

FOR YOUR LITTLE FRIEND



Everymal

ANIMAL HOSPITAL FINDER

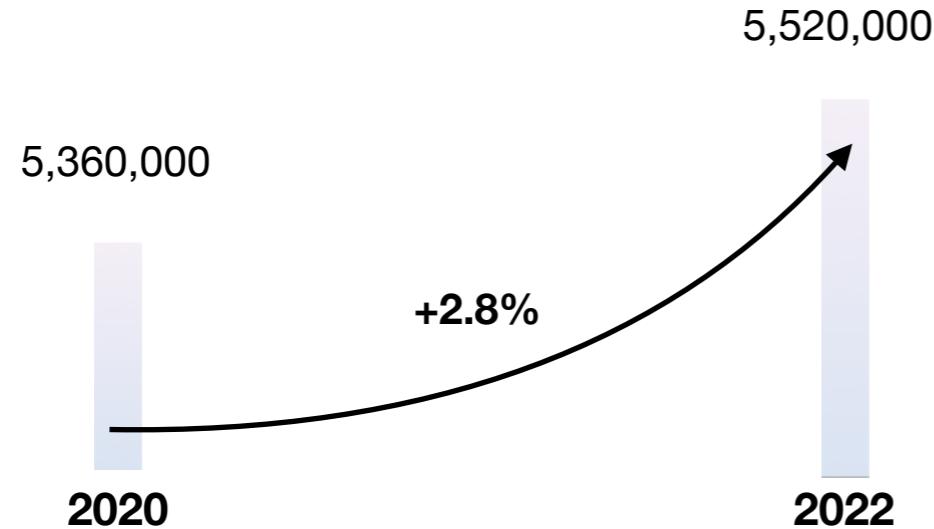
Portfolio

Project: Everymal

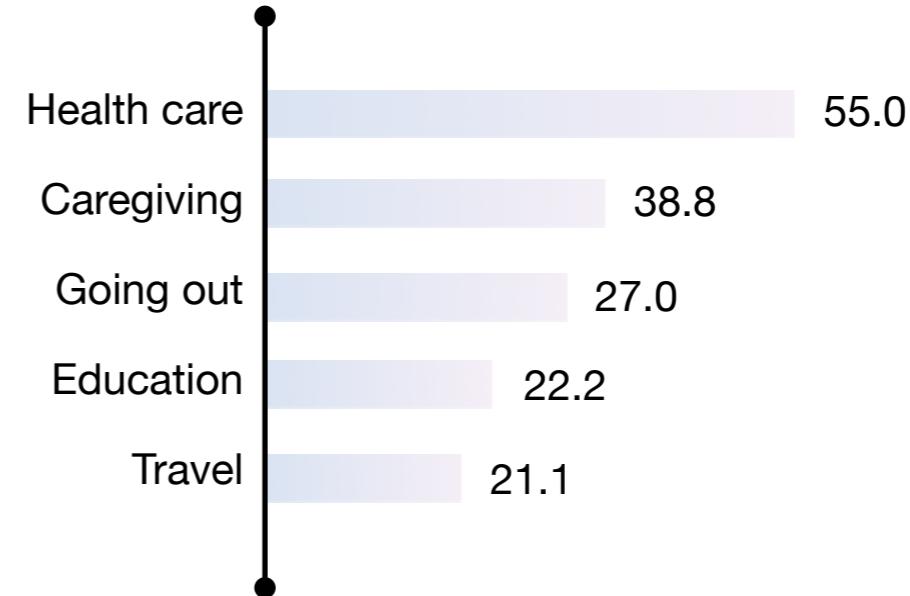
Goals

- Recognizing the absence of pet health management sites compared to the increasing trend of single-person households with pets, we aim to create a website that facilitates connections between various pet owners and veterinary clinics to address this gap.
- Creating a website that allows users to ask veterinarians simple questions directly, as well as search for veterinary clinics that possess necessary information for pet owners and assist with reservations.
- Providing pet taxi connections and 24-hour emergency room information tailored to single-person households.

Households with pets



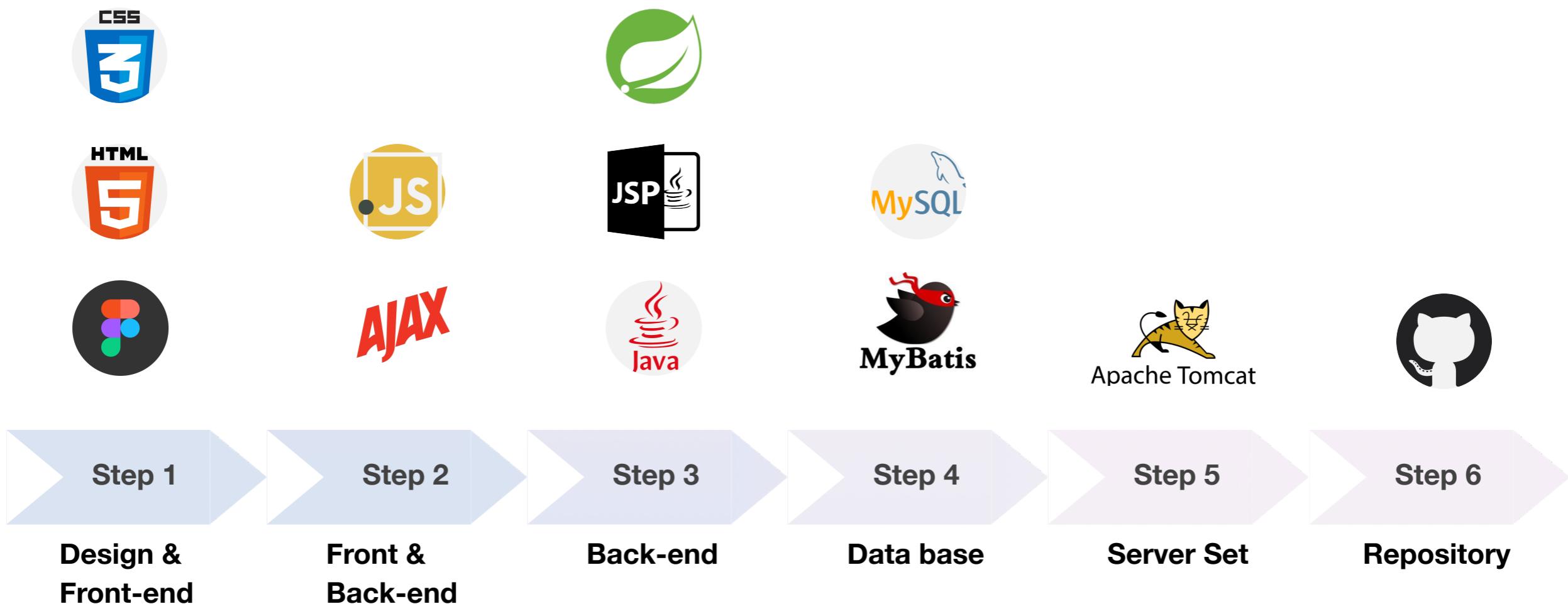
Recent interests regarding pet care by pet owners.



