	Search	User	Critical
US-09	As a user, I want to be able to buy pet tags.	Purchase	User	Medium

US-10	As a user, I want to be able to add pet tags to my shopping cart.	Purchase	User	Medium
US-11	As a user, I want to be able to checkout.	Purchase	User	Medium
US-12	As a user, I want to be able to add/ remove events from calendar.	Event	User	Low
US-13	As a admin, I want have privilege to delete user's account.	Authentication	Admin	High
US-14	As a admin, I want to be able to add or remove products.	Management	Admin	Medium
US-15	As a sitter, I want to be able to set up my online status.	Profile	Sitter	High
US-16	As a sitter, I want to be able to add contact information to profile.	Profile	Sitter	High
US-17	As a user, I want to see pet store around me on a map.	Map	User	Medium
US-18	As a user, I want to be able to edit my profile (user's photo and descriptions).	Profile	User	Medium
US-19	As a user, I want to be able to search for people to become friend.	Search	User	Medium
US-20	As a user, I want to be able to set up my dog sitting status.	Profile	User	High
US-21	As a user, I want to be able to email help desk.	Help Desk	User	Low

Functional Requirements:

ID	Requirement	Topic Area	Priority
FR-01	All users' (including sitters) profiles should be stored in the database.	Profile	Critical
FR-02	All products inventory should stored in the database.	Sale	Medium
FR-03	Sent receipt through email after purchase.	Notification	Medium
FR-04	A maximum of 1000 characters for users' profile descriptions.	Implementation	Medium
FR-05	Our system will use Hibernate to persist user profile information into MySQL database.	Hibernate	High

FR-06	A maximum of one picture for user and their pet.	Implementation	Medium
FR-07	A minimum 6 characters for password.	Implementation	Medium
FR-08	The homepage will have a navbar to all other pages on the website	implementation	Medium

Non-Functional Requirements:

ID	Description	Priority
NFR-01	Performance: During any search, system should provide an answer within 10 seconds.	Medium
NFR-02	Interface: The system should make request to Google Maps API to show the location of user who need a pet sitter.	Medium
NFR-03	Platform Constraints: Website should function identical on different platforms.	Medium
NFR-04	Website functionality will behave the same on different browsers	Medium
NFR-05	System will use Spring Security to keep information safe.	High
NFR-06	Any picture that a user will upload as their profile picture must have a JPEG extension	Medium

Data Storage:

For Data Storage, we are going to use mysql database through Hibernate.

Sign In

Email

Password

Sign In

Create Account

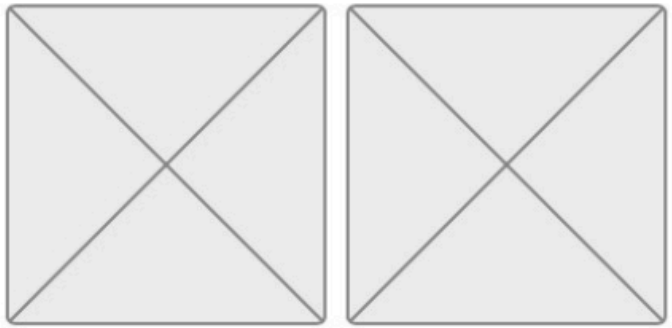
First name

Last name

Email address

New password

Username



Add user photo

Add pet photo

Edit profile

View your friends

Search for friends

Search for dog sitter

Become available for dog sitting by changing status

Not Available for dog sitting

Avaliable for dog sitting

Not Available for dog sitting

By setting status as Available for dog sitting, You will show up in the search for dog sitters queue and you will gain access to edit your dog sitting information when editting your profile

Buy pet tag

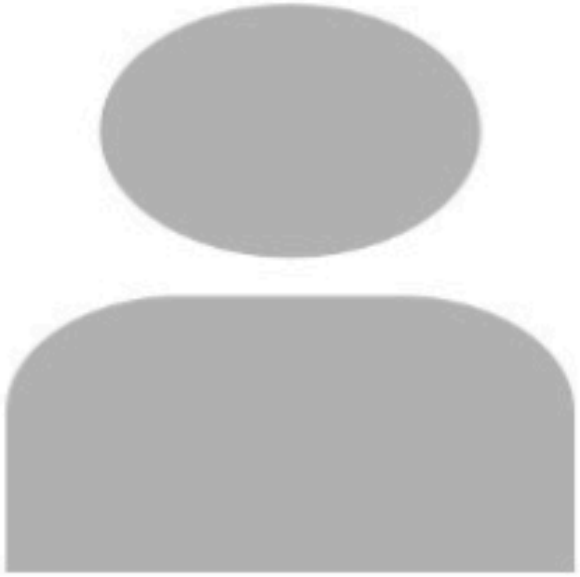
Find pet stores

Calendar

Sign out

Delete account

Username



About me and my pet....

Return to homepage

Save

Username

About me and my pet....

Information on your dog sitting services i.e how to contact you.