Source: "2023 Korean Pet Report" by KB

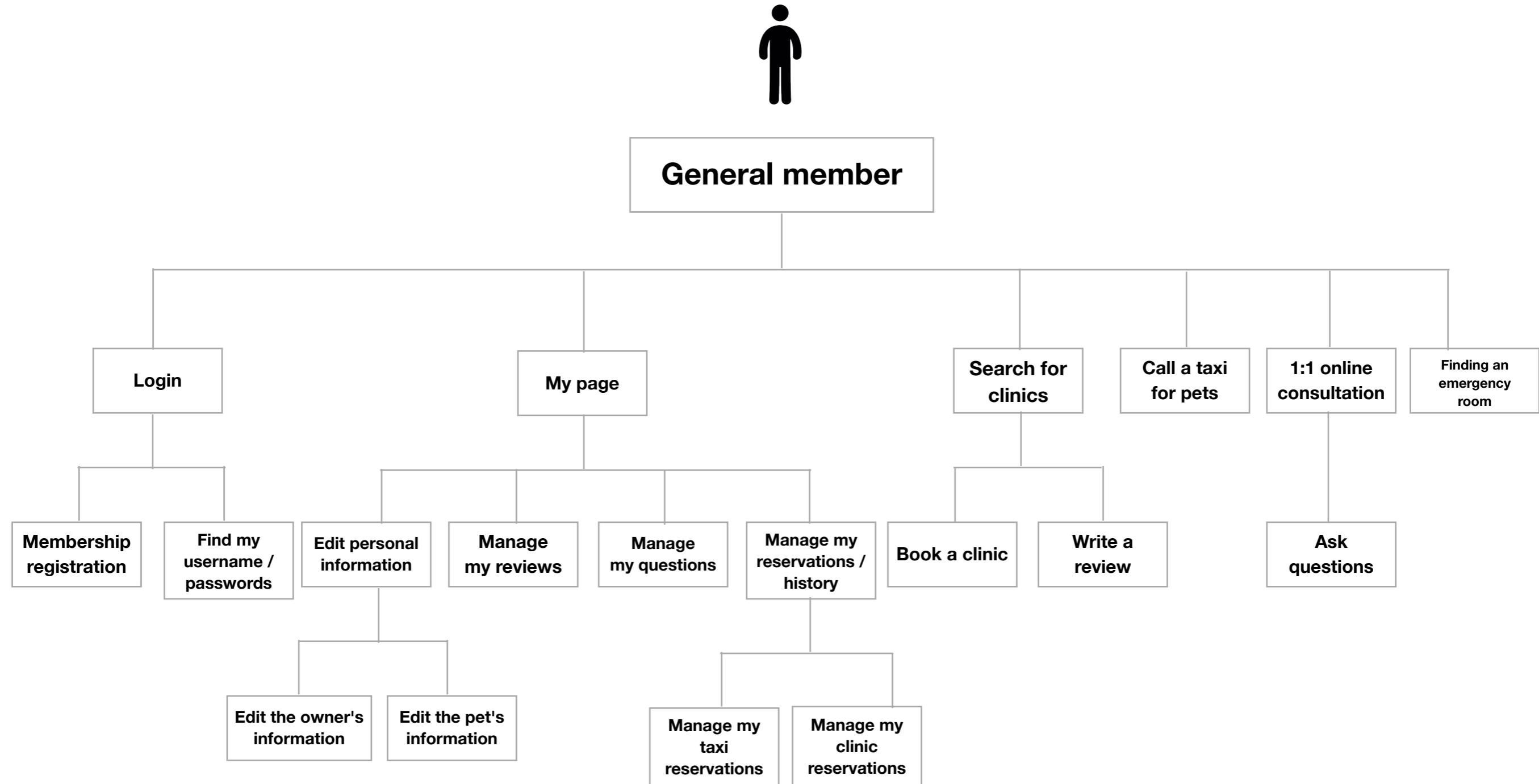
Project: Everymal

Development Language and Environment



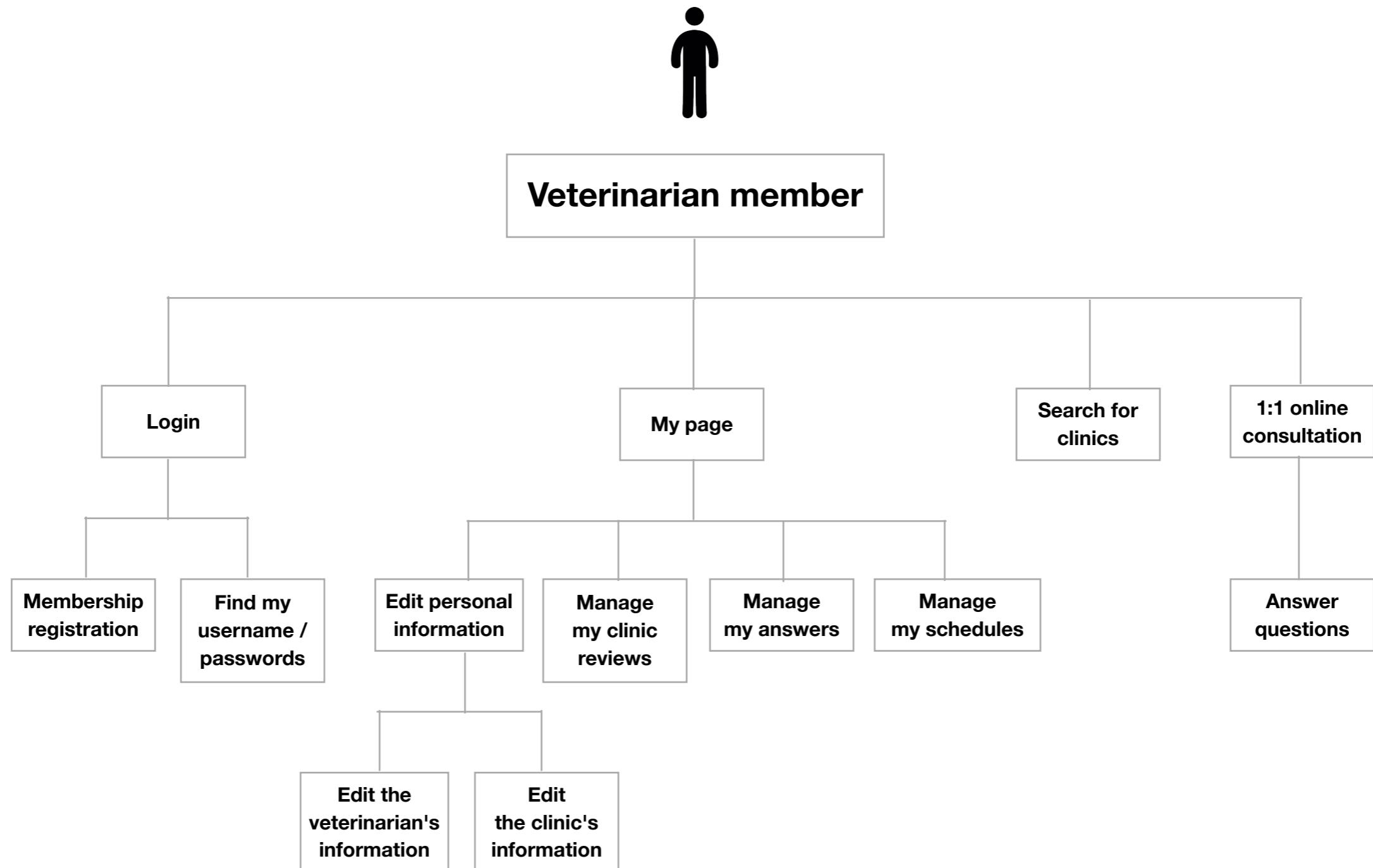
Project: Everymal

Use Case I



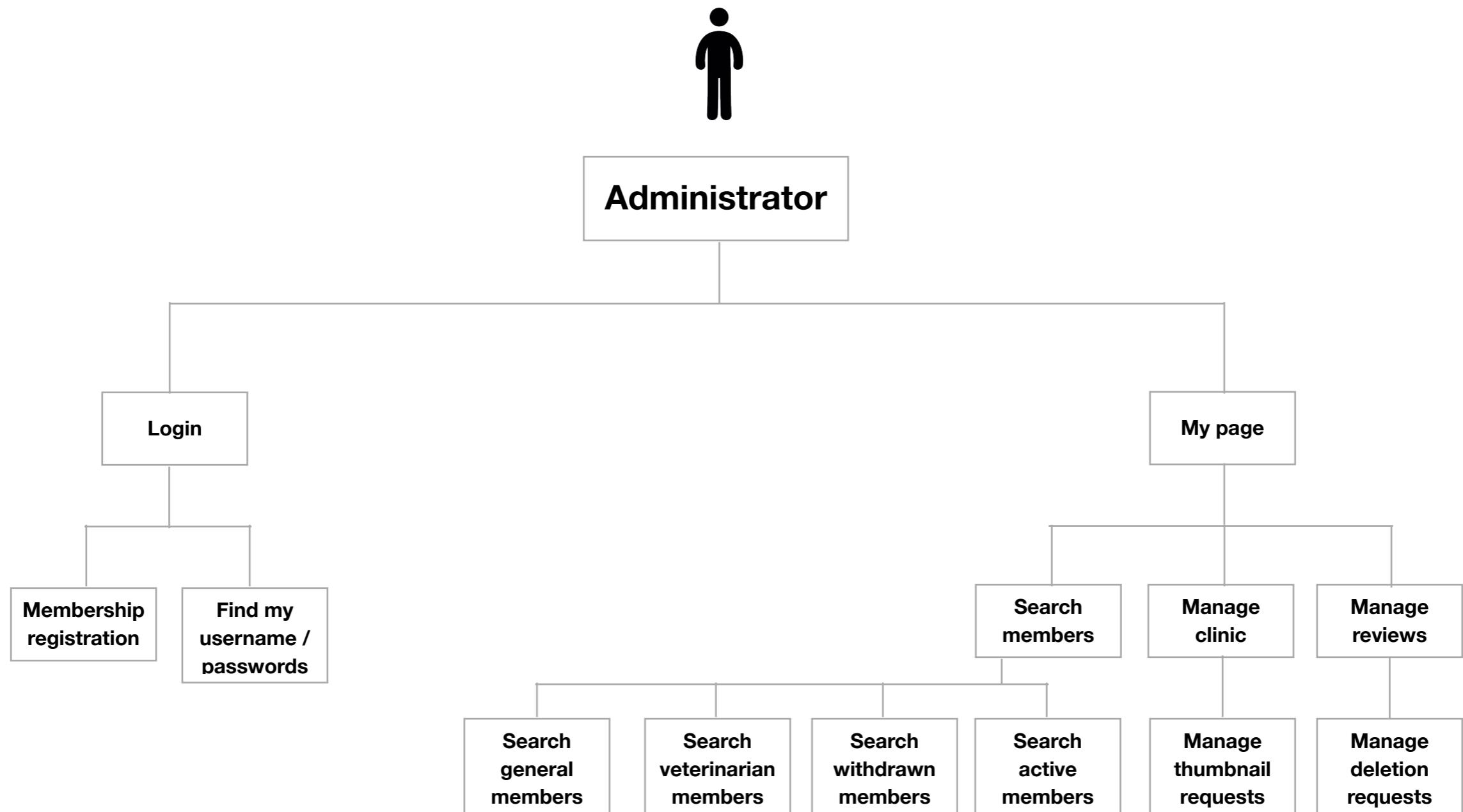
Project: Everymal

Use Case II



Project: Everymal

Use Case III



Project: Everymal

Software Requirements Specification I

	Details	URL	Function	MapperURL	Controller	Method	Service	DAO	Jsp	Table	Note
	Main page	http://127.0.0.1:8080/everymal/index.jsp									
General, veterinarian members	Sign-up Selection	http://127.0.0.1:8090/Everymal/join_choice.jsp	Selection for membership type	-	-	-	-	-	-		
general members	General membership registration	http://127.0.0.1:8090/Everymal/join_Member/join_user.jsp	General Membership Registration Form	-	memberController	-		memberDAO	join_user.jsp	GMEMBERTBL, PETINFOTBL	Using Kakao API for Address Search
			Username duplicate check	member/idChk.do	memberController	fn_idCheck()		memberDAO		GMEMBERTBL	
			Register a member	member/addMember.do	memberController	submit()		memberDAO		GMEMBERTBL	
			Register a pet	animal/addAnimal.do	animalController	submit()		animalDAO		PETINFOTBL	
			Add a pet in user's list	-	-	animalplus()		-		PETINFOTBL	
			Delete a pet in user's list	-	-	animalminus()		-		PETINFOTBL	
Veterinarian members	Veterinarian membership registration	http://127.0.0.1:8090/Everymal/join_Member/join_hospital.jsp	Veterinarian Membership Registration Form	-	hospitalController	-		hospitalDAO	join_hospital.jsp	HOSJOINTBL, HOSFILTERTBL	Using Kakao API for Address Search
			Username duplicate check	hospital/idChk.do	hospitalController	fn_idCheck()		hospitalDAO			
			Register a member	hospital/addHospital.do	hospitalController	submit()		hospitalDAO			
General members	Login	http://localhost:8090/everymal/login.jsp	Login Form	/login.jsp	LoginController	-	-	-	login.jsp	-	
		http://localhost:8090/everymal/logindb/	Login	/logindb/		-	-		redirect: /index.jsp	GMEMBERTBL	
		http://localhost:8090/everymal/logindb/logout	Logout	/logindb/logout		logout()	-		redirect: /index.jsp	-	
		http://localhost:8090/everymal/search_id.jsp	Username retrieval form	/search_id.jsp		-	-		search_id.jsp	-	
		http://localhost:8090/everymal/logindb/searchId	Username Retrieval Result	/logindb/searchId		searchId()	-		search_id.jsp	GMEMBERTBL	
		http://localhost:8090/everymal/search_pass.jsp	Password retrieval form	/search_pass.jsp		-	-		search_pass.jsp	-	
		http://localhost:8090/everymal/logindb/searchpass	Password retrieval result	/logindb/searchpass		searchpass()	-		search_pass.jsp	GMEMBERTBL	

Project: Everymal

Software Requirements Specification II

	Details	URL	Function	MapperURL	Controller	Method	Service	DAO	Jsp	Table	Note
My page (general member)	Manage general member's information	http://localhost:8090/everymal/userMypage/isValid.do	Modify profile picture	/modInfo.do	UserController	selectUserInfo()	UserService	UserDAO	myInfo.jsp	GMEMBERTBL	-
			Manage general member's information	/myInfo.do	userProimgController	updateUserInfo					-
	Manage a general member's reviews	http://localhost:8090/everymal/userMypage/myReview.do	View list of reviews	/myReview.do	UserController	selectAllReview()			myReview.jsp	GMEMBERTBL	-
			Delete a review	/viewReview.do		reviewcount()				REVIEWTBL	-
			Modify a review	/modMyReview.do		requestDelRev()					-
				/myRevDel.do		selectReview()					-
				/returnMyReview.do		updateUserMyreview()					-
	Manage a general member's questions	http://localhost:8090/everymal/userMypage/myQuestion.do	View list of posted questions	/myQuestion.do		selectAllQ()	HistoryService	HistroyDAO	myQustion.jsp	GMEMBERTBL	-
			Delete a posted question	/myQusDel.do		requestDelQus()			viewMyReview.jsp	BOARDTBL	-
			Modify a posted question								-
	Manage veterinarian clinic reservations	http://localhost:8090/everymal/history	view recent reservations	/viewHistory	HistoryController	hosResList()			myHosReserv.jsp	HOSRESTBL	-
			View previous reservations	/history		hosRecent()			userResView		-
			View details			selectHis()					-
	Manage pet taxi reservations	http://localhost:8090/everymal/history/myPetTaxiReserv.do	View recent taxi reservations	/myPetTaxiReserv.do		txResList()	HistoryService	HistroyDAO	myPetTaxiReserv.jsp	TAXIRESTBL	-
			View previous taxi reservations			txRecent()					-
Manage pet's information	Manage pet's information	http://localhost:8090/everymal/petinfo/myPetList.do	View my pets	/petinfo/myPetList.do	PetInfoController	selectPetList()	PetInfoDAO	PetInfoDAO	myPet_Info.jsp		
		http://localhost:8090/everymal/petinfo/modPet.do	Modify pet's information	/petinfo/modPet.do		updatePet()			redirect: myPet_Info.jsp		
		http://localhost:8090/everymal/petinfo/removePetInfo.do	Delete pet Information	/petinfo/removePetInfo.do		removePetInfo()			redirect: myPet_Info.jsp	petinfotbl	
		http://localhost:8090/everymal/petinfo/addPet.do	Add pet Information	/petinfo/addPet.do		insertPet()			redirect: myPet_Info.jsp		

Project: Everymal

Software Requirements Specification III

	Details	URL	Function	MapperURL	Controller	Method	Service	DAO	Jsp	Table	Note
My page (veterinarian member)	Confirm password	http://127.0.0.1:8080/everymal/hosMypageInfo	My page (password confirmation)	/hosMypageInfo/isValid.do	HosMypageInfoController	isValid()		HosMypageInfoDAO	hos_MypageInfo/hosMypage.jsp	HOSJOINTBL	
	Modify member's information	http://127.0.0.1:8080/everymal/hosMypageInfo/isValid.do	Retrieve veterinarian member information	/hosMypageInfo/myInfo.do	HosMypageInfoController	callhosInfo() selecthosInfo()	HosMypageInfoService	HosMypageInfoDAO	hos_MypageInfo/hosUserInformation.jsp	HOSJOINTBL	
		http://127.0.0.1:8080/everymal/hosMypageInfo/modhosInfo.do	Modify veterinarian member information	/hosMypageInfo/modhosInfo.do	HosMypageInfoController	modhosInfo()		HosMypageInfoDAO	hos_MypageInfo/hosUserInformation.jsp	HOSJOINTBL	Using Kakao API for Address Search
		http://127.0.0.1:8080/everymal/hosMypageInfo/myHosInfo.do	Retrieve clinic information	/hosMypageInfo/myHosInfo.do	HosMypageInfoController	callhosInfo() selecthosInfo()	HosMypageInfoService	HosMypageInfoDAO	hos_MypageInfo/hosInformation.jsp	HOSJOINTBL HOSFILTERTBL	
			Modify clinic's information	/hosMypageInfo/modmyHosInfo.do	HosMypageInfoController	modmyHosInfo()		HosMypageInfoDAO	hos_MypageInfo/hosInformation.jsp	HOSJOINTBL HOSFILTERTBL	
	Manage clinic's reviews	http://127.0.0.1:8090/everymal/hosReview	View list of reviews	/hosReview	HosReviewController	selectAllRev()	HosReviewService	HosReviewDAO	hos_ReviewInfo/hosReviewList.jsp	REVIEWTBL	
		http://127.0.0.1:8090/everymal/hosReview/viewReview.do	View detailed reviews			selectReview()			hos_ReviewInfo/hosViewReview.jsp	REVIEWTBL	
		http://127.0.0.1:8090/everymal/hosReview/hosRevDel.do	Request to delete reviews			requestDelRev()			hos_ReviewInfo/hosReviewList.jsp	REVIEWTBL	
	Manage answers	http://127.0.0.1:8090/everymal/hosReply/listMyReply.do	View list of posted answers	/hosReply	HosMyReplyController	selectAllRev()	HosMyReplyService	HosMyReplyDAO	hos_MyReplyInfo/listMyReply.jsp	BOARDTBL	
		http://127.0.0.1:8090/everymal/hosReply/viewReply.do	View detailed answers			selectReply()			qna_Board/qnaboard.jsp	BOARDTBL	
		http://127.0.0.1:8090/everymal/hosReply/hosReplyDel.do	Delete answers			deleteReply()			hos_MyReplyInfo/listMyReply.jsp	BOARDTBL	

Project: Everymal

Software Requirements Specification IV

	Details	URL	Function	MapperURL	Controller	Method	Service	DAO	Jsp	Table	Note
Search clinics	View clinics	http://localhost:8090/everymal/hosfilter	Filtering list of all clinics	/hosfilter	HosFilterController	listHos()	FilterService	FilterDAO	hos_List/hosFilter.jsp	hosfiltertbl, hosjointbl	Display clinic list based on user's selected filters
		http://localhost:8090/everymal/hosfilter	Filtering list of clinics sorted by most reviews	/hosfilter		revListHos()			hos_List/hosFilter.jsp	hosfiltertbl, hosjointbl, reviewtbl	Display clinic list sorted by most reviews based on user's selected filters
		http://127.0.0.1:8080/everymal/hosfilter	Paging list	/hosfilter/listHos.do	HosfilterController	selectAllActivles()	FilterDAO	hos_List/hospital_detail_page.jsp	HOSJOINTBL		
Detailed page for clinic	Detailed information of clinic	http://127.0.0.1:8090/Everymal/hospital/Hospital_detail.do?hos_code=\${hos_code}	View clinic details	/hospital/Hospital_detail.do	hospitalController HosMainImgCon HosReviewController	selectHos()	-	hospitalDAO	hos_List/hopital_detail_page.jsp	HOSJOINTBL	
		http://127.0.0.1:8090/everymal/hos_Reservation/reservationForm.jsp	Make a reservation	/reserv	ReservationController	hosReserv()	-	ReservationDAO	hos_Reservation/reservationForm.jsp	HOSRESTBL	
		http://127.0.0.1:8090/everymal/hospital/addReview.do	Write a review	/hospital	hospitalController	addreview()	-	hospitalDAO	hos_List/reviewPopup.jsp	REVIEWTBL	
Make a reservation		http://127.0.0.1:8080/everymal/hospital/reservationForm.do?hos_code=hos0001	Make a reservation	/hospital/reservation.do	hospitalController	hosReserv()	ReservationDAO	hos_Reservation/reservationForm.jsp	HOSRESTBL HOSJOINTBL GMEMBERTBL		
		http://127.0.0.1:8080/everymal/hospital/hos_Reservation/listPopup.jsp	Retrieve pet information	/hospital/hos_reservation/listPopup.jsp	hospitalController	animalList()					
24-hour Emergency Room		http://127.0.0.1:8080/everymal/emr_Page/emergency.jsp	Map search						emr_Page/emergency.jsp		

Project: Everymal

Software Requirements Specification V

	Details	URL	Function	MapperURL	Controller	Method	Service	DAO	Jsp	Table	Note
Pet taxi	Pet Taxi Search and Reservation	http://localhost:8090/everymal/petTaxiPage	Pet Taxi Search by Location	/petTaxiPage	PetTaxiController	taxiList()	petTaxiService	petTaxiDAO	petTaxiPage.jsp	TAXIINFOTBL	-
			Pet Taxi Reservation	/reservPetTaxi.do		taxiReserv()			taxiList.jsp	TAXIRESTBL	-
						getNewTxrCode()					-
24-hour Emergency Room	24-hour Emergency Room	http://127.0.0.1:8080/everymal/emr_Page/emergency.jsp	Map Search						emr_Page/emergency.jsp		Using Kakao API for Address Search
1:1 online consultation	Question Board	http://localhost:8090/everymal/board/listArticles.do	View Question Board	/board/listArticles.do	BoardController	listArticles()	BoardService	BoardDAO	qnaboardMain.jsp	BOARDTBL	-
		http://localhost:8090/everymal/board/viewArticle.do?q_code=N	View Question Board Detail	/board/viewArticle.do		viewArticle()			qnaboard.jsp		-
		http://localhost:8090/everymal/board/viewArticle.do?q_code=N	View Answer on Question Board	/board/viewArticle.do		viewArticle()			qnaboard.jsp		-
		http://localhost:8090/everymal/board/articleForm.do	Posting on Question Board	/board/articleForm.do		addArticle()			qnaboardWrite.jsp		-
		http://localhost:8090/everymal/board/modArticle.do	Edit Question Board	/board/modArticle.do		modArticle()			qnaboard.jsp		-
		http://localhost:8090/everymal/board/removeArticle.do	Delete Question Board	/board/removeArticle.do		removeArticle()			redirect:qnaboardMain.jsp		-
		http://localhost:8090/everymal/board/addReply.do	Write Answer	/board/addReply.do		addReply()			qnaboard.jsp		-
		http://localhost:8090/everymal/board/modReply.do	Modify Answer	/board/modReply.do		modArticle()			qnaboard.jsp		-
		http://localhost:8090/everymal/board/removeReply.do	Delete Answer	/board/removeReply.do		removeReply()			qnaboard.jsp		-

Project: Everymal

Database Design: Table Specifications I

TABLE NAME	GMEMBER						
COMMENT	Table of General Member Information						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	USER_CODE	User code of a general member	VARCHAR2	PK	NN	20	
2	USER_ID	User ID of a general member	VARCHAR2		NN	100	
3	USER_PWD	Password of a general member	VARCHAR2		NN	20	
4	USER_NAME	Name of a general member	VARCHAR2		NN	100	
5	USER_TEL	Contact information of a general member	VARCHAR2		NN	20	
6	USER_EMAIL	Email of a general member	VARCHAR2			150	
7	USER_ADDR1	Address of a general member	VARCHAR2			500	
8	USER_IMGNAM	Profile image of a general member	VARCHAR2			300	
9	J_DATE	Date of registration for general member	DATE		NN		sysdate
10	USER_ZIPCODE	Postal address of general member	VARCHAR2			10	
11	USER_ADDR2	Detail address of general member	VARCHAR2			500	
12	USER_ENABLED	Membership withdrawal status of general member	CHAR		NN	1	

TABLE NAME	PETINFOTBL						
COMMENT	Table of Pet Information						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	PET_CODE	Pet Code	VARCHAR2	PK	NN	20	
2	PET_NAME	Pet Name	VARCHAR2		NN	100	
3	PET_AGE	Pet Age	NUMBER			3	
4	PET_SEX	Pet Gender	CHAR			1	
5	PET_TYPES	Pet Types	VARCHAR2		NN	50	
6	PET_NUMBER	Pet Registration Number	VARCHAR2			50	
7	B_TYPE	Pet Blood Type	VARCHAR2			20	
8	PET_WEIGHT	Pet Weight	NUMBER			4,1	
9	PET_ETC	Other Information of Pet	VARCHAR2			2000	
10	USER_CODE	User code of a general member	VARCHAR2		NN	20	

Project: Everymal

Database Design: Table Specifications II

TABLE NAME	HOSJOINTBL						
COMMENT	Table of Hospital Registration						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	HOS_CODE	User code of a veterinarian member	VARCHAR2	PK	NN	10	
2	HOS_PRO	Profile image of a Hospital	VARCHAR2			300	
3	HOS_ID	User ID of a veterinarian member	VARCHAR2		NN	100	
4	HOS_USERNAME	Name of a veterinarian member	VARCHAR2		NN	100	
5	HOS_USERTEL	Contact information of a veterinarian member	VARCHAR2		NN	20	
6	HOS_PWD	Password of a veterinarian member	VARCHAR2		NN	20	
7	HOS_DNO	Business Registration Number	NUMBER		NN	25	
8	HOS_NAME	Hospital Name	VARCHAR2		NN	100	
9	HOS_TEL	Hospital Contact Information	VARCHAR2		NN	20	
10	HOS_ZIPCODE	Hospital Postal Address	VARCHAR2		NN	10	
11	HOS_ADDR1	Hospital Address	VARCHAR2		NN	500	
12	HOS_ADDR2	Hospital Detailed Address	VARCHAR2		NN	500	
13	HOS_EMAIL	Hospital Email Address	VARCHAR2			150	
14	HOS_INTRO	Introduction of the Hospital	VARCHAR2			1000	
15	HOS_THUM	Hospital Thumbnail	VARCHAR2			300	
16	HOS_ENABLED	Membership withdrawal status of veterinarian member	CHAR		NN	1	
17	HOS_STATUS	Thumbnail Registration Status	VARCHAR2			200	
18	J_DATE	Date of registration for veterinarian member	DATE		NN		sysdate

Project: Everymal

Database Design: Table Specifications III

TABLE NAME	HOSFILTERTBL						
COMMENT	Table of Hospital Filter						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	HOS_CODE	User code of a veterinarian member	VARCHAR2	PK	NN	20	
2	HOS_24	24-hour operation status	CHAR		NN	1	
3	HOS_365	Business days	VARCHAR2		NN	300	
4	HOS_TIME_S	Opening hours	VARCHAR2		NN	20	
5	HOS_TIME_E	Closing time	VARCHAR2		NN	20	
6	HOS_DOG	Dog	CHAR		NN	1	
7	HOS_CAT	Cat	CHAR		NN	1	
8	HOS_SMALL	Small animal	CHAR		NN	1	
9	HOS_FISH	Fish	CHAR		NN	1	
10	HOS_BIRD	Bird	CHAR		NN	1	
11	HOS_REP	Reptiles	CHAR		NN	1	
12	HOS_ETC	Other animals	CHAR		NN	1	
13	HOS_GS	General Surgery	CHAR		NN	1	
14	HOS_IM	Internal Medicine	CHAR		NN	1	
15	HOS_EM	Emergency	CHAR		NN	1	
16	HOS_RM	Rehabilitational Medecine	CHAR		NN	1	
17	HOS_VAC	Vaccination	CHAR		NN	1	