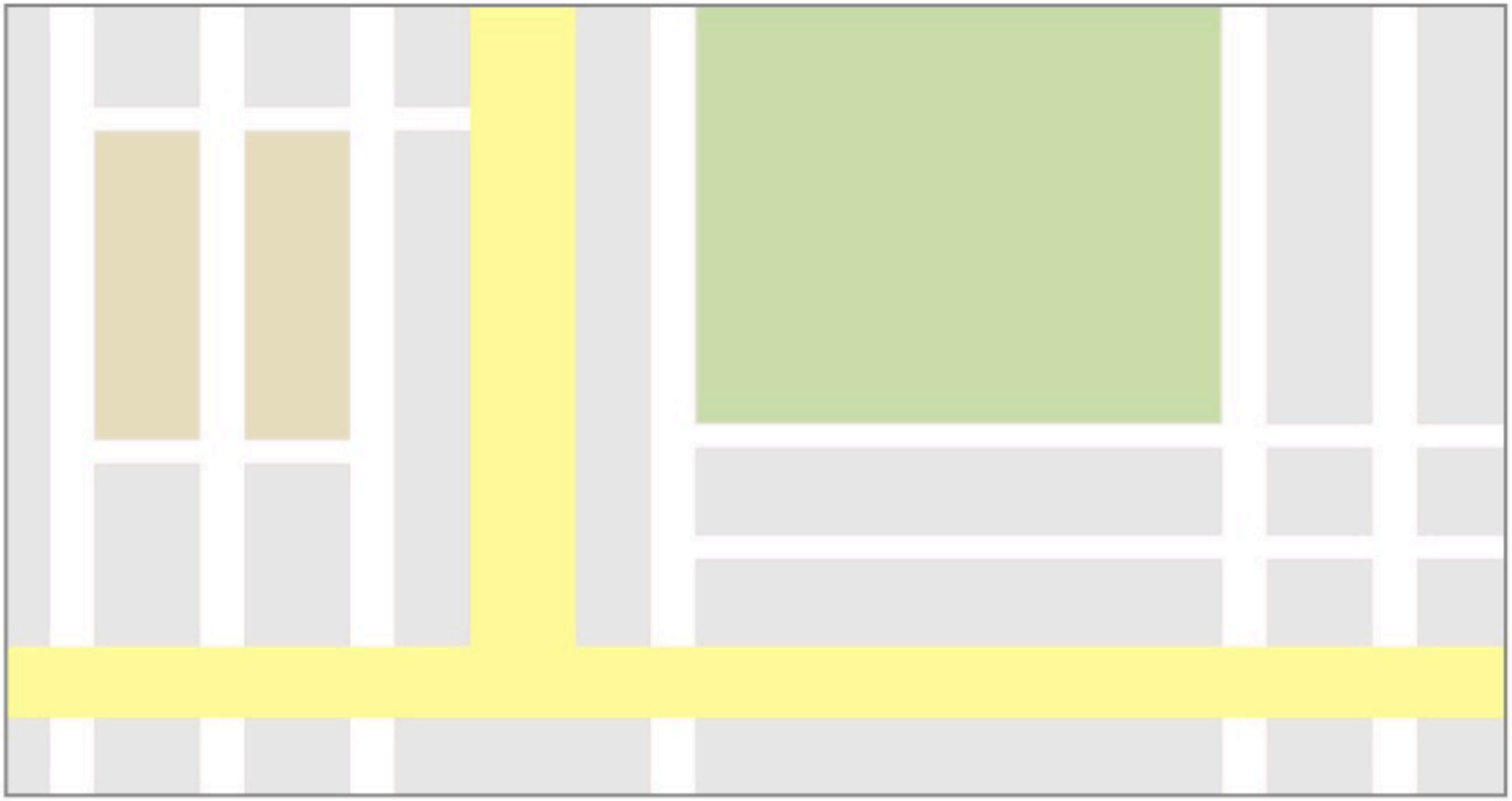
Return to homepage

Save

Pet stores with 30 miles of

zip code

submit



Return to homepage

Click on date to add, delete, or edit event

2017

January

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

February

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

March

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

April

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

May

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

June

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

July

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

August

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

September

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

October

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

November

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

December

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Year

Return to homepage

Event title

add event title

Event description

add event details here

Return to homepage | Save | Delete

Search for new friends within 5 miles or by username Search by username or name

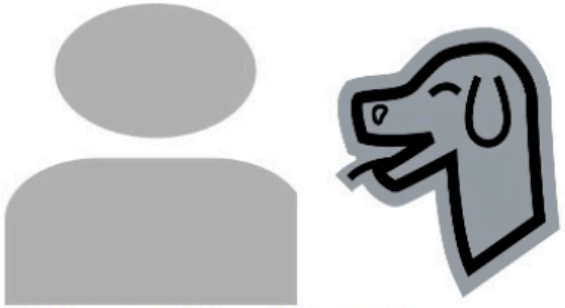
Name + Username

Name + Username

Name + Username

Name + Username

Name + Username



Name + Username

Name + Username

Name + Username

Name + Username

Return to homepage

Previous 1 2 3 Next

Search for dog sitters within 5 miles or by username Search by username or name

Name + Username

Name + Username

Name + Username

Name + Username

Name + Username

Name + Username

Name + Username

Name + Username

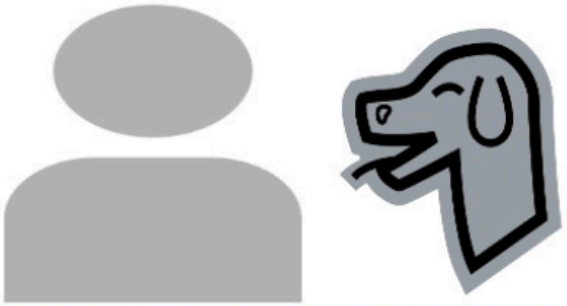
Name + Username

Return to homepage

Previous 1 2 3 Next

Search your friends

Search by name or username



Friend's name + username

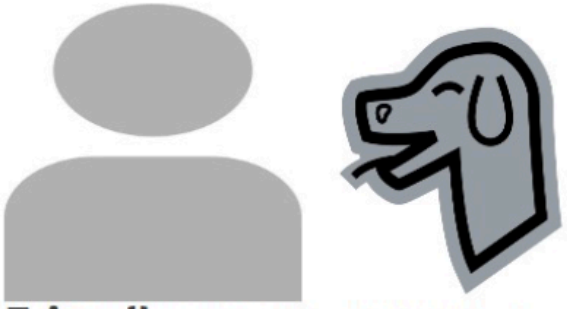
Friend's name + username

Friend's name + username

Friend's name + username

Friend's name + username

Friend's name + username



Friend's name + username

Friend's name + username

Friend's name + username

Return to homepage

Previous 1 2 3 Next

Name + Username

About me and my pet....

Return to homepage

Add friend

Friend's name + username

About me and my pet....

Return to homepage

Delete friend

Shipping Address

Full Name

Address line 1:

Address line 2:

City

State

Zip

Country

Phone number

Billing Address

Full Name

Address line 1:

Address line 2:

City

State

Zip

Country

Phone number

Payment method

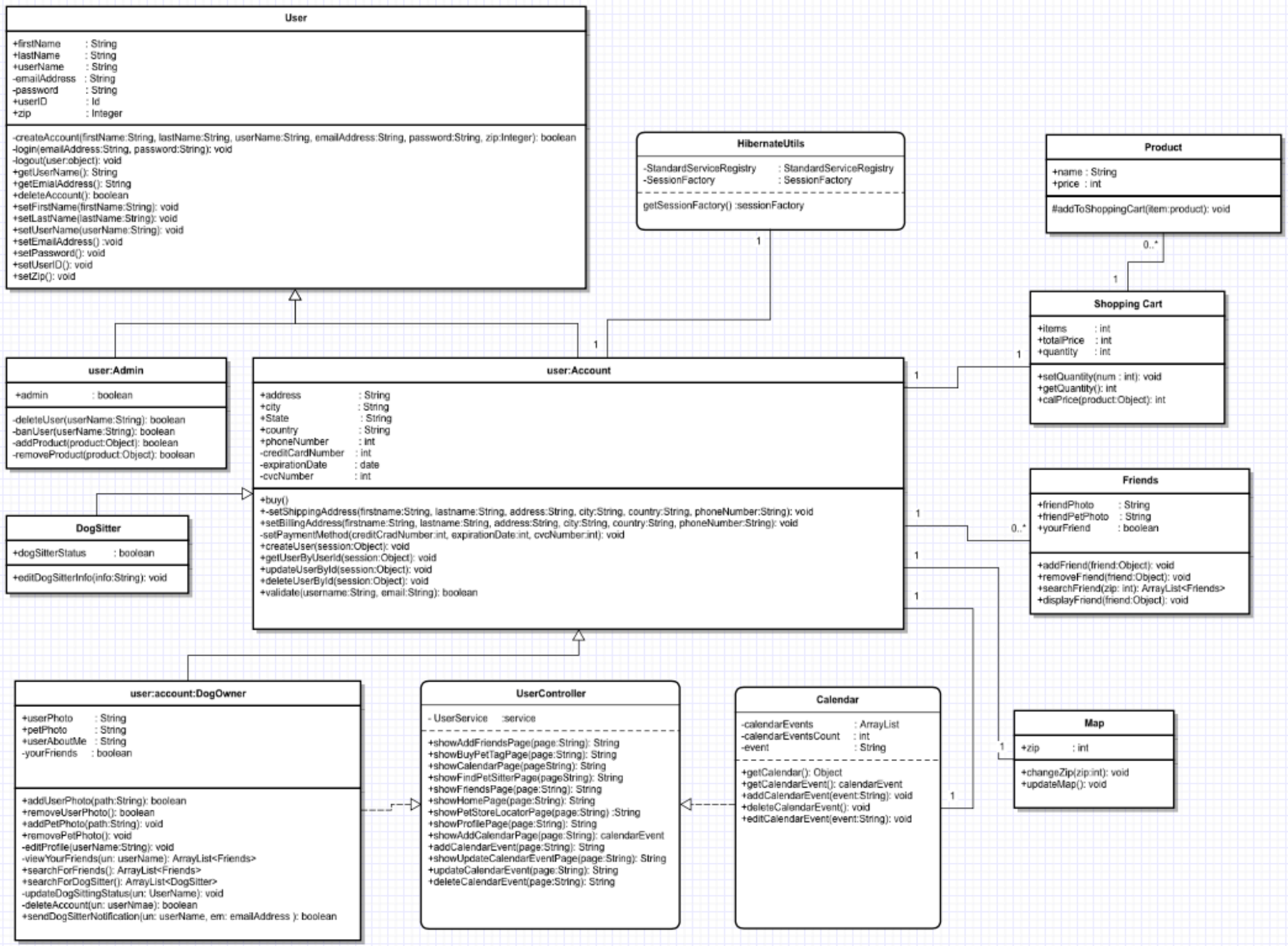
Credit card number

Expiration date 3 2019

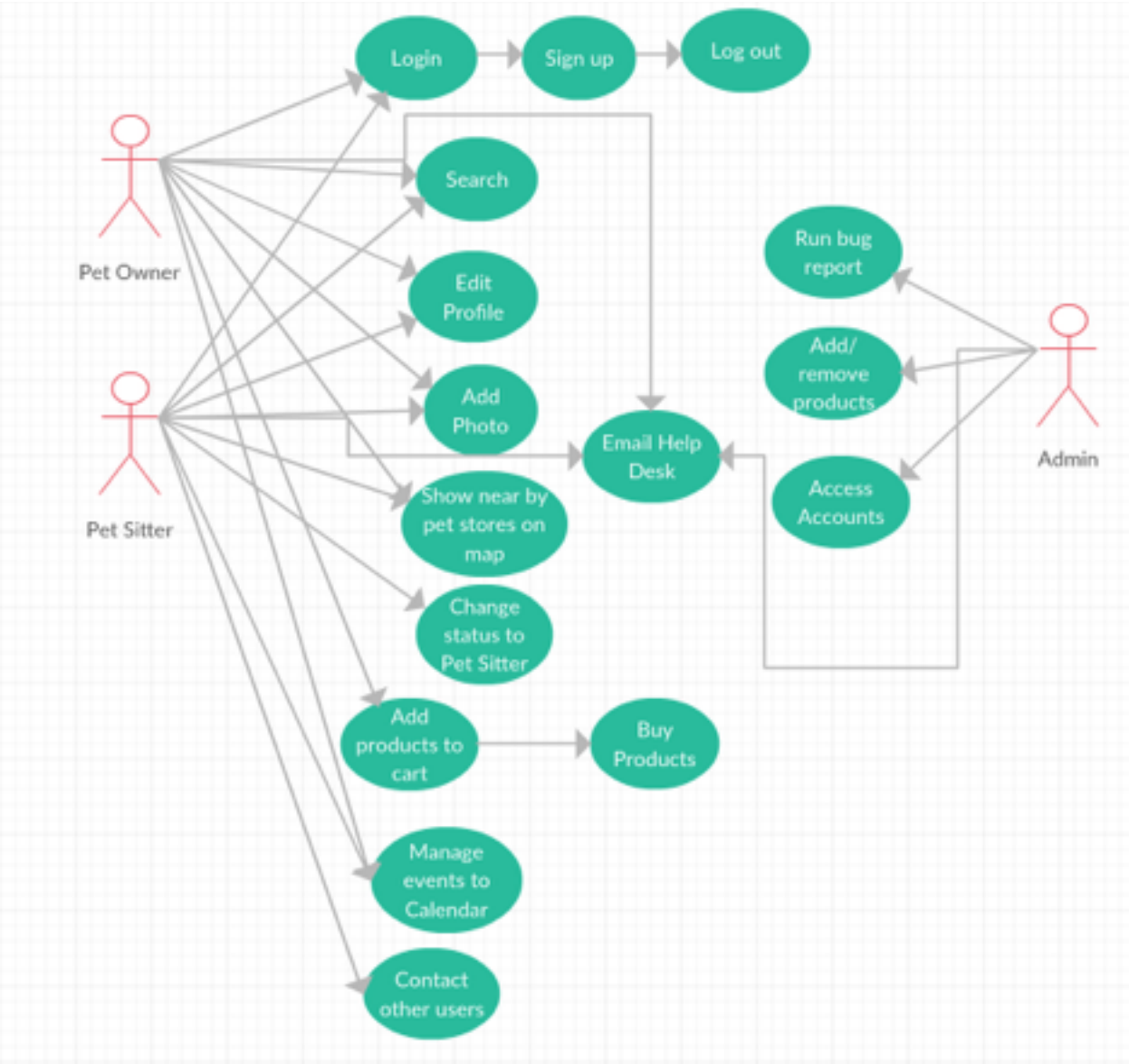
CVC number

Buy

Return to homepage


User
+firstName : String
+lastName : String
+userName : String
-emailAddress : String
-password : String
+userID : Id
+zip : Integer
-createAccount(firstName:String, lastName:String, userName:String, emailAddress:String, password:String, zip:Integer): boolean
-login(emailAddress:String, password:String): void
-logout(user:object): void
+getUserName(): String
+getEmialAddress(): String
+deleteAccount(): boolean
+setFirstName(firstName:String): void
+setLastName(lastName:String): void
+setUserName(userName:String): void
+setEmailAddress() :void
+setPassword(): void
+setUserID(): void
+setZip(): void
HibernateUtils
-StandardServiceRegistry : StandardServiceRegistry
-SessionFactory : SessionFactory
getSessionFactory() :sessionFactory
Product
+name : String
+price : int
#addToShoppingCart(item:product): void
0..*
1
Shopping Cart
+items : int
+totalPrice : int
+quantity : int
+setQuantity(num : int): void
+getQuantity(): int
+calPrice(product:Object): int
1
1
1
user:Admin
+admin : boolean
-deleteUser(userName:String): boolean
-banUser(userName:String): boolean
-addProduct(product:Object): boolean
-removeProduct(product:Object): boolean
user:Account
+address : String
+city : String
+State : String
+country : String
+phoneNumber : int
-creditCardNumber : int
-expirationDate : date
-cvcNumber : int
+buy()
+-setShippingAddress(firstname:String, lastname:String, address:String, city:String, country:String, phoneNumber:String): void
+setBillingAddress(firstname:String, lastname:String, address:String, city:String, country:String, phoneNumber:String): void
-setPaymentMethod(creditCradNumber:int, expirationDate:int, cvcNumber:int): void
+createUser(session:Object): void
+getUserByUserId(session:Object): void
+updateUserById(session:Object): void
+deleteUserById(session:Object): void
+validate(username:String, email:String): boolean