TABLE NAME	HOSIMGTBL						
COMMENT	Table of Hospital Image						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	HOS_CODE	User code of a veterinarian member	VARCHAR2	FK	NN	20	
2	HIMG1	Hospital Image No.1	VARCHAR2			300	
3	HIMG2	Hospital Image No.2	VARCHAR2			300	
4	HIMG3	Hospital Image No.3	VARCHAR2			300	

Project: Everymal

Database Design: Table Specifications IV

TABLE NAME	HOSRESTBL						
COMMENT	Table of Hospital Reservation						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	RES_CODE	Reservation code	VARCHAR2	PK	NN	20	
2	USER_NAME	Name of a general member	VARCHAR2		NN	100	
3	USER_TEL	Contact information of a veterinarian member	VARCHAR2		NN	20	
4	PET_NAME	Pet Name	VARCHAR2			100	
5	PET_AGE	Pet Age	NUMBER			3	
6	PET_SEX	Pet Gender	CHAR			1	
7	HOS_NAME	Hospital Name	VARCHAR2		NN	100	
8	RES_DATE	Preferred booking date	DATE		NN		sysdate
9	RES_TIME	Preferred booking time	VARCHAR2		NN	20	
10	HOS_SUB	Preferred medical specialty	VARCHAR2		NN	500	
11	RES_ETC	Requirements and additional information	VARCHAR2			2000	
12	HOS_CODE	User code of a veterinarian member	VARCHAR2	FK	NN	20	
13	PET_CODE	Pet code	VARCHAR2			20	
14	USER_CODE	User code of a general member	VARCHAR2	FK	NN	20	
15	RECEIPT_DATE	Reservation reception date	DATE		NN		sysdate
16	USER_ID	User ID of a general member	VARCHAR2		NN	100	
17	HOS_THUM	Hospital Thumbnail	VARCHAR2			300	
18	B_TYPE	Pet Blood Type	VARCHAR2			20	
19	PET_WEIGHT	Pet Weight	NUMBER			4,1	
20	PET_TYPES	Pet Types	VARCHAR2		NN	50	
21	PET_NUMBER	Pet Registration Number	VARCHAR2			50	

Project: Everymal

Database Design: Table Specifications V

TABLE NAME	BOARDTBL						
COMMENT	Table of Board						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	Q_CODE	Question code	NUMBER	PK	NN	10	
2	Q_DATE	Question date	DATE		NN		sysdate
3	USER_ID	User ID of a general member	VARCHAR2		NN	100	
4	Q_TITLE	Question title	VARCHAR2		NN	1000	
5	Q_TEXT	Question content	VARCHAR2		NN	4000	
6	A_CODE	Answer cod	NUMBER			10	
7	A_DATE	Answer date	DATE				sysdate
8	HOS_NAME	Hospital Name	VARCHAR2			150	
9	A_TITLE	Answer title	VARCHAR2			1000	
10	A_TEXT	Answer content	VARCHAR2			4000	
11	USER_CODE	User Code of a general member	VARCHAR2		NN	20	
12	PETRACE	Animal to inquire about	VARCHAR2		NN	50	
13	HOS_ID	User ID of a veterinarian member	VARCHAR2			100	
TABLE NAME	REVIEWTBL						
COMMENT	Table of Review of Hospital						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	RV_CODE	Review code	VARCHAR2	PK	NN	20	
2	RV_TITLE	Review title	VARCHAR2		NN	2000	
3	RV_TEXT	Review content	VARCHAR2		NN	4000	
4	RV_RATE	Review rating	NUMBER		NN	2	
5	RV_DATE	Review date	DATE		NN		sysdate
6	HOS_NAME	Hospital name	VARCHAR2		NN	100	
7	USER_ID	User ID of a general member	VARCHAR2		NN	100	
8	RV_DELREQ	Review selection status	CHAR			1	0
9	RES_CODE	Reservation code	VARCHAR2		NN	20	
10	HOS_CODE	User code of a veterinarian member	VARCHAR2		NN	20	

Project: Everymal

Database Design: Table Specifications VI

TABLE NAME	TAXIINFOTBL						
COMMENT	Table of Taxi driver's Information						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	TX_CODE	Taxi code	VARCHAR2	PK	NN	20	
2	TX_NUMBER	Taxi number	VARCHAR2		NN	50	
3	TX_NAME	Taxi driver's Name	VARCHAR2		NN	100	
4	TX_TEL	Taxi driver's contact number	VARCHAR2		NN	20	
5	TX_LOCAL	Taxi driver's area	VARCHAR2		NN	500	
6	TX_IMG	Image of Taxi driver	VARCHAR2		NN	300	
7	TX_INTRO	Introduction of Taxi driver	VARCHAR2			1000	

TABLE NAME	TAXIRESTBL						
COMMENT	Table of Pet taxi Registration						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	TX_RES_CODE	Taxi reservation code	VARCHAR2	PK	NN	20	
2	TX_RESERV_DAY	Reservation date	DATE		NN		sysdate
3	TX_DEP	Departure point	VARCHAR2		NN	500	
4	TX_ARR	Destination	VARCHAR2		NN	500	
5	USER_CODE	User code of a general member	VARCHAR2		NN	20	
6	TX_CODE	Taxi code	VARCHAR2		NN	20	
7	TX_NUMBER	Taxi number	VARCHAR2		NN	50	
8	TX_NAME	Taxi driver's name	VARCHAR2		NN	100	
9	TX_TEL	Taxi driver's contact number	VARCHAR2		NN	20	
10	RES_NAME	Name of the person who made the reservation	VARCHAR2		NN	100	
11	TX_RESTIME	Taxi usage time	VARCHAR2		NN	20	

Project: Everymal

Database Design: Table Specifications VII

TABLE NAME	HOSRMTBL						
COMMENT	Table of Hospital Thumbnail Request						
No	Column Name	Comments	Data Type	KEY	NULL	SIZE	DEFAULT
1	RM_CODE	Request Code	VARCHAR2	PK	NN	20	
2	RM_CHK	Application Status	CHAR		NN	1	
3	RM_DATE	Application Date	DATE		NN		sysdate
4	RM_STATUS	Status of progress	VARCHAR2		NN	100	
5	HOS_NAME	Hospital name	VARCHAR2		NN	100	
6	HOS_ID	User ID of a veterinarian member	VARCHAR2		NN	100	
7	HOS_CODE	User code of a veterinarian member	VARCHAR2		NN	20	
8	RM_SAY	Administrator's guide	VARCHAR2			4000	

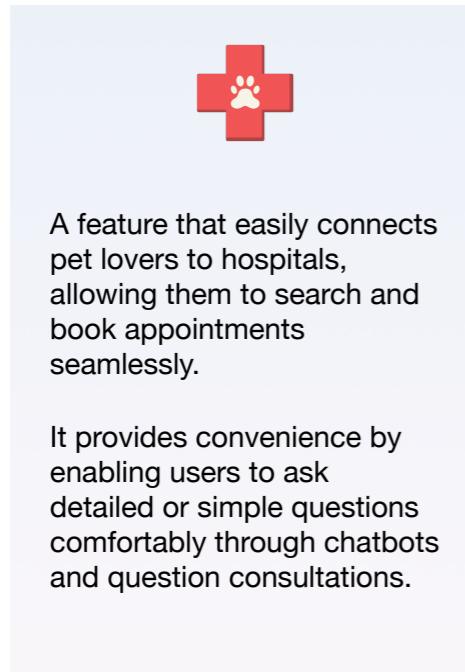
Project: Everymal

데이터베이스 설계: 다이어그램

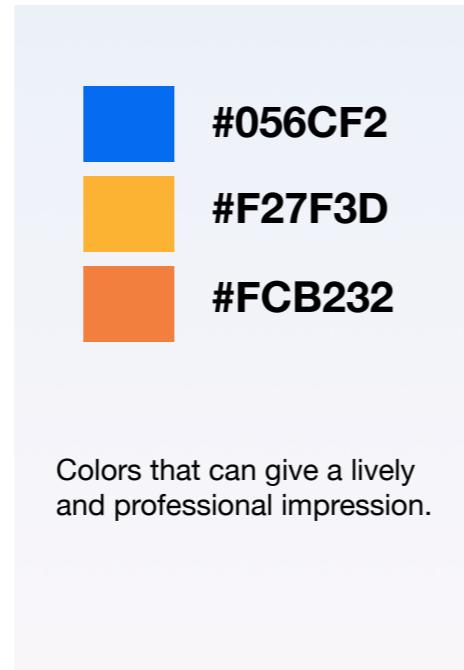


Project: Everymal

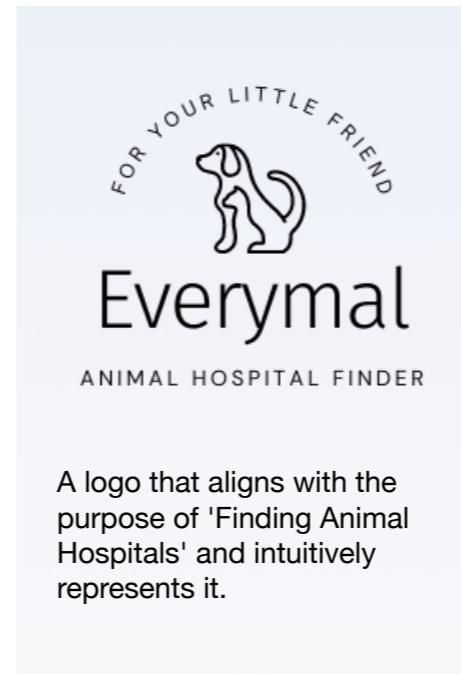
Style Concept



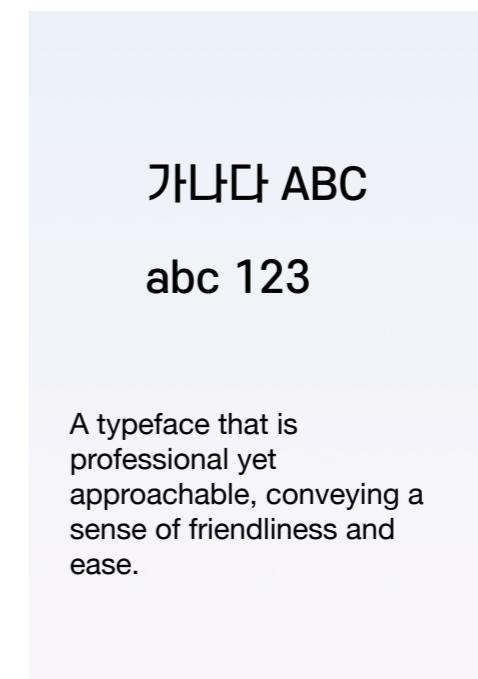
Concept



Color



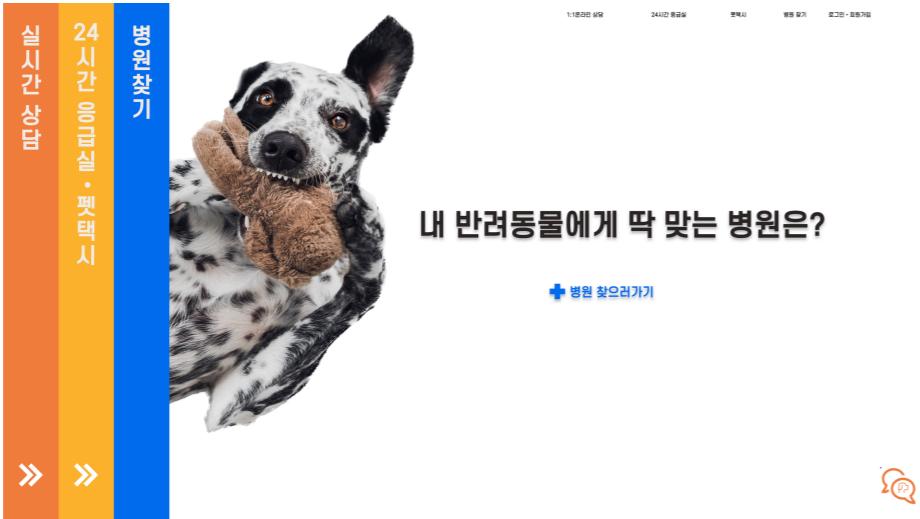
Logo



Font

Project: Everymal

Main page



When "Find 24-hour Emergency Room / Pet taxi" button is clicked

When clicking on the "Find 24-hour Emergency Room Near Me" button, it will direct a user to the 24-hour emergency page.

When clicking on the "Book Pet Taxi" button, it will direct a user to the pet taxi reservation page.

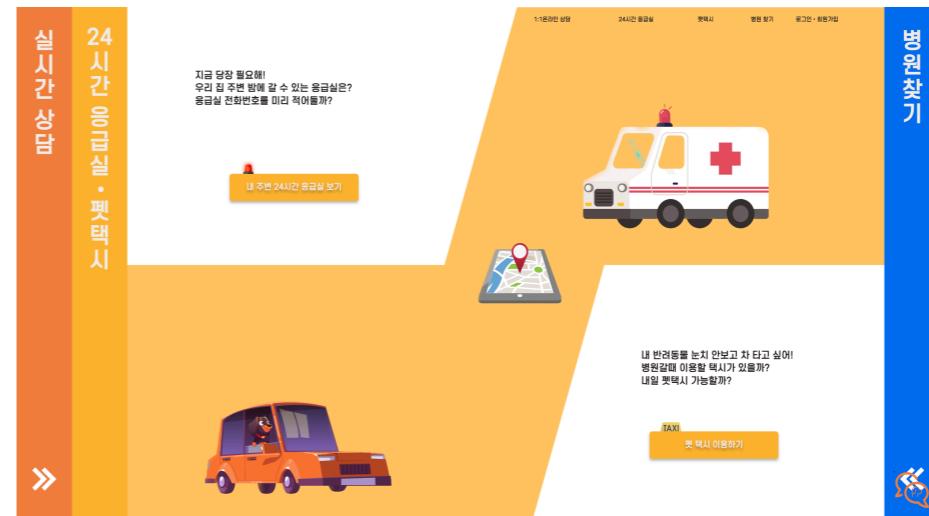


Main page / when "Clinic search" button is clicked

Placing the "Clinic Search" button and the search page prominently in the center of the homepage to intuitively express the purpose of the site.

Positioning the menu at the top for easy access to 1:1 Online Consultation, 24-hour Emergency Room, Pet Taxi, Hospital Search, and Login/Registration.

Implementing a chatbot feature accessible by clicking on the bottom-right image, allowing for simple symptom searches based on pet type.



When "Real-time consultation" button is clicked

Real-time consultations are conducted through live chat

The current number of posted questions and answers are displayed, and clicking on the 'Ask Now' button directs a user to the 1:1 online consultation page

Project: Everymal

Login/Password & UserID Recovery Page

The screenshot shows the main login interface. At the top, there are links for '1:1 대화문 상담' (1:1 Conversation Consultation), '24시간 응급실' (24-hour Emergency Room), '병원 찾기' (Find Hospital), and '로그인 · 회원가입' (Login · Sign Up). Below these are tabs for '회원' (Member) and '병원' (Hospital). The main form contains fields for '아이디' (ID) and '비밀번호' (Password), both with placeholder text. Below the fields are buttons for '로그인' (Login), '회원가입' (Sign Up), '아이디 찾기' (Find ID), and '비밀번호 찾기' (Find Password).

When clicking on login

After selecting a type of membership (veterinarian member or general member), the login function operates accordingly.

This screenshot shows the password recovery process for a 'general member'. It features two sections: '일반회원' (General Member) and '병원회원' (Hospital Member). The '일반회원' section includes fields for '이름' (Name), '아이디' (ID), and '이메일 주소' (Email Address). The '병원회원' section includes fields for '병원회원' (Hospital Member), '관리자 이름' (Administrator Name), '아이디' (ID), and '사업자등록번호' (Business Registration Number). A large purple button labeled '비밀번호 찾기' (Find Password) is prominently displayed at the bottom.

When clicking on Find my password

After selecting a type of membership (veterinarian member or general member), the password can be retrieved using the name, username, and email address

This screenshot shows the user ID recovery process for a 'general member'. It has two sections: '일반회원' (General Member) and '병원회원' (Hospital Member). The '일반회원' section contains fields for '이름' (Name) and '이메일 주소' (Email Address). The '병원회원' section contains fields for '병원회원' (Hospital Member), '사업자등록번호' (Business Registration Number), and '사업자등록번호' (Business Registration Number). Buttons for '아이디 찾기' (Find ID) and '비밀번호 찾기' (Find Password) are located at the bottom. The footer of the page includes the 'Everymal' logo, 'ANIMAL HOSPITAL FINDER' text, and contact information: 'Everymal', '1:1 대화문 상담 123-123-1234', '24시간 응급실 123-123-1234', '병원 찾기', '로그인', '회원가입', '이메일', '전화번호', and '주소'.