DogSitter
+dogSitterStatus : boolean
+editDogSitterInfo(info:String): void
Friends
+friendPhoto : String
+friendPetPhoto : String
+yourFriend : boolean
+addFriend(friend:Object): void
+removeFriend(friend:Object): void
+searchFriend(zip: int): ArrayList<Friends>
+displayFriend(friend:Object): void
1
0..*
1
1
1
1
user:account:DogOwner
+userPhoto : String
+petPhoto : String
+userAboutMe : String
-yourFriends : boolean
+addUserPhoto(path:String): boolean
+removeUserPhoto(): boolean
+addPetPhoto(path:String): void
+removePetPhoto(): void
-editProfile(userName:String): void
-viewYourFriends(un: userName): ArrayList<Friends>
+searchForFriends(): ArrayList<Friends>
+searchForDogSitter(): ArrayList<DogSitter>
-updateDogSittingStatus(un: UserName): void
-deleteAccount(un: userNmae): boolean
+sendDogSitterNotification(un: userName, em: emailAddress): boolean
UserController
- UserService :service
+showAddFriendsPage(page:String): String
+showBuyPetTagPage(page:String): String
+showCalendarPage(pageString): String
+showFindPetSitterPage(pageString): String
+showFriendsPage(page:String): String
+showHomePage(page:String): String
+showPetStoreLocatorPage(page:String) :String
+showProfilePage(page:String): String
+showAddCalendarPage(page:String): calendarEvent
+addCalendarEvent(page:String): String
+showUpdateCalendarEventPage(page:String): String
+updateCalendarEvent(page:String): String
+deleteCalendarEvent(page:String): String
Calendar
-calendarEvents : ArrayList
-calendarEventsCount : int
-event : String
+getCalendar(): Object
+getCalendarEvent(): calendarEvent
+addCalendarEvent(event:String): void
+deleteCalendarEvent(): void
+editCalendarEvent(event:String): void
1
Map
+zip : int
+changeZip(zip:int): void
+updateMap(): void

Use Cases:


Login
Sign up
Log out
Pet Owner
Search
Edit Profile
Add Photo
Pet Sitter
Email Help Desk
Show near by pet stores on map
Change status to Pet Sitter
Add products to cart
Buy Products
Manage events to Calendar
Contact other users
Run bug report
Add/ remove products
Access Accounts
Admin