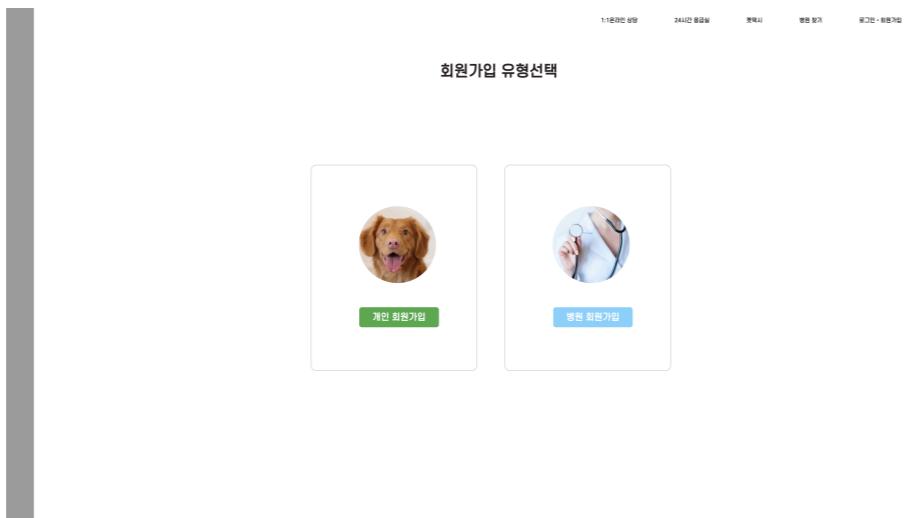
When clicking on Find my username

After selecting a type of membership (veterinarian member or general member), the username can be retrieved using the name and email address



Project: Everymal

Sign-up Page I



When "Sign Up" is clicked,

Implement the sign-up functionality according to whether the user is a veterinarian member or an individual member.

Input of Basic Member Information

Name/Username/Password/Confirm
Password/Contact number/Address/Email

회원정보

성명* 중길동
아이디* everymal ID중복체크
비밀번호* 영문/숫자 4-8자 이내
비밀번호 확인* 한번 더 입력해주세요
연락처* 선택
주소* 우편번호찾기
이메일* everymal@gmail.com

회원가입

반려동물 정보

반려동물 이름* 도리
반려동물 나이*
반려동물 성별* ♀ ♂ 선택
반려동물 종류* 선택
반려동물을 등록번호* 선택
반려동물을 혈액형* 선택
반려동물을 몸무게*
반려동물을 빠이시창

회원가입

Everymal
FOR YOUR LITTLE FRIEND
ANIMAL HOSPITAL FINDER
11:02:00 AM 24시간 응급실 환자 찾기 회원 찾기 로그인
고객센터
T. 1111-2222 (08:00 ~ 19:00) E. everymal@gmail.com

Input of Pet Information

Pet name/age/gender/species/registration number/blood type/weight/special notes

Additional pet information can be added or deleted using the + and - buttons.

Project: Everymal

Sign-up Page II

The image displays two side-by-side screenshots of the Everymal sign-up process. Both screens are titled "회원가입" (Member Sign-up) and feature a "초기화" (Reset) button at the bottom right.

Left Screen (Hospital Information):

- 병원정보 (Hospital Information):** Fields include 병원명 (Hospital Name), 관리자아이디 (Administrator ID), 관리자이름 (Administrator Name), 관리자연락처 (Administrator Contact), 비밀번호 (Password), 비밀번호 확인 (Confirm Password), 사업자등록번호 (Business Registration Number), and 병원 연락처 (Hospital Contact).
- A note at the top right states: "※ 반드시 입력해야 하는 항목입니다." (Fields that must be entered).
- Below the form is a logo for "Everymal ANIMAL HOSPITAL FINDER" featuring a small dog icon.
- At the bottom are navigation links: 1:1문의인 상담, 24시간 응급실, 찾으시, 회원 찾기, 로그인, 회원가입, 찾으시, 회원 찾기, 로그인.

Right Screen (Member Information):

- 회원가입 (Member Sign-up):** Fields include 이메일 (Email), 병원영업일 (Business Days), 영업시간 (Business Hours), 진료종목 (Medical Specialties), and 진료소개글 (Hospital Introduction).
- Below the form is a logo for "Everymal ANIMAL HOSPITAL FINDER" featuring a small dog icon.
- At the bottom are navigation links: 1:1문의인 상담, 24시간 응급실, 찾으시, 회원 찾기, 로그인, 회원가입, 찾으시, 회원 찾기, 로그인.

Input of Basic Member Information and Hospital Information

Hospital name/ID/Name/Contact number/Password/Confirm password/
Business registration number/Hospital contact number/Hospital address/
Email/Hospital business days>Select business days/Business hours/Treated
animal species/Medical specialties/Hospital introduction information

Project: Everymal

My Page: General Member I

This screenshot shows a password confirmation step. It features a large green rectangular box highlighting the central input field. The text inside the box reads: "회원님의 회원정보를 안전하게 보호하기 위해 비밀번호를 한번 더 확인해주세요. 비밀번호가 타인에게 노출되지 않도록 주의해 주세요." Below the text is a password input field labeled "비밀번호" and "비밀번호 입력". At the bottom are two buttons: "취소" (Cancel) and "확인" (Confirm).



Password Validation Check

Information can be modified after confirming the member's password.

This screenshot shows the member profile page after password validation. The "비밀번호 확인" field now contains the user's input. The rest of the profile information remains the same as the previous screenshot.

This screenshot shows the member profile page after updating information. The "비밀번호 확인" field is empty again. The "연락처" field now contains "010-12345678". The "주소" field now contains "23035 우편번호 찾기" and "인천 강화군 강화읍 남면간길 2-1aa". The "이메일" field now contains "slendunk@gmail.com". At the bottom, there are two buttons: "프로필 등록" (Profile Register) and "수정" (Modify).

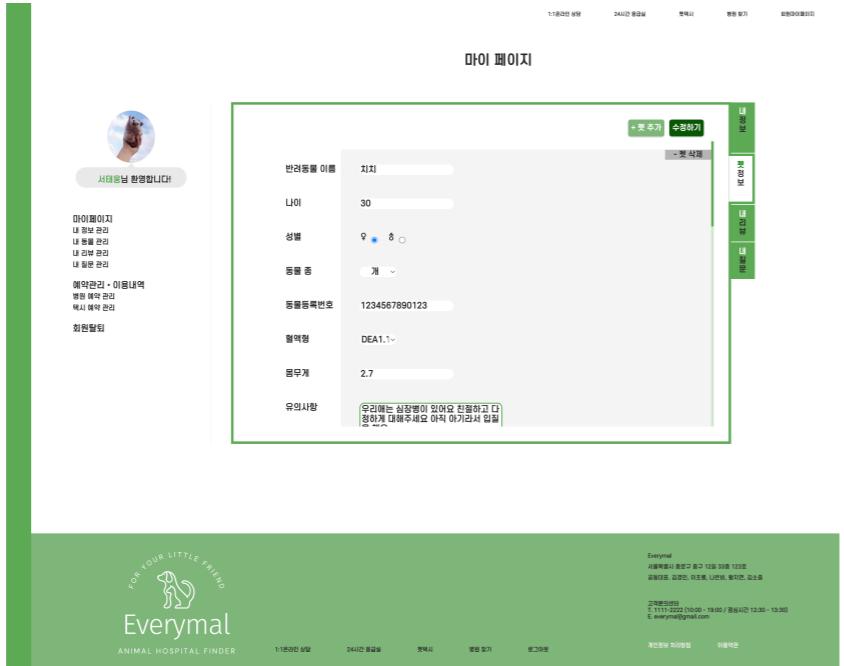
Member Information Update

password/contact information/address/email/profile picture modification page.

Information can be modified after password validity check.

Project: Everymal

My Page: General Member II



Member Pet Information Management

A page where you can add/modify/delete member's pet information.

- Functions to add/modify/delete pet name/age/gender/species/registration number/blood type/weight/notes.
- Clicking on the "Add Pet" button allows you to add one pet's information at a time.
- Clicking on the "Modify" button allows you to apply all modifications at once.
- Clicking on the "Delete Pet" button allows you to delete all information of the corresponding pet.

Project: Everymal

My Page: General Member III

선택	등록 번호	별점	리뷰제목	등록일
<input type="checkbox"/>	rv0007	★★★	조금 더 친절하게 대화주세요 ㅠ	2023-05-16
<input type="checkbox"/>	rv0006	★★★★★	친절한 선생님들!	2023-05-15
<input type="checkbox"/>	rv0012	★★	한달 뒤에 이사를 가게 됐서 아쉬워요.	2023-05-14
<input type="checkbox"/>	rv0011	★	제가 이 병원 단점이라 잘 아는데 한장 바꿔고 나서	2023-05-13
<input type="checkbox"/>	rv0010	★★★	처음 보는 선생님이 계시던데...▶-	2023-05-12
<input type="checkbox"/>	rv0009	★★★	이 병원 이용했었는데...잘 워라고 해야할지	2023-05-11
<input type="checkbox"/>	rv0008	★★★★★	보호자와 반려동물을 모두 만족하는 최상의 서비스를 제공...	2023-05-10
<input type="checkbox"/>	rv0005	★★★★★	최고예요 항상 이 병원만 이용해요	2023-05-09

Managing Member Reviews

A page for members to manage their reviews, allowing them to edit or delete all reviews they have written.

- Clicking on a review will take a user to a specific review details page.

Project: Everymal

My Page: General Member IV

The screenshot displays the 'My Page: General Member IV' interface. On the left, there's a sidebar with a profile picture and a 'Veterinarian responds' button. Below it are links for 'My Profile', 'Member Management', 'Pet Management', 'Consultation Management', and 'Logout'. The main content area has two main sections:

- 1:1 Online Consultation Question Management:** A table listing 20 consultation questions. Each row includes a checkbox, question ID, title, date, and response status. A vertical navigation bar on the right side of the table has buttons for 'New', 'Edit', 'Delete', and 'Detail'.
- 1:1 Online Consultation:** Two examples of consultation threads. Each thread shows a question from a member and a response from a veterinarian. Buttons for 'Edit' and 'Delete' are located at the bottom of each thread.

At the bottom of the page, there's a footer with the 'Everymal' logo, address (Seoul, Jongno-gu, 123-123), phone number (T. 1111-2222), email (E. everymal@gmail.com), and business hours (10:00 - 19:00 / 12:30 - 13:30). It also lists 'Animal Hospital Finder', 'Veterinarian Information', and 'Member Information'.

1:1 Online Consultation Question Management

A page where you can edit/delete all consultation questions you have written.

- Clicking on a question navigates to a specific question detail page.
- Only questions written by the member can be edited/deleted.

Project: Everymal

My Page: General Member V

The screenshot displays a user profile page for a general member. At the top, there are navigation links: 1:1문의(1:1 Inquiry), 24시간 응급실(24-hour Emergency Room), 콘택시(Contact), and 흥정 찾기(Find Bidding). Below these are sections for '예약관리 이용내역' (Reservation Management Usage History) and '최근 예약 정보' (Recent Reservation Information). The 'Recent Reservation Information' section shows a photo of a medical room and details: 예약 일시 (Reservation Date: 2023-09-05), 예약 접수 (Reservation Received: 2023-03-19), and 예약 병원 (Hospital: 일곡동물병원). Below this is a table for '지난 이용 내역' (Past Usage History) with rows for res0001 to res0005. A green sidebar on the left contains a profile picture placeholder, a '서브유저 환경입니다!' (Subuser Environment) message, and a '마이페이지' (My Page) menu with options like 마이페이지, 내 정보 관리, 내 동물 관리, 내 고양 관리, and 내 강아지 관리. At the bottom, there's a footer with the 'Everymal FOR YOUR LITTLE FRIEND' logo, address (서울특별시 종로구 종로 12길 38번 11층), contact (고객센터 T. 1111-2222 (10:00 - 19:00 / 평일시간 12:30 - 13:30) E. everymal@gmail.com), and links for 회원가입 (Join), 이용약관 (Terms of Service), and 개인정보 처리방침 (Privacy Policy).

Animal Hospital Reservation Management

A page where you can manage all booked hospital information.

- Clicking on a hospital navigates to the reservation detail page.

This screenshot shows a detailed reservation page for a specific appointment. At the top, there are navigation links: 1:1문의(1:1 Inquiry), 24시간 응급실(24-hour Emergency Room), 콘택시(Contact), and 흥정 찾기(Find Bidding). The main content area is titled '예약 상세정보' (Reservation Detailed Information) with the appointment ID N0127149137. It includes fields for 이름 (Name: 서태웅), 연락처 (Contact: 010-32690288), 반려동물 이름 (Pet Name: 카지), 등급 (Grade: 개), 나이 (Age: 30), 반려동물 성별 (Pet Gender: ♀), 반려동물 등록번호 (Pet Registration Number: 1234567890123), 반려동물 혈액형 (Pet Blood Type: DEA1.1), 반려동물 몸무게 (Pet Weight: 1234567890123), 병원 이름 (Hospital Name: 일곡동물병원), and 회장 예약날짜 (President Appointment Date: 2023. 09. 05.). Below this is a summary table for 'Everymal ANIMAL HOSPITAL FINDER' with columns for 예약코드 (Appointment Code), 반려동물 혈액형 (Pet Blood Type), 반려동물 몸무게 (Pet Weight), 병원 이름 (Hospital Name), 회장 예약날짜 (President Appointment Date), 오전 05:00 (Morning 05:00), and 친절 경력 (Kindness Experience). A note at the bottom states: 고생연예가 좋은 영양제와 애완동물을 맞추고 싶습니다. 신장에 무리가 가진 애견까요? (Good doggy wants to find a good nutrition and pet food. Is your dog having a hard time with its joints?). At the bottom, there's a '목록보기' (List View) button and a footer with the 'Everymal FOR YOUR LITTLE FRIEND' logo, address (서울특별시 종로구 종로 12길 38번 11층), contact (고객센터 T. 1111-2222 (10:00 - 19:00 / 평일시간 12:30 - 13:30) E. everymal@gmail.com), and links for 회원가입 (Join), 이용약관 (Terms of Service), and 개인정보 처리방침 (Privacy Policy).

Project: Everymal

My Page: General Member VI

The screenshots show the 'Everymal' mobile application interface for a general member.

Top Screenshot: Pet Taxi Reservation Management

- Recent Booking Information:**

예약번호	출발지	도착지	택시 번호	예약 날짜
txr0001	경남구성	교대역 시거리	37번1234	2023-04-04
txr0002	서울시청	경기도 파주시 문산	37번1235	2023-04-05
txr0003	종로2가	이천 아카데미	37번1236	2023-04-06
txr0004	을지로 임구역	광화문 경복궁	37번1237	2023-04-07
txr0005	성수역	독성유원지	37번1238	2023-04-08
- Past Usage History:** A table showing a list of previous taxi bookings.

Bottom Screenshot: Pet Taxi Reservation Form

A reservation form for a pet taxi. The form includes fields for:

- Pet Name: 김정수
- Taxi Number: 37번1234
- Reservation Date: 2023. 04. 04.
- Reservation Time: 오후 12:00
- Start Location: 경남구성
- End Location: 교대역 시거리
- Notes: 운전경력 10년 무사고! 반려동물 케어 경력 20년!

Pet Taxi Reservation Management

A page where you can manage all booked taxi information.

- Recent Booking Information: View pet taxi reservation information booked after the current time.
- Past Usage History: View all pet taxi reservation information booked up to the current time.
- Clicking on a taxi reservation navigates to the reservation detail page.

Project: Everymal

My Page: Veterinarian Member

The image shows four screenshots of the Everymal application interface. The top two screenshots are from a mobile device, and the bottom two are from a desktop browser.

- Login Page:** Shows a password input field with a lock icon and placeholder text: "회원님의 회원정보를 안전하게 보호하기 위해 비밀번호를 한번 더 확인해주세요. 비밀번호가 타인에게 노출되지 않도록 주의해 주세요." Below the input field are "취소" and "확인" buttons.
- Profile Management Page (Mobile):** Shows a profile picture placeholder with "회원님 환영합니다!". Below it is a table with member information:

이름	김경민
아이디	kkm1234
비밀번호	...
비밀번호 확인	...
연락처	010 - 90318342
사업자 등록번호	3857261540
이메일	kkm1234@gmail.com
프로필 이미지	[Profile Picture Placeholder]

Buttons include "프로필 선택" and "수정하기".
- Profile Management Page (Desktop):** Shows a similar profile management interface with a "내 정보 관리" header. It includes a "비밀번호 확인" step with a note: "이 입력란을 작성하세요." and a yellow warning icon.
- Home Page:** Shows the Everymal logo, address (서울특별시 강남구 흥국 12층 33동 123호), phone number (010-1234-5678), and email (E. everymal@gmail.com). Navigation links include "1:1문의는 상당", "24시간 응급실", "찾아서", "병원 찾기", and "로그인".

Password Validation Check

Information can be modified after confirming the member's password.

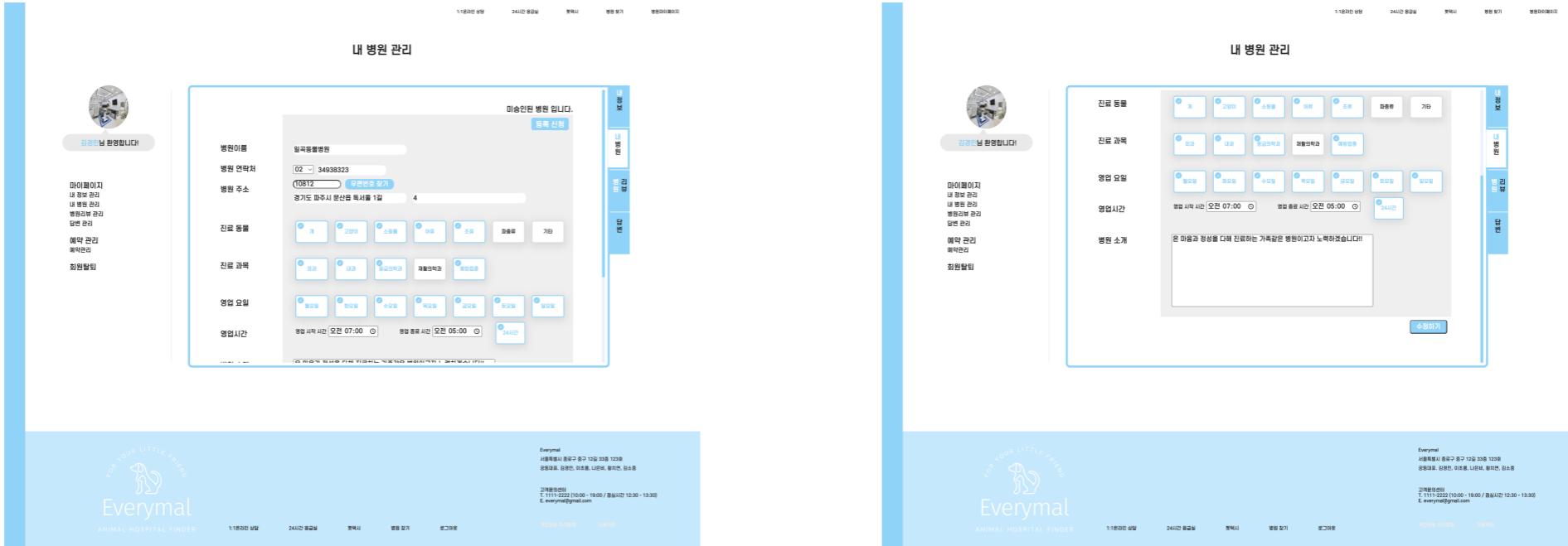
Member Information Update

Name/password/contact information/business registration number/email/profile picture modification page.

Information can be modified after password validity check.

Project: Everymal

My Page: Veterinarian Member II



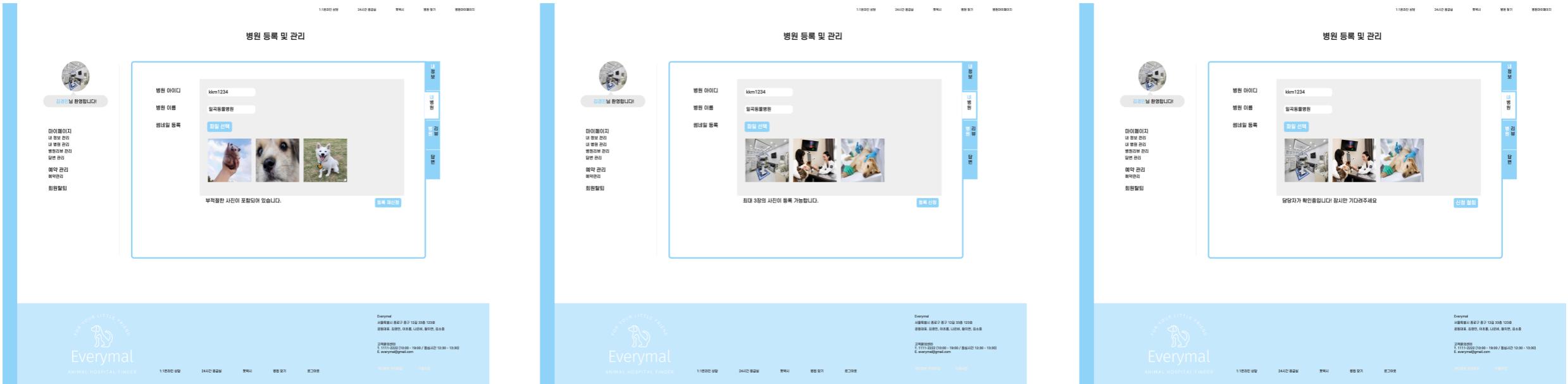
Hospital Information Update I

Hospital name/contact information/address/treated animals/medical specialties/business days/business hours/hospital introduction modification page.

- The modified information on the page will be displayed on the hospital detail page.

Project: Everymal

My Page: Veterinarian Member II



Hospital Information Update II

When the registration request button is clicked, it navigates to the thumbnail modification page.

- If it's the first time requesting registration, "This hospital is not approved" will be displayed, then it moves to the registration request page.
- If waiting for approval after registration request, "Under review" will be displayed, then it moves to the confirmation page.
- If registration is rejected, it moves to the reason confirmation page.
- If re-registering, it moves to the modification page.

Project: Everymal

My Page: Veterinarian Member III

The image shows two screenshots of the Everymal platform. The left screenshot displays the 'Hospital Review Management' page, which lists reviews for a specific hospital. The right screenshot shows a detailed view of a single review.

Hospital Review Management (Left Screenshot):

- Score: 3.70/5
- Review Count: 10
- Reviews:

 - rv0012: 2 stars, "한달 뒤에 이사를 가게 돼서 아쉬워요.", Date: 2023-05-14
 - rv0011: 1 star, "제가 이 병원 단골이라 잘 아는데 환장 봐주고 나서.", Date: 2023-05-13
 - rv0010: 3 stars, "처음 보는 선생님이 계시던데... 좋...", Date: 2023-05-12
 - rv0009: 3 stars, "이 병원 이용했었는데~ 잘 둑라고 해야할지...", Date: 2023-05-11
 - rv0008: 5 stars, "보호자와 반려동물 모두 만족하는 최상의 서비스를 제공해주세요...", Date: 2023-05-10
 - rv0005: 5 stars, "최고예요 항상 이 병원만 애용 해요", Date: 2023-05-09
 - rv0004: 3 stars, "정성스럽게 풀반준 등을영원히! 추천합니다.", Date: 2023-05-08
 - rv0003: 5 stars, "의료진의 열정과 애정으로 내 동물의 건강을 완벽하게 지켜준...", Date: 2023-05-07

Hospital Review Detail View (Right Screenshot):

- Title: "한달 뒤에 이사를 가게 돼서 아쉬워요."
- Date: 2023-05-14
- Score: 2 stars
- Content: "항상 친절하고 대정하게 반려동물을 켜어 해주시던 김진경 선생님, 언제나 아프지 않게 예방 접종을 해주시던 나상범 선생님, 모두 감사 합니다."

Hospital Review Management

A page to check the score and quantity of hospital reviews.

- Displayed as a paging list, allowing you to request deletion of selected reviews.

Hospital Review Detail View

Clicking on a review takes you to the detail view page where you can check the review title, date of writing, review score, and detailed content.

Project: Everymal

My Page: Veterinarian Member

The screenshot displays two main sections of the platform:

Left Section (1:1 Online Consultation Response Management):

- Header: 1:1온라인 상담, 24시간 응급실, 쪽지시, 병원 찾기, 로그아웃.
- Section Title: 답변 관리.
- Table: A list of responses from pet owners. The table includes columns: 선택 (checkbox), 답변 등록 번호, 답변 제목, 답변등록일, and a column for deleting responses (Delete).

선택	답변 등록 번호	답변 제목	답변등록일	삭제
<input type="checkbox"/>	20	강아지 잘 문제로 걱정이시군요	2023-03-20	
<input type="checkbox"/>	19	강아지들도 성격이 각자 재각각입니다.	2023-03-19	
<input type="checkbox"/>	18	고양이 손톱 관리	2023-03-18	
<input type="checkbox"/>	17	정상입니다.	2023-03-17	
<input type="checkbox"/>	16	미세먼지 심한 날 산책을 자제하는 게 좋습니다.	2023-03-16	
<input type="checkbox"/>	15	귀여운 고양이네요	2023-03-15	
<input type="checkbox"/>	14	사람 음식은 안 됩니다.	2023-03-14	
<input type="checkbox"/>	12	산왕보다 귀하다는 풀냥이~	2023-03-12	

Right Section (Response Detail Page):

 - Header: 1:1온라인 상담, 24시간 응급실, 쪽지시, 병원 찾기, 로그아웃.
 - Section Title: 질문으로.
 - List Item: 질문 제목: 강아지가 잠이 너무 많아요.ㅜㅜ, 작성자: teewong, 작성일: 2023-02-17.
 - Text: 건강에 문제가 없고 밥도 잘 먹는데 하루종일 자요. 산책도 그렇게 좋아하지 않고 자고만 싶어요. 건강에 문제가 있는데 계속 이 상태를 유지해도 될까요?
 - Buttons: 수정하기, 질문삭제.
 - List Item: 답변 제목: 강아지 잘 문제로 걱정이시군요, 작성자: 일국동물병원, 작성일: 2023-03-20.
 - Text: 건강에 문제가 없다면 큰 걱정은 안 하셔도 괜찮습니다. 강아지의 평균 수면시간은 성인견을 기준으로 약 16시간 정도입니다.
 - Buttons: 수정하기, 답변삭제.

Everymal
서울특별시 종로구 종로 17길 12층 1238
문점대표: 김현민, 이소정, 나은원, 황지연, 김소정
고객센터
T. 1111-2222 (10:00 - 19:00 / 톨신시간 12:30 - 13:30)
E. everymal@gmail.com

Everymal
서울특별시 종로구 종로 17길 12층 1238
문점대표: 김현민, 이소정, 나은원, 황지연, 김소정
고객센터
T. 1111-2222 (10:00 - 19:00 / 톨신시간 12:30 - 13:30)
E. everymal@gmail.com

1:1 Online Consultation Response Management

A page where you can manage the responses to online consultation inquiries that you have written.

- Allows selection of responses to delete, and upon clicking the delete button, the response deletion function is executed.
- Clicking on a response allows you to move to the response detail page, where you can make modifications.

Project: Everymal

My Page: Administrator I

This screenshot shows a list of general members (일반회원) on the 'Administrator I' page. The table includes columns for 회원코드 (Member Code), 회원아이디 (Member ID), 회원종류 (Member Type), 가입일 (Registration Date), and 탈퇴여부 (Withdrawal Status). A dropdown menu labeled '회원 관리' is visible on the right side of the table.

회원코드	회원아이디	회원종류	가입일	탈퇴여부
use0001	taewong	일반회원	2023-04-29	N
use0002	baekho	일반회원	2023-04-30	N
use0003	daeman	일반회원	2023-05-01	N
use0004	taeseop	일반회원	2023-05-02	N
use0005	chisu	일반회원	2023-05-03	N
use0006	kkm	일반회원	2023-05-04	N

1 2 3 4 ... 19 다음 >

Everymal
ANIMAL HOSPITAL FINDER
서울특별시 강남구 흥국 12길 308 123호
영업执照: 00000, 의약执照: 00000, 대행执照: 00000, 환경执照: 00000
고객센터
E. everyma@gmail.com

1:18:00 09 24시간 응급실 혼자서 병원 찾기 프로그램 회원 가입 회원 로그인

This screenshot shows a list of veterinarian members (병원회원) on the 'Administrator I' page. The table structure is identical to the general members list, showing columns for 회원코드, 회원아이디, 회원종류, 가입일, and 탈퇴여부. A dropdown menu labeled '회원 관리' is visible on the right side.

회원코드	회원아이디	회원종류	가입일	탈퇴여부
hos0104	hoy5702	병원회원	2023-06-05	N
hos0105	hoy5703	병원회원	2023-06-05	N
hos0106	hoy5704	병원회원	2023-06-05	N
hos0107	hoy5705	병원회원	2023-06-05	N
hos0108	hoy5706	병원회원	2023-06-05	N
hos0109	hoy5707	병원회원	2023-06-05	N

1 2 3 4 ... 19 다음 >

Everymal
ANIMAL HOSPITAL FINDER
서울특별시 강남구 흥국 12길 308 123호
영업执照: 00000, 의약执照: 00000, 대행执照: 00000, 환경执照: 00000
고객센터
E. everyma@gmail.com

1:18:00 09 24시간 응급실 혼자서 병원 찾기 프로그램 회원 가입 회원 로그인

This screenshot shows a list of general members (일반회원) on the 'Administrator I' page. The table structure is identical to the previous pages, showing columns for 회원코드, 회원아이디, 회원종류, 가입일, and 탈퇴여부. A dropdown menu labeled '회원 관리' is visible on the right side.

회원코드	회원아이디	회원종류	가입일	탈퇴여부
use0001	taewong	일반회원	2023-04-29	N
use0002	baekho	일반회원	2023-04-30	N
use0003	daeman	일반회원	2023-05-01	N
use0004	taeseop	일반회원	2023-05-02	N
use0005	chisu	일반회원	2023-05-03	N
use0006	kkm	일반회원	2023-05-04	N

1 2 3 4 ... 20 다음 >

Everymal
ANIMAL HOSPITAL FINDER
서울특별시 강남구 흥국 12길 308 123호
영업执照: 00000, 의약执照: 00000, 대행执照: 00000, 환경执照: 00000
고객센터
E. everyma@gmail.com

1:18:00 09 24시간 응급실 혼자서 병원 찾기 프로그램 회원 가입 회원 로그인

This screenshot shows a list of withdrawn members (탈퇴회원) on the 'Administrator I' page. The table includes columns for 회원코드 (Member Code), 회원아이디 (Member ID), 회원종류 (Member Type), 가입일 (Registration Date), and 탈퇴여부 (Withdrawal Status). A dropdown menu labeled '회원 관리' is visible on the right side. A message at the top of the table states '등록된 회원이 존재하지 않습니다.' (No registered members found).

회원코드	회원아이디	회원종류	가입일	탈퇴여부
등록된 회원이 존재하지 않습니다.				

Everymal
ANIMAL HOSPITAL FINDER
서울특별시 강남구 흥국 12길 308 123호
영업执照: 00000, 의약执照: 00000, 대행执照: 00000, 환경执照: 00000
고객센터
E. everyma@gmail.com

1:18:00 09 24시간 응급실 혼자서 병원 찾기 프로그램 회원 가입 회원 로그인

Member Management

A page where you can view lists of members categorized as All Members, Veterinarian Members, General Members, Withdrawn Members, and Active Members.

You can check the member code, member ID, member type, registration date, and withdrawal status.

The functionality to display the lists is implemented using Ajax.

Project: Everymal

My Page: Administrator II

The screenshot shows a table titled "병원 등록 관리" (Hospital Registration Management) with columns: 신청 번호 (Request Number), 신청 명 (Hospital Name), 신청일 (Request Date), and 상태 (Status). The data includes:

신청 번호	병원명	신청일	상태
rm0001	일국동물병원	2023-06-06	완료
rm0002	강릉은 풍물병원	2023-06-06	거절
rm0003	빛고을 풍물의료센터	2023-06-06	미승인
rm0004	24시 언제나 풍물병원	2023-06-06	미승인
rm0005	이은동물병원	2023-06-06	미승인
rm0006	소암동물병원	2023-06-06	미승인
rm0007	광주동물보호소 부속동물병원	2023-06-06	미승인
rm0008	경북대학교 수의 кли닉 부속동물병원	2023-06-06	미승인

Buttons at the bottom include "등록" (Register) and "등록 취소" (Cancel Registration).

Hospital Thumbnail Registration Management

A page for managing the list of hospital thumbnail registration requests.

Displays request number, hospital name, request date, and registration status. Clicking on an entry navigates to the detailed view page.

- Unapproved: If unapproved, the administrator can approve or reject the registration.
- Approved: Upon approval, the hospital will be displayed in the hospital list search feature.
- Rejected: If rejected, the reason for rejection will be provided.

The screenshots show the "병원 등록 신청서" (Hospital Registration Application) page with three stages:

- 승인 진행 중입니다.** (Approval in progress): Shows three preview images of the hospital and the text "승인 진행 중입니다."
- 승인 완료 된 것입니다.** (Approval completed): Shows three preview images and the text "승인 완료 된 것입니다."
- 반려 된 건입니다.** (Rejected): Shows three preview images and the text "반려 된 건입니다." followed by the reason "반려 사유 : 부적절한 사진이 포함되어있습니다."

The screenshots show the "Everymal ANIMAL HOSPITAL FINDER" homepage with the following details:

- Header: Everymal ANIMAL HOSPITAL FINDER
- Links: 1:1문의, 24시간 응급실, 청탁처, 환경 찾기, 보그문
- Footer: Everymal ANIMAL HOSPITAL FINDER, 주소: 대전광역시 동구 대전로 123, 운영시간: 평일 09:00-18:00 / 토요일 09:00-17:00, 이메일: everymal@gmail.com

Project: Everymal

My Page: Administrator III

The screenshot displays two main panels of the Everymal Admin III application. The left panel, titled '관리자 페이지' (Administrator Page), shows a table of reviews for deletion. The table includes columns for '선택' (Select), '리뷰등록번호' (Review Registration Number), '병원코드' (Hospital Code), '병원명' (Hospital Name), and '등록일' (Registration Date). Three rows are listed:

선택	리뷰등록번호	병원코드	병원명	등록일
<input type="checkbox"/>	rv0012	hos0001	일곡동동물병원	2023-05-14
<input type="checkbox"/>	rv0011	hos0001	일곡동동물병원	2023-05-13
<input type="checkbox"/>	rv0001	hos0001	일곡동동물병원	2023-05-05

The right panel, titled '리뷰 보기' (Review View), shows a detailed view of a specific review. It includes fields for '제목' (Title), '날짜' (Date), '별점' (Rating), and '내용' (Content). The review content is: "항상 친절하고 다행하게 반려동물들을 케어 해주시던 김진경 선생님, 언제나 아프지 않게 예방 접종을 해주시던 나상범 선생님, 모두 감사 합니다."

The bottom of the interface features a footer with the Everymal logo, contact information, and various navigation links.