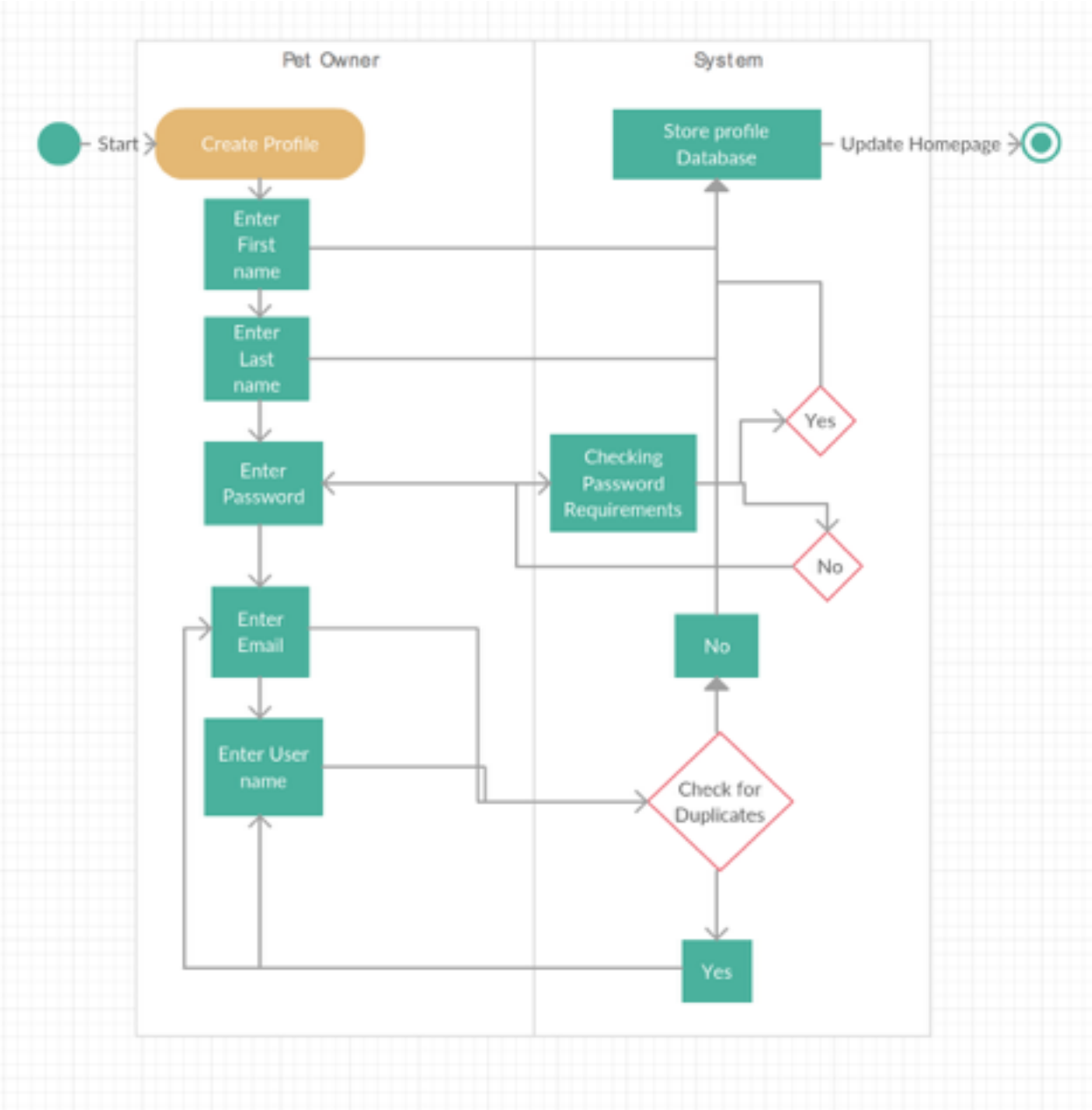
Use Case documentations:

Use Case ID:	US-01
Use Case Name:	User sign up.
Descriptions:	User will create account by entering their first and last name, Email address, username, and password.
Actors:	User
Pre-conditions:	User never signed up in the system before (no duplicate email address or username in the system).
Post-conditions:	User's account informations stored in the database.
Frequency of use:	One time only for each user.
Flow of events:	Actor action: System Response: 1.Enter first name 2.Enter last name 3.Enter Password 3. Check if the password is valid 4.Enter Email 4. Check duplicates/exist in database 5.Enter username 5. Check duplicates/exist in database 6.Click Create account 6. Store everything in database
Variations:	If the password is invalid or the email or username already exist in the database, a error will showed up and return to sign up page.
Notes and Issues:	
Developer Notes:	