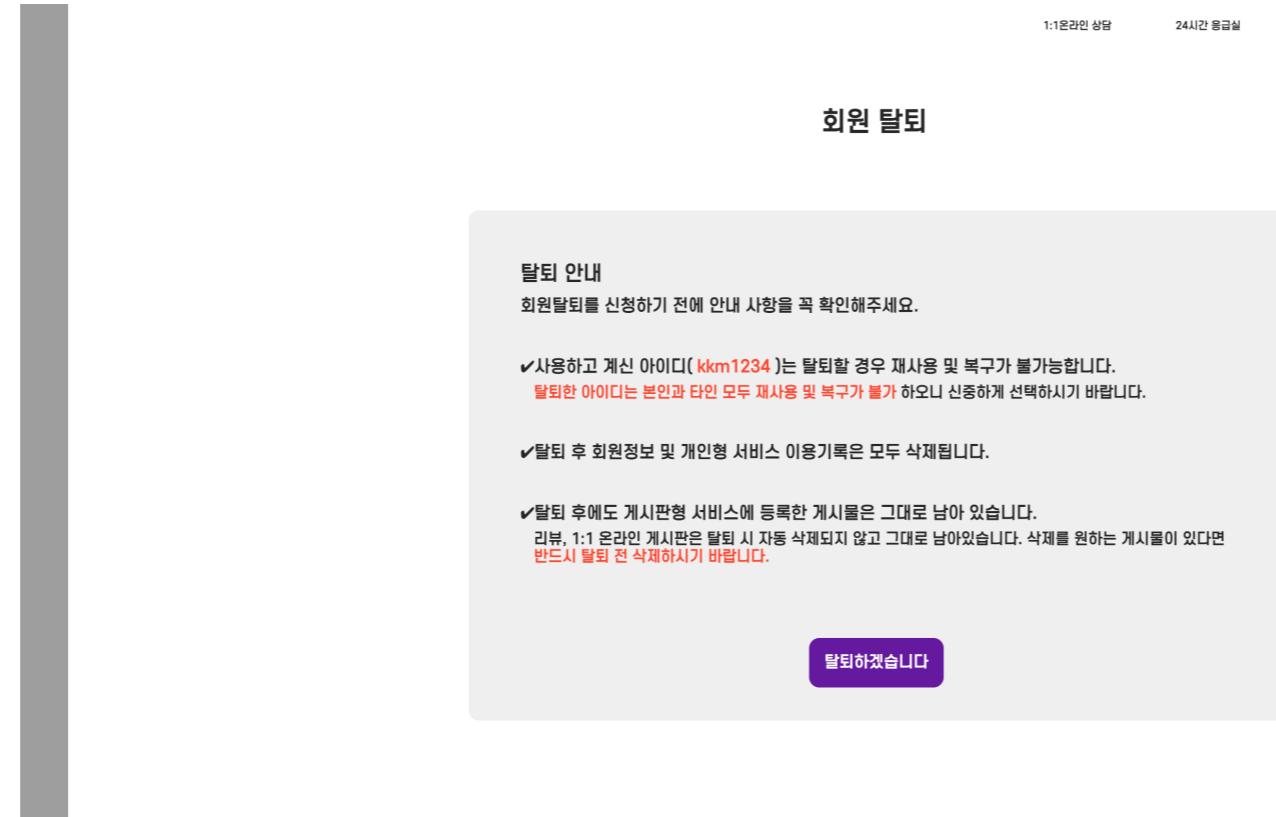
Review Management

A page for managing reviews that veterinarian members have requested to be deleted.

- Clicking on the review that has been requested for deletion allows you to view the review details. After confirmation, clicking on the delete button will complete the deletion.

Project: Everymal

Member Withdrawal



Member Withdrawal

Clicking on the withdrawal button on the My Page will move you to the member withdrawal page.

When applying for withdrawal, guidance will be provided, and then you can proceed to the withdrawal application page.

Project: Everymal

Hospital Search Page

The screenshot shows the 'Everymal' Hospital Search Page. At the top, there are three filter sections: '동물' (Animal) with options for 개 (Dog), 고양이 (Cat), 소동물 (Small animals), 어류 (Fish), and 조류 (Others); '진료과목' (Medical Specialties) with options for 외과 (Surgery), 내과 (Internal Medicine), 응급의학과 (Emergency Medicine), 재활의학과 (Rehabilitation Medicine), and 예방접종 (Vaccination); and '진료 시간' (Consultation Hours) with options for 24시간 (24 hours), 월 (Monday), 화 (Tuesday), 수 (Wednesday), and 목 (Thursday). Below these filters is a search bar labeled '병원명 검색' (Search by hospital name) and a '필터 적용' (Apply filter) button. The main content area displays a grid of 12 hospital cards, each with a thumbnail, name, address, phone number, and a '자세히 보기' (View details) button. The hospitals listed are: 일광동물병원 (Ilgwang Animal Hospital), 침종은동물병원 (Chimjeongeun Animal Hospital), 김애강동물병원 (Kim Ae-gang Animal Hospital), 박고을동물의료센터 (Parkgo-eul Animal Medical Center), 이드동물병원 (Id Animal Hospital), 소담동물병원 (Sodam Animal Hospital), 경북대학교 수의과 대학 부속동물병원 (Kyungpook National University Veterinary Hospital), 다온동물병원 (Dacon Animal Hospital), 신한동물병원 (Shinhan Animal Hospital), and 동명동물병원 (Doming Animal Hospital). At the bottom, there is a footer with the Everymal logo, contact information, and links for '개인정보 처리방침' (Privacy Policy) and '이용약관' (Terms of Service).

Searching for a list of hospitals matching the filters:

- Sorting options: General sorting / Review sorting
- Animal types: Dog / Cat / Small animals / Fish / Birds / Reptiles / Others
- Medical specialties: Surgery / Internal Medicine / Emergency Medicine / Rehabilitation Medicine / Vaccination
- Consultation hours: 24 hours / Mon / Tue / Wed / Thu / Fri / Sat / Sun
- Search by hospital name.

Project: Everymal

Hospital Detail Page

The screenshot displays the 'Hospital Detail Page' for '일곡동물병원' (Ilgok Animal Hospital). The page includes:

- Header:** Includes links for '1:1문의 신청', '24시간 응급실', '전화번호', '병원 찾기', and '로그인 / 회원가입'.
- Profile Section:** Features a large image of a medical room, consultation hours (07:00-04:00), business days (월요일~일요일), a quote about compassionate treatment, medical specialties (외과 / 내과 / 응급의학과 / 허방합증), treated animals (개 / 고양이 / 소豚 / 아우), and a '예약하기' (Book Now) button.
- Review Section:** Shows a rating of 3.70/5 with three reviews:
 - Excellent service and friendly staff.
 - Good services and friendly staff.
 - Good services and friendly staff.A '리뷰 쓰기' (Write Review) button is present.
- Map Section:** A map showing the hospital's location in Paju, Gyeonggi-do.
- Footer:** Includes the Everymal logo ('FOR YOUR LITTLE FRIEND Everymal ANIMAL HOSPITAL FINDER'), address ('서울특별시 강남구 풍무로 12길 31호 123호'), contact info ('T. 1111-2222 (10:00 - 19:00 / 월요일 12:30 - 13:30)', 'E. everymal@gmail.com'), and links for '개인정보 처리방침' and '이용약관'.

Hospital Detail Page

Displays information including consultation hours, business days, hospital introduction, medical specialties, treated animals, map, and address. Clicking on it connects to the reservation page.

When the "Write Review" button is clicked:

Reviews can only be written by members who have previously booked an appointment at the hospital.

Project: Everymal

Hospital Reservation Page

The screenshot shows the '예약하기' (Reservation) page of the Everymal Animal Hospital Finder website. The page includes fields for Name, Contact, Pet Information (Species, Age, Gender), and Booking Details (Date, Time). A prominent purple button labeled '예약하기' (Make Reservation) is at the bottom right. Below the main form is a purple footer bar with the Everymal logo and contact information.

The screenshot shows a modal dialog titled '나의 반려동물 정보' (My Pet Information) overlaid on the reservation form. It displays a table with pet details: Species (Chihuahua), Age (30), Gender (Male), and Name (Kim Hyun-sik). There is also a '반려동물을 선택' (Select Pet) button.

When the "Make Reservation" button is clicked:

- Enter Name/Contact Information/Pet Name/Animal Species/Age/Pet Gender/Pet Registration Number/Pet Blood Type/Pet Weight/Preferred Booking Date/Preferred Booking Time/Medical Specialty/Additional Requirements.
- If existing member's pet information is registered, clicking the "Load My Pet Information" button will retrieve and allow entry of the saved pet information.

Project: Everymal

1:1 Online Consultation Page |

The screenshot shows a list of posts from members. Each post includes a profile picture, the member's name, and a brief message. The posts are as follows:

- 강아지가 잠이 너무 많아요.ㅜㅜ
- 강아지 잘 문제로 걱정이시군요
- 강아지도 거짓말을 하나요?
- 강아지들도 성격이 각자 채각각입니다.
- 고양이 손톱 잘 자르는 방법
- 고양이 손톱 관리
- 개가 스트레스 발산하는 방식이 뭔이에요
- 정상입니다.
- 미세먼지가 심한 날 산책시키면 안 되겠죠?
- 미세먼지 심한 날 산책을 자제하는 게 좋습니다.

At the bottom, there are navigation buttons: 1, 2, 3, 4, and a '글쓰기' (Write) button.

The screenshot shows a detailed view of a post. The post title is "강아지가 잠이 너무 많아요.ㅜㅜ". It includes the member's name (taewong), the date (2023-02-17), and a reply from a staff member (Everymal). The reply reads: "건강엔 문제가 없고 밥도 잘 먹는데 하루종일 자요. 산책도 그렇게 좋아하지 않고 자고만 싶어요. 건강엔 문제가 없는데 계속 이 상태를 유지해도 될까요?". At the bottom, there are buttons for '수정하기' (Edit) and '답변삭제' (Delete).

List Page

- A list where you can see question titles and answer titles.
- It is shown along with the profile page of the member who wrote it. Clicking on the writing button moves to the question writing page, and clicking on the post moves to the post detail page.

The screenshot shows a list of posts from members. Each post includes a profile picture, the member's name, and a brief message. The posts are as follows:

- Everymal
- 고양이에게 물어보세요

Post Detail Page

- A page where you can view question posts and answer posts in detail.
- Editing and deletion are only possible if the member who wrote them is the same.

Project: Everymal

1:1 Online Consultation Page II

1:1온라인 상담 24시간 응급실 봇해시 방문 찾기 흰돌이아름다워

1:1 온라인 상담

목록으로

제목을 입력해주세요.

종료 타입: **문**

내용을 입력해주세요.

등록하기



Everymal
ANIMAL HOSPITAL FINDER

고객센터번호
T. 010-1222 (10:00 ~ 19:00) / 접수시간 12:30 ~ 13:30
E. everymal@gmail.com

개인정보 처리방침 이용약관

1:1문의는 상담
24시간 응급실
찾아서
병원 찾기
회원가입

보내드로

제목을 입력해주세요.

내용을 입력해주세요.

등록하기

Everymel
서울특별시 종로구 종로 12길 53 123
운영대표: 김경민, 이조룡, 나란비, 황치연, 김소옹
고객문의번호
1588-2222 10:00 ~ 19:00 / 휴식시간 12:30 ~ 13:30
E-mail: everymel@gmail.com

개인정보 처리방침
이용약관
FOR YOUR LITTLE FRIEND
Everymel
ANIMAL HOSPITAL FINDER

Question Writing Page

- Writing a question is available only when logged in.
 - Selectable animal species for asking questions.

Answer Writing Page

- Answer writing is available only for veterinarian members.

Project: Everymal

Emergency Room - 24 Hours Page

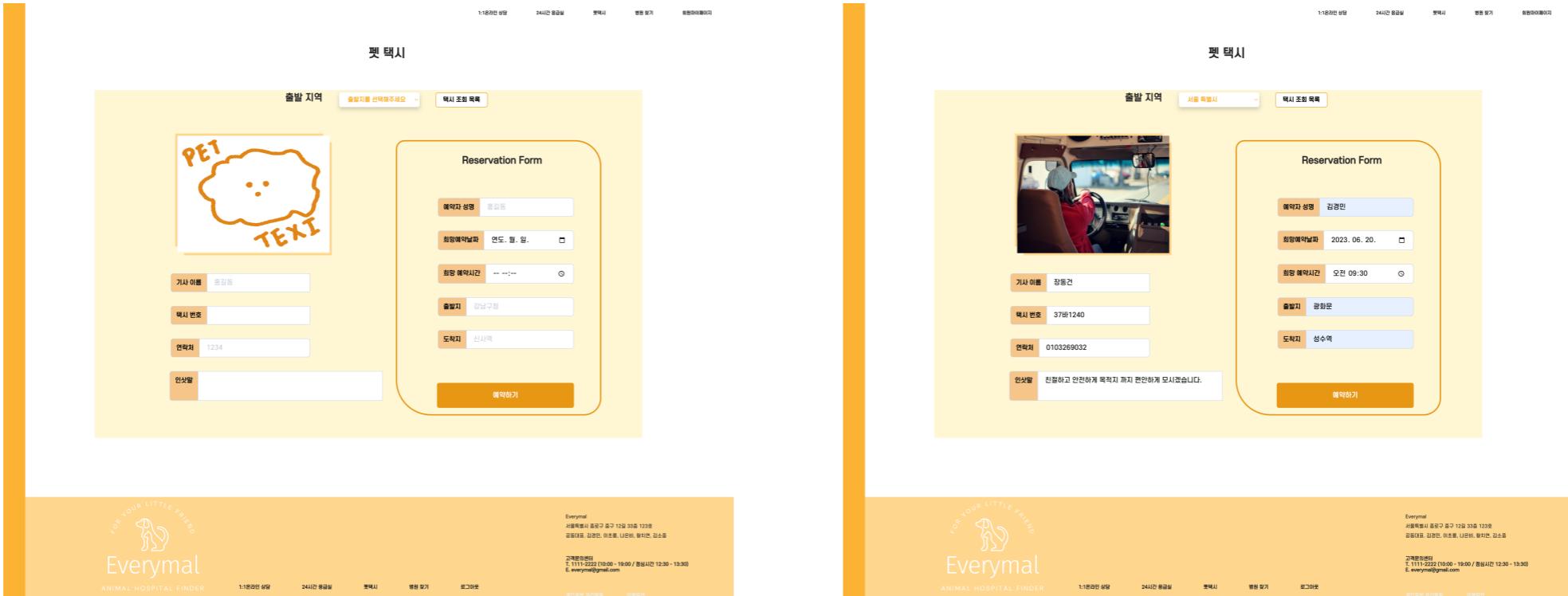


24-Hour Emergency Room and Map Search Page

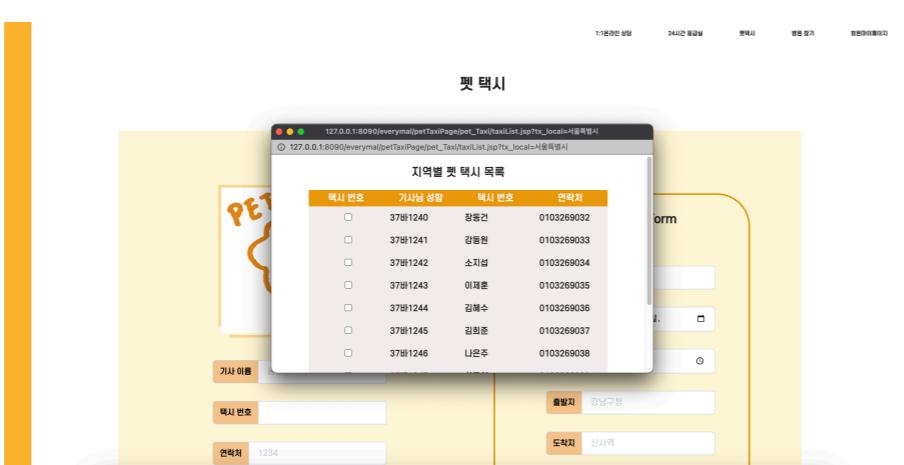
- By default, search results for 24-hour animal hospitals are implemented, and other animal hospitals can also be searched.
- Utilizes the Kakao API.

Project: Everymal

Pet Taxi Page



The screenshot shows the Pet Taxi reservation page. At the top, there are navigation links: '1:1문의란' (FAQ), '24시간 응급실' (24-hour emergency room), '펫 택시' (Pet Taxi), '병원 찾기' (Find a hospital), and '원천화이트화이트' (White Label). Below this is a search bar labeled '출발 지역' (Departure location) with a dropdown menu showing '서울특별시 강남구 풍미동'. To the right is a button '택시 조회 목록' (Taxi search list). The main form is titled 'Reservation Form' and contains fields for '예약자 성명' (Booker's name), '희망예약날짜' (Desired reservation date), '희망 예약시간' (Desired reservation time), '출발지' (Departure location), and '도착지' (Arrival location). There is also a '기사 이름' (Driver's name) field set to '강동건', a '택시 번호' (Taxi number) field set to '3781240', and a '연락처' (Contact number) field set to '0103269032'. A note at the bottom says '인삿말' (Greeting) and '친절하고 안전하게 목적지까지 편안하게 모시겠습니다.' (We will safely and kindly transport you to your destination). A large orange '예약하기' (Reserve) button is at the bottom right.



This screenshot shows a modal window titled '지역별 펫 택시 목록' (List of regional pet taxis) over the reservation form. It lists drivers with their names and contact numbers:

택시 번호	기사님 성함	택시 번호	연락처
3781240	강동건	0103269032	
3781241	강동원	0103269033	
3781242	소지섭	0103269034	
3781243	이재훈	0103269035	
3781244	김혜수	0103269036	
3781245	김희준	0103269037	
3781246	나은주	0103269038	

Pet Taxi Search and Reservation Page

- Implement a popup window where you can view a list of drivers corresponding to the selected departure location.
- After selecting a driver, enter the booker's name, desired reservation date, preferred reservation time, departure location, and destination. Upon completing the reservation, redirect to the My Page Pet Taxi list.