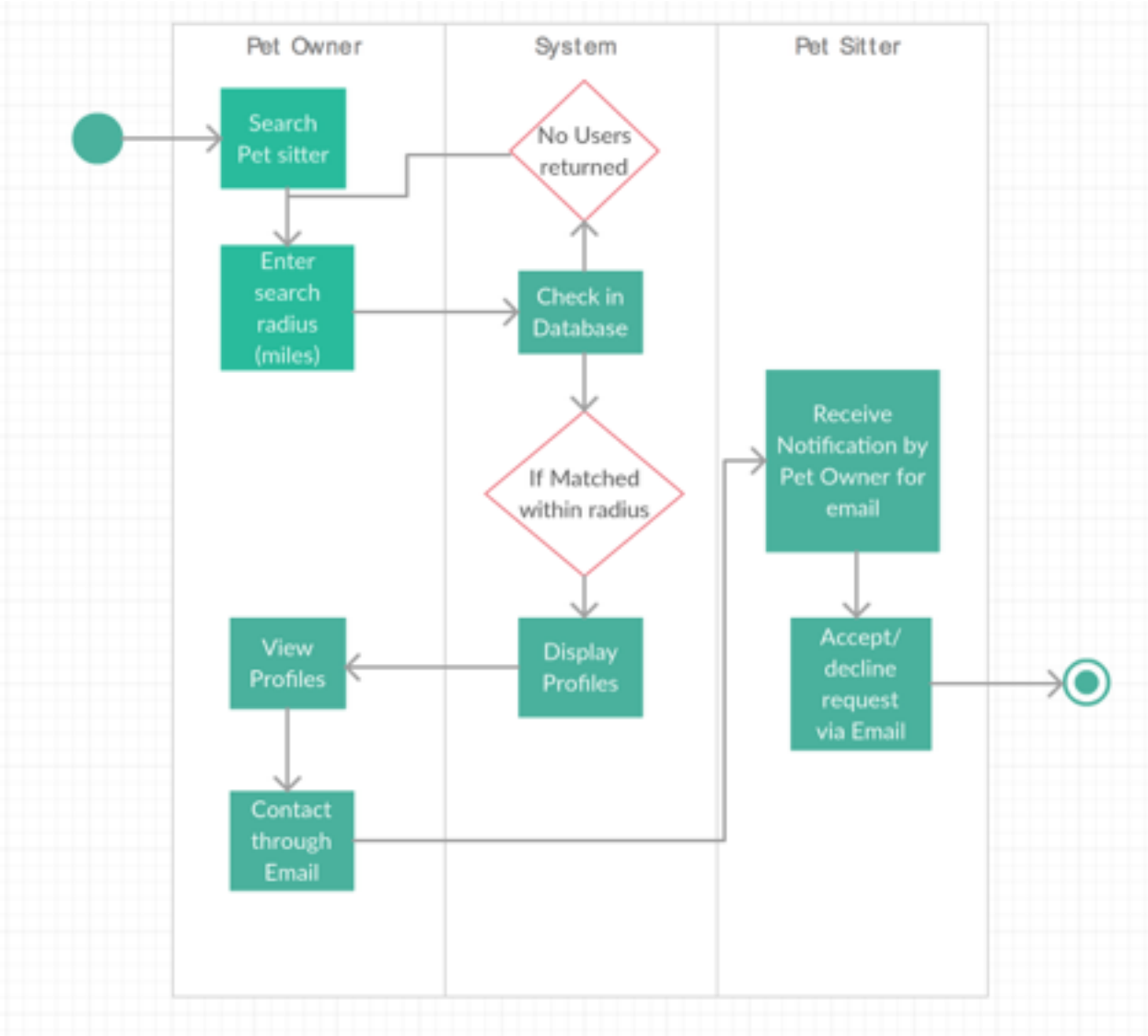
Use Case ID:	US-08
Use Case Name:	Search for pet sitter.
Descriptions:	User (pet owner) will search all available pet sitters around him/her and get contact informations of the pet sitter.
Actors:	Pet owner
Pre-conditions:	Pet owner have a valid address. Pet owner logged in and there is at least one available sitter around the pet owner.
Post-conditions:	Pet owner have pet sitter's contact information
Frequency of use:	Daily
Flow of events:	Actor action: System Response: 1.Enter address 2.Enter the search radius 2. Check available pet sitters within the radius 3. Display all available sitters' profile 4.Contact sitter through Email
Variations:	If there is no pet sitter around certain radius, it will go back to search page with a error "No matchups, please change radius and try again."
Notes and Issues:	
Developer Notes:	

Activity Diagram:

User Case And Requirement ID : US-01
Use Case Description :
As a user, I want to be able to create a profile (including the picture and a description for my pet.

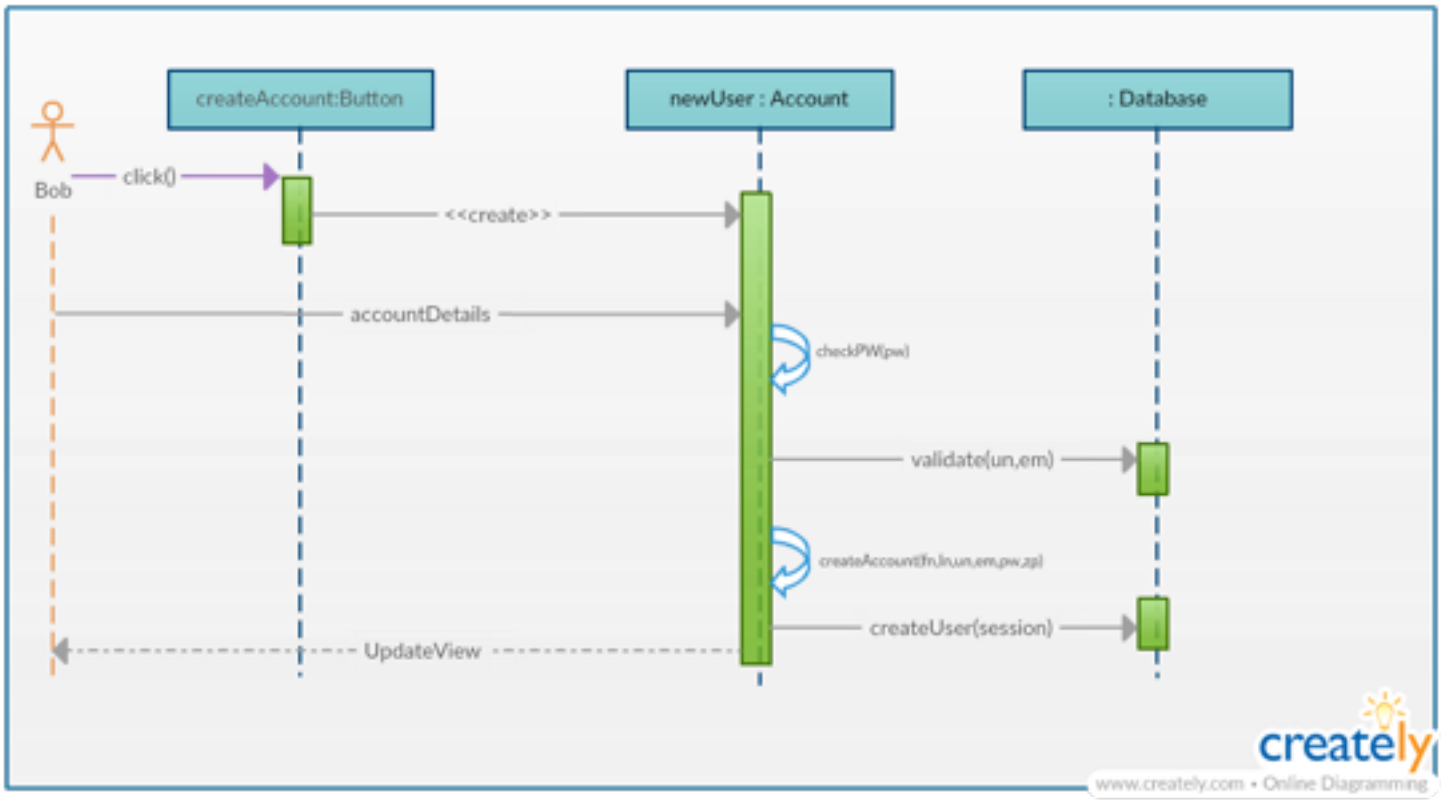


User Case And Requirement ID : US-08
Use Case Description :
As a user, I want to be able to search for pet sitters around me.

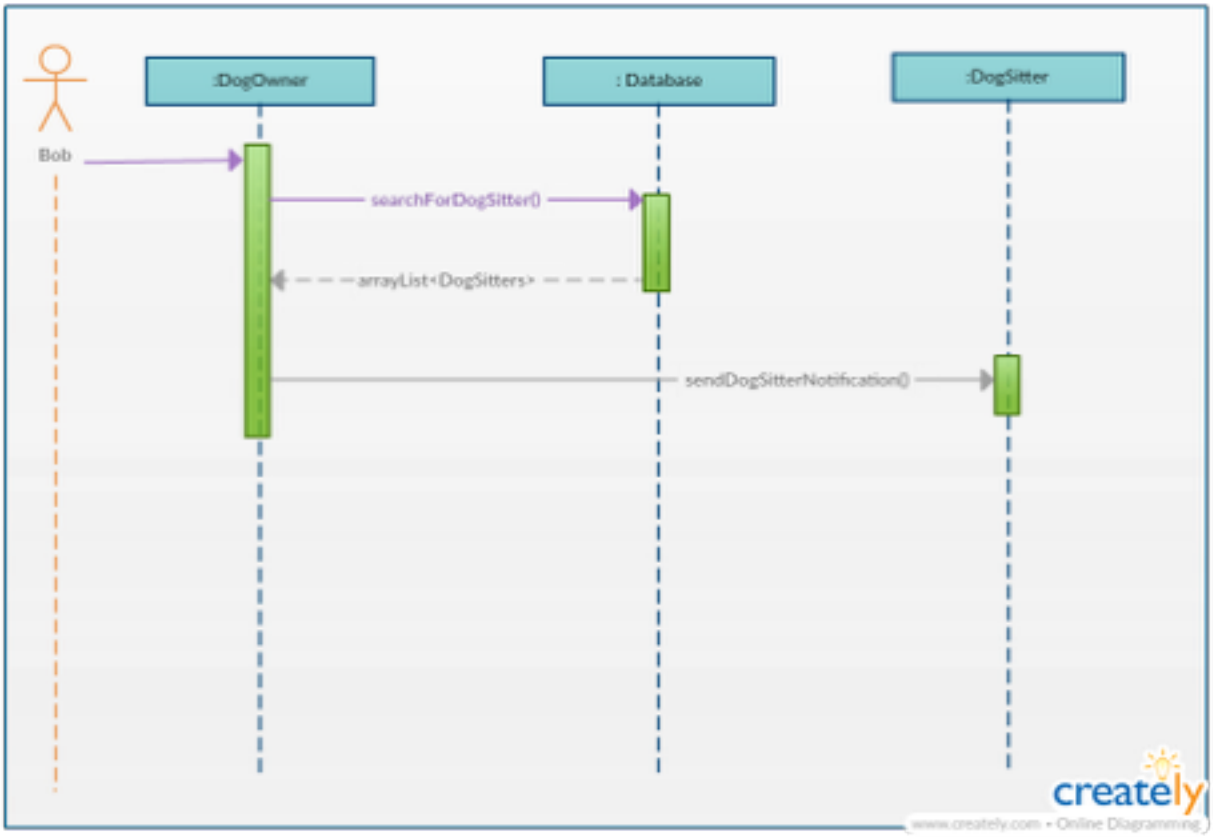


Sequence Diagram:

Use Case - 01



Use case - 08


:DogOwner
: Database
:DogSitter
Bob
searchForDogSitter()
arrayList<DogSitters>
sendDogSitterNotification()
creately
www.creately.com • Online Diagramming