Project: Everymal

Major Code I: User Registration and Validation

JavaScript in join_user.jsp

```
function submit() {
    //회원정보 저장
    data = $("#joinForm").serialize();
    $.ajax({
        type: "post",
        async: false,
        dataType: "text",
        url: "${contextPath}/member/addMember.do",
        data: data,
        success: function (result) {
            var user_code = result;
            //동물정보 저장
            $("form[name=joinAnimalForm]").each(function (index, item) {
                data = $("#joinAnimalForm_" + index).serialize();
                data = data + "&user_code=" + user_code;
                $.ajax({
                    type: "post",
                    async: false,
                    dataType: "text",
                    url: "${contextPath}/animal/addAnimal.do",
                    data: data
                });
            });
        }
    });
    next();
}

function next() {
    alert('회원가입완료.');
    location.href = "${contextPath}/main.do";
}
```

submit function: Saves member information and pet information to the server.

- **Member information Ajax:** Processes without using asynchronous communication to /member/addMember.do.
- **Pet information Ajax:** Processes asynchronously to /animal/addAnimal.do.
- Each pet information entry form (joinAnimalForm) generates an Ajax request to send pet information to the server, and each pet information request adds a user_code parameter to link the member and pet.
- Once all Ajax requests are completed, it displays a message confirming successful registration using the next function and redirects to the main page.

Project: Everymal

Major Code I: User Registration and Validation

```
function check() {
    if (a == '') {
        alert('중복체크를 해주세요');
        joinForm.user_id.focus();
        return false;
    }
    if (a == 'not_usable') {
        alert('사용할 수 없는 ID입니다. 다시 중복체크를 해주세요');
        joinForm.user_id.focus();
        return false;
    }
    /* 이름 유효성 검사 */
    if (joinForm.user_name.value.length == 0) {
        alert("이름을 입력해주세요.");
        joinForm.user_name.focus();
        return false;
    }
    /* 비밀번호 및 비밀번호 확인 유효성 검사 */
    if (joinForm.user_pwd.value.length == 0) {
        alert("비밀번호를 입력해주세요.");
        joinForm.user_pwd.focus(); // 포커스를 이동시켜 바로 비밀번호를 입력할 수 있게
        return false;
    }
    var pwdRegExp = /^[a-zA-Z0-9]{4,8}$/; //비밀번호 유효성 검사
    if (!pwdRegExp.test(joinForm.user_pwd.value)) {
        alert("비밀번호는 영문 대소문자와 숫자 4~8자리로 입력해야합니다.");
        joinForm.user_pwd.focus();
        return false;
    }
    if (joinForm.confirmPassword.value.length == 0) {
        alert("비밀번호 확인을 입력해주세요.");
        joinForm.confirmPassword.focus();
        return false;
    }
    if (joinForm.user_pwd.value != joinForm.confirmPassword.value) {
        alert("비밀번호가 일치하지 않습니다.");
        joinForm.confirmPassword.select(); //이걸로 하면 focus 이동하면서 입력한게 블록으로 선택됨
        return false;
    }
}
```

check function: Performs validation of the registration form.

- Validates the following fields: ID (checks for emptiness and if it's not_usable), name, password and password confirmation, phone number, address, and email.

JavaScript in join_user.jsp

```
/* 전화번호 유효성 검사 */
if (joinForm.user_tel1.value == "") {
    alert("번호를 선택해주세요.");
    joinForm.user_tel1.focus();
    return false;
}
if (joinForm.user_tel2.value.length == 0) {
    alert("전화번호를 입력해주세요.");
    joinForm.user_tel2.focus();
    return false;
}
if (joinForm.user_tel3.value.length == 0) {
    alert("전화번호를 입력해주세요.");
    joinForm.user_tel3.focus();
    return false;
}
if (isNaN(joinForm.user_tel2.value)) {
    alert("전화번호는 숫자만 입력가능합니다.");
    joinForm.user_tel2.focus();
    return false;
}
if (isNaN(joinForm.user_tel3.value)) {
    alert("전화번호는 숫자만 입력가능합니다.");
    joinForm.user_tel3.focus();
    return false;
}
/* 주소 유효성 검사 */
if (joinForm.user_zipcode.value.length == 0) {
    alert("주소를 입력해주세요.");
    joinForm.user_zipcode.focus();
    return false;
}
if (joinForm.user_addr1.value.length == 0) {
    alert("주소를 입력해주세요.");
    joinForm.user_addr1.focus();
    return false;
}
if (joinForm.user_addr2.value.length == 0) {
    alert("주소를 입력해주세요.");
    joinForm.user_addr2.focus();
    return false;
}
/* 이메일 유효성 검사 */
if (joinForm.user_email.value.length == 0) {
    alert("이메일을 입력해주세요.");
    joinForm.user_email.focus();
    return false;
}
```

Project: Everymal

Major Code I: User Registration and Validation

JavaScript in join_user.jsp

```
/* 동물 유효성 */
try {
    $("form[name=joinAnimalForm]").each(function (index, item) {
        if ($("#joinAnimalForm_" + index).find('#pet_name').val() == '') {
            alert("반려동물의 이름을 입력해주세요.");
            $("#joinAnimalForm_" + index).find('#pet_name').focus();
            throw 'finish';
        }
        if ($("#joinAnimalForm_" + index).find('#pet_age').val() == '') {
            alert("반려동물의 나이를 입력해주세요.");
            $("#joinAnimalForm_" + index).find('#pet_age').focus();
            throw 'finish';
        }
        flag = false;
        for (i = 0; i < ($("#joinAnimalForm_" + index).find('input[name="pet_sex"]').length; i++) {
            if ($("#joinAnimalForm_" + index).find('input[name="pet_sex"]')[i].checked) {
                flag = true;
                break; // 라디오박스에서는 1개만 선택될 수 있으므로 사용가능
            }
        }
        if (flag == false) {
            $("#joinAnimalForm_" + index).find('#pet_sex_f').focus();
            alert("반려동물을 성별을 체크해주세요.");
            throw 'finish';
        }
        if ($("#joinAnimalForm_" + index).find('#pet_types').val() == '') {
            alert("반려동물 종을 선택해주세요.");
            $("#joinAnimalForm_" + index).find('#pet_types').focus();
            throw 'finish';
        }
        if ($("#joinAnimalForm_" + index).find('#pet_number').val() == '') {
            alert("반려동물 등록번호를 입력해주세요.");
            $("#joinAnimalForm_" + index).find('#pet_number').focus();
            throw 'finish';
        }
        if (isNaN($("#joinAnimalForm_" + index).find('#pet_number').val())) {
            alert("반려동물 등록번호는 숫자만 입력 가능합니다.");
            $("#joinAnimalForm_" + index).find('#pet_number').select();
            throw 'finish';
        }
        if ($("#joinAnimalForm_" + index).find('#b_type').val() == '') {
            alert("반려동물의 혈액형을 선택해주세요.");
            $("#joinAnimalForm_" + index).find('#b_type').focus();
            throw 'finish';
        }
        if ($("#joinAnimalForm_" + index).find('#pet_weight').val() == '') {
            alert("반려동물의 몸무게를 입력해주세요.");
            $("#joinAnimalForm_" + index).find('#pet_weight').focus();
            throw 'finish';
        }
        if (isNaN($("#joinAnimalForm_" + index).find('#pet_weight').val())) {
            alert("반려동물 몸무게는 숫자만 입력 가능합니다.");
            $("#joinAnimalForm_" + index).find('#pet_weight').select();
            throw 'finish';
        }
    });
} catch (Exception) {
    if (Exception !== 'finish') throw Exception;
    else return false;
}
submit();
}
```

Validation of pet information:

Since pet information can be dynamically added through Ajax, validation is performed by iterating through each form using the 'each' function. Each form is distinguished by its index.

- Validate the following fields for each pet: name, age, gender, species, registration number, blood type, and weight.

Project: Everymal

Major Code I: User Registration and Validation

JavaScript in join_user.jsp

```
var a = '';
function fn_idCheck() {
    let _id = $("#id").val();
    if (_id == "") {
        alert("중복체크할 ID를 입력하세요");
        return; //함수를 종료한다
    }
    $.ajax({
        type: "post",
        async: true, //비동기로 수행
        dataType: "text",
        url: "${contextPath}/member/idChk.do",
        data: { id: _id }, //서버에 전송할 자료= > 매개변수이름:값 _id(체크할ID)를 id이름으로 서버에 전송
        success: function (idChkVal, textStatus) { //idChkVal은 서버에서 받은 자료
            if (idChkVal == 'usable') {
                a = idChkVal;
                $("#resultMessage").text("사용할 수 있는 ID입니다.");
            } else {
                a = idChkVal;
                $("#resultMessage").text("사용할 수 없는 ID입니다.");
            }
        },
        error: function () {
            alert("서버와 통신 중 에러 발생");
        }
    });
}
```

The **fn_idCheck** function is used to check for duplicate IDs.

- Initialize variable 'a' with an empty string to store the result of the duplicate check.
- 'id': the ID entered by the user.
- Use Ajax to asynchronously request a duplicate check from the server at /member/idChk.do.
- Check the received 'idChkVal' value from the server. If it is 'usable', print 'The ID is available for use'. Otherwise, print 'The ID is not available for use'.
- Store the 'idChkVal' value in the 'a' variable so that it can be used for ID validation checks when the 'check' function is executed.

MemberControllerImpl.java

```
@RequestMapping(value = "/member/idChk.do", method = RequestMethod.POST)
public void idChk(@ModelAttribute("memberVO") MemberVO memberVO, HttpServletRequest request, HttpServletResponse response) throws Exception {
    request.setCharacterEncoding("utf-8");
    response.setContentType("text/html;charset=utf-8");
    PrintWriter out = response.getWriter();
    boolean result=false;
    String id = memberService.idChk(memberVO);
    result=Boolean.parseBoolean(id);
    if(result==true) {
        out.print("not_usable");
    }else{
        out.print("usable");
    }
}
```

/member/idChk.do

The idChk function in the memberService is used to check for the duplication of an ID and returns the result.

- If the ID is already in use and cannot be used: "not_usable" is returned.
- If the ID is not duplicated and can be used: "usable" is returned.

Project: Everymal

Major Code II: x List

hosfilter.xml

```
<!-- 일반정렬 병원목록 -->
<select id="selectAllHos" resultMap="hosResult" parameterType="filterVO">
    <![CDATA[
        SELECT * FROM(SELECT ROWNUM AS recNum, res.* FROM(SELECT * from hosfiltertbl NATURAL JOIN hosjointbl order by hos_code) res where 1=1
    ]]>
    <if test="hos_dog!=null and !hos_dog.equals('')">
        and hos_dog=#{hos_dog}
    </if>
    <if test="hos_cat!=null and !hos_cat.equals('')">
        and hos_cat=#{hos_cat}
    </if>
    <if test="hos_small!=null and !hos_small.equals('')">
        and hos_small=#{hos_small}
    </if>
    <if test="hos_fish!=null and !hos_fish.equals('')">
        and hos_fish=#{hos_fish}
    </if>
    <if test="hos_bird!=null and !hos_bird.equals('')">
        and hos_bird=#{hos_bird}
    </if>
    <if test="hos_rep!=null and !hos_rep.equals('')">
        and hos_rep=#{hos_rep}
    </if>
    <if test="hos_etc!=null and !hos_etc.equals('')">
        and hos_etc=#{hos_etc}
    </if>
    <if test="hos_gs!=null and !hos_gs.equals('')">
        and hos_gs=#{hos_gs}
    </if>
    <if test="hos_im!=null and !hos_im.equals('')">
        and hos_im=#{hos_im}
    </if>
    <if test="hos_em!=null and !hos_em.equals('')">
        and hos_em=#{hos_em}
    </if>
    <if test="hos_rm!=null and !hos_rm.equals('')">
        and hos_rm=#{hos_rm}
    </if>
    <if test="hos_vac!=null and !hos_vac.equals('')">
        and hos_vac=#{hos_vac}
    </if>
    <if test="hos_24!=null and !hos_24.equals('')">
        and hos_24=#{hos_24}
    </if>
    <if test="hos_mon!=null and !hos_mon.equals('')">
        and hos_365 like '%mon%'
    </if>
    <if test="hos_tue!=null and !hos_tue.equals('')">
        and hos_365 like '%tue%'
    </if>
    <if test="hos_wed!=null and !hos_wed.equals('')">
        and hos_365 like '%wed%'
    </if>
    <if test="hos_thu!=null and !hos_thu.equals('')">
        and hos_365 like '%thu%'
    </if>
    <if test="hos_fri!=null and !hos_fri.equals('')">
        and hos_365 like '%fri%'
    </if>
    <if test="hos_sat!=null and !hos_sat.equals('')">
        and hos_365 like '%sat%'
    </if>
    <if test="hos_sun!=null and !hos_sun.equals('')">
        and hos_365 like '%sun%'
    </if>
    <if test="hos_sch_name!=null and !hos_sch_name.equals('')">
        and hos_name like '%'||#{hos_sch_name}||'%'
    </if>
    ) WHERE recNum BETWEEN (#{pageNum}-1)*12+1 AND #{section}*12
</select>
```

HosFilterServiceImpl.java

```
@Override
public List selectAllHos(Map pagingMap) throws DataAccessException {
    List<FilterVO> hosList=new ArrayList<FilterVO>();
    int section=(int) pagingMap.get("section");
    int pageNum=(int) pagingMap.get("pageNum");
    FilterVO filter=(FilterVO) pagingMap.get("filterVO");
    filter.setPageNum((section-1)*5+pageNum);
    filter.setSection((section-1)*5+pageNum+12);
    hosList = sqlSession.selectList("mapper.hosfilter.selectAllHos", filter);
    return hosList;
}
```

Dynamic SQL in MyBatis: it is used to search hospital listings flexibly based on complex conditions.

- When executing a query, it checks the conditions based on the attributes of the filterVO object passed as a parameter, and only adds the corresponding condition to the query if the condition is true.

hosFilterService:

- Utilizes the selectAllHos function from hosFilterDAO to retrieve hospital listings filtered based on pagingMap from the database and returns them as a List.
- Uses the selectTotHos function from hosFilterDAO to retrieve the total number of hospitals.
- The retrieved hospital listings and the total number of hospitals are stored in the articleMap under the keys hosList and totalHos, respectively,

Project: Everymal

Major Code III: Implementation of List Page using Ajax/Json

memberList.jsp

```
<div class="tabcontents">
  <div id="tabBox" class="tab-content">
    <form id="memberlistForm">
      <select name="orderby" id="orderby">
        <option value="list" selected>전체회원</option>
        <option value="glist">일반회원</option>
        <option value="hlist">병원회원</option>
        <option value="elist">탈퇴회원</option>
        <option value="alist">활동회원</option>
      </select>
      <p class="total_review total_p">회원 수: <span class="gatsoo" id="count">${totalList}</span>명</p>
      <div class="tableBox">
        <table style="height: 100%;">
          <colgroup>
            <col width="20%" />
            <col width="20%" />
            <col width="20%" />
            <col width="20%" />
            <col width="20%" />
          </colgroup>
          <thead>
            <tr class="member-row">
              <th>회원코드</th>
              <th>회원아이디</th>
              <th>회원종류</th>
              <th>가입일</th>
              <th>탈퇴여부</th>
            </tr>
          </thead>
          <tbody id="userListTableBody">
          </tbody>
        </table>
      </div>
    </form>
    <div id="divPaging">
    </div>
  </div>
</div>
```

Based on the value of the options in the Select tag (list/glist/hlist/elist/alist), the following actions are performed:

1. Display the total number of members (\${totalList}).
2. Populate the tbody with content.
3. Display the paging list inside a div with the class name divPaging.

Project: Everymal

Major Code III: Implementation of List Page using Ajax/Json

JavaScript in memberList.jsp

```
// select 요소 값 변경 시 이벤트 처리
$("#orderby").change(function() {
    var contextPath = "${pageContext.request.contextPath}";
    var selectedValue = $(this).val(); // 선택된 값 가져오기
    console.log(selectedValue);
    $.ajax({
        type: "GET",
        url: contextPath + "/administrator/member_List.do",
        dataType: "json",
        data: { orderby: selectedValue },
        success: function(data) {
            updateList(data);
            updatePaging(data);
        },
        error: function(error) {
            alert("데이터를 가져오는데 실패했습니다.");
            console.log("error: " + error);
        }
    });
});
```

Based on the selected value (orderby) from the Select tag, JSON-formatted data is sent to the server using Ajax (/administrator/member_List.do). The controller processes the request and sends back the data, which is then received and utilized by the **updateList** and **updatePaging** functions.

JavaScript in memberList.jsp

```
$("#orderby").val("list").change();

function updateList(data) {
    var userList = data.userList; // 업데이트할 부분에 해당하는 데이터 추출
    var count = data.totalList
    $("#count").text(count);

    var tbody = ""; // 업데이트할 tbody를 담을 변수

    if (userList.length === 0) {
        // userList가 비어있을 경우 메시지를 표시
        tbody = "<tr><td colspan='5' align='center'>등록된 회원이 존재하지 않습니다.</td></tr>";
    } else {
        // userList를 순회하며 테이블의 각 행을 생성하여 tbody에 추가
        for (var i = 0; i < userList.length; i++) {
            var user = userList[i];
            tbody += "<tr>";
            tbody += "<td>" + user.user_code + "</td>";
            tbody += "<td>" + user.user_id + "</td>";
            tbody += "<td>";
            tbody += user.user_code.startsWith('hos') ? "병원회원" : "일반회원";
            tbody += "</td>";
            tbody += "<td>" + user.j_date + "</td>";
            tbody += "<td>";
            tbody += user.user_enabled == 0 ? "N" : "Y";
            tbody += "</td>";
            tbody += "</tr>";
        }
    }

    // 업데이트된 tbody로 기존의 tbody를 교체하여 업데이트
    $("#userListTableBody").html(tbody);
};
```

updateList function:

1. Display the total number of members that match the list.
2. If statement: If there is no data to display, output "No registered members exist."
3. If statement: If there is data to display, use a for loop to populate the tbody with member code, member ID, membership type (veterinarian member or regular member), registration date, and withdrawal status (N or Y).

Project: Everymal

Major Code III: Implementation of List Page using Ajax/Json

JavaScript in memberList.jsp

```
function updatePaging(data) {
    var currentPage = data.currentPage;
    var totalPages = data.totalPages;
    var totalList = data.totalList;

    var pagingInfo = "전체 " + totalList + "개 중 " + currentPage + " 페이지";
    $("#pagingInfo").text(pagingInfo);
    console.log(pagingInfo);

    var pagingHtml = ""; // 페이지 링크를 담을 변수

    // 이전 페이지 링크 생성
    if (currentPage > 1) {
        pagingHtml += "<a href='#' onclick='goToPage(" + (currentPage - 1) + ")'>&lt; 이전</a> ";
    }

    // 페이지 번호 링크 생성
    var startPage = Math.max(1, currentPage - 2); // 현재 페이지를 중심으로 양쪽에 2개 페이지 번호 출력
    var endPage = Math.min(totalPages, startPage + 4); // 시작 페이지로부터 최대 5개의 페이지 번호 출력

    if (startPage > 1) {
        pagingHtml += "<a href='#' onclick='goToPage(1)'>1</a> ";
        if (startPage > 2) {
            pagingHtml += "... "; // 중간 페이지 생략 부분 표시
        }
    }

    for (var i = startPage; i <= endPage; i++) {
        if (i === currentPage) {
            pagingHtml += "<span style='color: #9e9e9e;'>" + i + "</span> ";
        } else {
            pagingHtml += "<a href='#' onclick='goToPage(" + i + ")">" + i + "</a> ";
        }
    }

    if (endPage < totalPages) {
        if (endPage < totalPages - 1) {
            pagingHtml += "... "; // 중간 페이지 생략 부분 표시
        }
        pagingHtml += "<a href='#' onclick='goToPage(" + totalPages + ")">" + totalPages + "</a> "; // 마지막 페이지로 이동하는 링크
    }

    // 다음 페이지 링크 생성
    if (currentPage < totalPages) {
        pagingHtml += "<a href='#' onclick='goToPage(" + (currentPage + 1) + ")">다음 &gt;</a> ";
    }

    // 생성된 페이지 HTML을 페이지 영역에 업데이트
    $("#divPaging").html(pagingHtml);
};
```

The **updatePaging** function receives the current page (currentPage), total pages (totalPage), and total list count (totalList) data.

1. Previous page link: It uses the **goToPage** function to navigate to the current page minus one page.
2. Page number generation:
 - It determines the number of pages to be displayed using the min and max functions (from the Math class).
 - Since the current page keeps changing, it uses a for loop to update the links for the current page.
3. Next page link: It uses the **goToPage** function to navigate to the current page plus one page.

JavaScript in memberList.jsp

```
// 페이지 이동 함수
function goToPage(page) {
    var contextPath = "${pageContext.request.contextPath}";
    var selectedValue = $("#orderby").val();
    console.log(selectedValue);

    $.ajax({
        type: "GET",
        url: contextPath + "/administrator/member_List.do",
        dataType: "json",
        data: { orderby: selectedValue, pageNum: page },
        success: function(data) {
            updateList(data);
            updatePaging(data);
        },
        error: function(error) {
            alert("데이터를 가져오는데 실패했습니다.");
            console.log("error: " + error);
        }
    });
};
```

The function **goToPage** adds functionality for navigating through the paging list.

It sends a request to the server using Ajax in JSON format based on the selected value (orderby) of the Select tag (/administrator/member_List.do). Then, it receives the data output through the controller as **updateList** and **updatePaging**.

Project: Everymal

Major Code III: Implementation of List Page using Ajax/Json

adminControllerImpl.java

```
//일반 jsp보여주는 걸 mapping해줫다. 그레아 밑에 아래스 험수 jsp에서 호출 가능
@RequestMapping(value = "/administrator/memberList.do", method = RequestMethod.GET)
public ModelAndView valid1(@NodeAttribute("administrator") AdminVO adminVO, @RequestParam(value = "action", required = false) String action,
RedirectAttributes rAttr, HttpServletRequest request, HttpServletResponse response) throws Exception {
    String viewName=(String) request.getAttribute("viewName");
    HttpSession session=request.getSession();
    session.setAttribute("action", action);
    ModelAndView mav = new ModelAndView(viewName);
    return mav;
}

//ResponseBody해야하고, pom.xml에 json추가해야한다.
@RequestMapping(value = "/administrator/member_list.do", produces = "application/json; charset=UTF-8")
@ResponseBody
public Map<String, Object> handleSelectedValue1(HttpServletRequest request) throws Exception {
    String orderBy = request.getParameter("orderBy");
    System.out.println(orderBy);
    String _section = request.getParameter("section");
    String _pageNum = request.getParameter("pageNum");
    System.out.println(_section);
    int section;
    if (_section == null || _section.isEmpty()) {
        section = 1; // 기본값 설정
    } else {
        try {
            section = Integer.parseInt(_section);
        } catch (NumberFormatException e) {
            section = 1; // 기본값으로 설정하거나 적절한 오류 처리를 수행하세요.
        }
    }

    int pageNum = Integer.parseInt(_pageNum == null ? "1" : _pageNum);

    Map<String, Object> pagingMap = new HashMap<String, Object>();
    pagingMap.put("section", section);
    pagingMap.put("pageNum", pageNum);
    pagingMap.put("orderBy", orderBy);
    System.out.println("section:::" + section);
    System.out.println("pageNum:::" + pageNum);

    Map<String, Object> result = adminService.memberList(pagingMap);
    List userList = (List) result.get("userList");
    int totalList = (int) result.get("totalList");
    int totalPages = (int) Math.ceil((double) totalList / 6); //6개씩 보여줄거니
    // System.out.println(result.values());
    System.out.println(totalList);

    Map<String, Object> response = new HashMap<String, Object>();
    response.put("userList", userList);
    response.put("totalList", totalList);
    response.put("currentPage", pageNum);
    response.put("totalPages", totalPages);

    return response;
}
```

/administrator/member_List.do

Through JSP, the orderBy, section, and pageNum parameters are received, and the adminService's memberList is executed accordingly. Based on the obtained values, userList, totalList, and totalPages are retrieved.

- The totalPages is calculated using the ceil function (Math class) to display 6 items per page.
- The received pageNum becomes the currentPage each time the action is executed.

adminServiceImpl.java

```
@Override
public Map<String, Object> memberList(Map<String, Object> pagingMap) throws DataAccessException {
    Map<String, Object> resultMap = new HashMap<String, Object>();
    List<AdminVO> userList = null;
    String orderBy = String.valueOf(pagingMap.get("orderBy"));
    int totalList=0;

    if ("list".equals(orderBy)) {
        userList = adminDAO.memberList(pagingMap);
        totalList=adminDAO.allmember();
    } else if ("glist".equals(orderBy)) {
        userList = adminDAO.userMemberList(pagingMap);
        totalList=adminDAO.gmember();
    } else if ("hlist".equals(orderBy)) {
        userList = adminDAO.hosMemberList(pagingMap);
        totalList=adminDAO.hosmember();
    } else if ("elist".equals(orderBy)) {
        userList = adminDAO.enabledMemberList(pagingMap);
        totalList=adminDAO.enabledmember();
    } else if ("alist".equals(orderBy)) {
        userList = adminDAO.abledMemberList(pagingMap);
        totalList=adminDAO.abledmember();
    } else {
        userList=adminDAO.memberList(pagingMap);
        totalList=adminDAO.allmember();
    }

    resultMap.put("userList", userList);
    resultMap.put("totalList", totalList);
    return resultMap;
}
```

The **memberList** executes the **adminDAO**, which can obtain the list of all members, regular members, veterinarian members, withdrawn members, and active members, as well as the total number of members data values based on the value of the orderBy parameter.

The obtained data values are placed in a **resultMap** (HashMap) with **userList** and **totalList** and returned.

→ The returned value is received in the controller.

Project: Everymal

Major Code IV: Multiple Image Upload

hosApplication.jsp

```
<div class="tabcontents">
    <div id="tabBox" class="tab-content">
        <div class="inputbox"></div>
        <form id="imgForm" method="post" enctype="multipart/form-data">
            <div class="agreeform formbox">
                <label for="">병원 아이디</label>
                <input type="text" name="hos_id" value="${hosmypageinfoV0.hos_id}" readonly />
            </div>
            <div class="agreeform formbox">
                <label for="">병원 이름</label>
                <input type="text" name="hos_name" value="${hosmypageinfoV0.hos_name}" readonly />
                <input type="hidden" name="hos_code" value="${hosmypageinfoV0.hos_code}" />
            </div>
            <div class="formbox">
                <p class="proImg infoTitle" for="thumimgup">썸네일 등록</p>
                <div class="upload-box">
                    <label class="file-label" for="img_upload">파일 선택</label>
                    <input class="file" id="img_upload" name="img_upload" type="file" multiple onchange="previewImage(this,'View_area')"/>
                </div>
            </div>
            <div id="imgBox">
                <c:choose>
                    <c:when test="${not empty hosmypageinfoV0.himg1}">
                        
                    </c:when>
                    <c:otherwise>
                        
                    </c:otherwise>
                </c:choose>
                <c:choose>
                    <c:when test="${not empty hosmypageinfoV0.himg2}">
                        
                    </c:when>
                    <c:otherwise>
                        
                    </c:otherwise>
                </c:choose>
                <c:choose>
                    <c:when test="${not empty hosmypageinfoV0.himg3}">
                        
                    </c:when>
                    <c:otherwise>
                        
                    </c:otherwise>
                </c:choose>
            </div>
            <span id='View_area'></span>
            <div id="resultDiv"><p th:text="${log }" style="color: red;"></p>
        </div>
        <c:choose>
            <c:when test="${hosmypagermstatusV0.rm_status == '승인'}">
                <div class="adminNotice noticebox">승인이 완료되었습니다.</div>
                <input type="button" class="bttns" onclick="dataSubmit();" value="썸네일 수정 요청">
                <a href="#" class="bttns">등록취소</a>
            </c:when>
            <c:when test="${hosmypagermstatusV0.rm_status == '미승인'}">
                <div class="adminNotice noticebox">최대 3장의 사진이 등록 가능합니다.</div>
                <input type="button" class="bttns" onclick="dataSubmit();" value="등록 신청">
            </c:when>
            <c:when test="${hosmypagermstatusV0.rm_status == '거절'}">
                <div class="adminNotice noticebox">${hosmypagermsayV0.rm_say }</div>
                <input type="button" class="bttns" onclick="dataSubmit();" value="등록 재신청">
            </c:when>
            <c:otherwise>
                <div class="noticebox">담당자가 확인중입니다! 잠시만 기다려주세요</div>
                <a href="${contextPath}/hos_MypageInfo/hosInformation.do?hos_id=${hosmypageinfoV0.hos_id }" class="bttns">신청 철회</a>
            </c:otherwise>
        </c:choose>
    </form>
</div>
```

- To handle multiple image uploads, specify the input tag with the class name "file" and set it to multiple.
- The previewImage function is used for preview functionality.
- Using JSTL tags (<c:choose>), it displays the thumbnail image or the specified image (noPhoto.png) depending on whether the previous thumbnail is registered or not.
- Using JSTL tags (<c:choose>), it displays the page differently based on the thumbnail registration status (rm.status), such as thumbnail modification request, registration request, registration re-request, or request withdrawal.

Project: Everymal

Major Code IV: Multiple Image Upload

```
//form데이터 전송
function dataSubmit() {
    var token = $("meta[name='_csrf']").attr("content");
    var header = $("meta[name='_csrf_header']").attr("content");
    var contextPath = "${pageContext.request.contextPath}";
    var hos_id = "${hosmypageinfoV0.hos_id}";

    var formData = new FormData();
    //var files = $("#img_upload")[0].files;

    // fileInfoArr 배열의 파일 정보를 formData에 추가
    for (var i = 0; i < fileInfoArr.length; i++) {
        formData.append('files', fileInfoArr[i]);
    }

    $.ajax({
        beforeSend: function (xhr) {
            xhr.setRequestHeader(header, token);
        },
        url: contextPath + "/hos_MypageInfo/imgUpload.do",
        data: formData,
        processData: false,
        contentType: false,
        type: "POST",
        success: function (result) {
            alert("썸네일 업로드가 되었습니다.");
            window.location.href = contextPath + "/hos_MypageInfo/hosInformation.do?hos_id=" + hos_id;
        },
        error: function (err) {
            console.log("err", err);
        },
    });
}
```

previewImage function:

- It retrieves the selected files from the targetObj and converts them into an array.
- By detecting the browser, it distinguishes between IE and non-IE browsers to implement the preview functionality.

dataSubmit function:

- Since multiple is specified, it implements the upload functionality using a for loop for the number of files.
- It utilizes Ajax to execute formData to /hos_MypageInfo/imgUpload.do.

JavaScript in hosApplication.jsp

```
//썸네일 미리보기.
function previewImage(targetObj, View_area) {
    var files=targetObj.files;
    fileArr=Array.prototype.slice.call(files);
    var preview = document.getElementById(View_area); //div id
    var ua = window.navigator.userAgent;
    var imgBox = document.getElementById('imgBox');
    // 기존의 이미지를 삭제
    imgBox.innerHTML = '';
    //ie일때(IE8 이하에서만 작동)
    if (ua.indexOf("MSIE") > -1) {
        targetObj.select();
        try {
            var src = document.selection.createRange().text; // get file full path(IE9, IE10에서 사용 불가)
            var ie_preview_error = document.getElementById("ie_preview_error_" + View_area);
            if (ie_preview_error) {
                preview.removeChild(ie_preview_error); //error가 있으면 delete
            }
            var img = document.getElementById(View_area); //이미지가 뿌려질 곳
            //이미지 로딩, sizingMethod는 div에 맞춰서 사이즈를 자동조절 하는 역할
            img.style.filter = "progid:DXImageTransform.Microsoft.AlphaImageLoader(src='"+src+"', sizingMethod='scale')";
        } catch (e) {
            if (!document.getElementById("ie_preview_error_" + View_area)) {
                var info = document.createElement("p");
                info.id = "ie_preview_error_" + View_area;
                info.innerHTML = e.name;
                preview.insertBefore(info, null);
            }
        }
        //ie가 아닐때(크롬, 사파리, FF)
    } else {
        var files = targetObj.files;
        for ( var i = 0; i < files.length; i++) {
            var file = files[i];
            fileInfoArr.push(file);
            var imageType = /image.*/; //이미지 파일일경우만.. 뿌려준다.
            if (!file.type.match(imageType))
                continue;
            var span=document.createElement('span');
            span.id="img_id_" +i;
            span.style.width = '150px';
            span.style.height = '150px';
            span.style.marginRight = '15px';
            preview.appendChild(span);
            var img = document.createElement("img");
            img.className="addImg";
            img.classList.add("obj");
            img.file = file;
            img.style.width='inherit';
            img.style.height='inherit';
            img.style.cursor='pointer';
            const idx=i;
            img.onclick=()=>fileRemove(idx); //이미지를 클릭했을 때.
            span.appendChild(img);
            if (window.FileReader) { // Firefox, Chrome, Opera 확인.
                var reader = new FileReader();
                reader.onloadend = (function(aImg) {
                    return function(e) {
                        aImg.src = e.target.result;
                    };
                })(img);
                reader.readAsDataURL(file);
            } else { //safari is not supported FileReader
                //alert('not supported FileReader');
                if (!document.getElementById("sfr_preview_error_" + View_area)) {
                    var info = document.createElement("p");
                    info.id = "sfr_preview_error_" + View_area;
                    info.innerHTML = "not supported FileReader";
                    preview.insertBefore(info, null);
                }
            }
        }
    }
}
```

Project: Everymal

Major Code IV: Multiple Image Upload

HosMypageInfoControllerImpl.java

```
@RequestMapping(value = "/hos_MypageInfo/imgUpload.do", method = RequestMethod.POST)
public void imgUpload(HosMypageInfoVO hosmypageinfoVO, MultipartHttpServletRequest req,
HttpServletResponse response) throws Exception {
    HttpSession session = request.getSession();
    String hos_code = (String) session.getAttribute("hos_code");
    //String hos_code="hos0001";

    List<MultipartFile> multipartFileList = new ArrayList<>();
    try {
        MultiValueMap<String, MultipartFile> files = req.getMultiFileMap();
        for (String key : files.keySet()) {
            List<MultipartFile> fileList = files.get(key);
            for (MultipartFile file : fileList) {
                if (file.isEmpty()) continue;
                multipartFileList.add(file);
            }
        }
    }

    if (multipartFileList.size() > 0) {
        String savePath = "/Users/kyungminkim/Desktop/everymal-main/emptyProject/src/main/webapp/
resources/imgRepo/hos_images/thumbnail/" + hos_code;
        // 기존 이미지 파일 삭제
        File directory = new File(savePath);
        if (directory.exists() && directory.isDirectory()) {
            File[] existingFiles = directory.listFiles();
            for (File existingFile : existingFiles) {
                existingFile.delete();
            }
        }

        // 디렉토리 생성
        if (!directory.exists()) {
            directory.mkdirs();
        }

        for (int i = 0; i < multipartFileList.size(); i++) {
            MultipartFile file = multipartFileList.get(i);
            if (file.isEmpty()) continue;
            String originalfilename = file.getOriginalFilename();
            String savefilename = savePath + "/" + originalfilename;
            file.transferTo(new File(savefilename));

            if (i == 0) {
                hosmypageinfoVO.setHimg1(originalfilename);
            } else if (i == 1) {
                hosmypageinfoVO.setHimg2(originalfilename);
            } else if (i == 2) {
                hosmypageinfoVO.setHimg3(originalfilename);
            }
        }
    }

    String hos_id = (String) session.getAttribute("log_id");
    String hosCode = hosmypageinfoDAO.searchuploadimg(hos_code);
    if (hosCode == null || hosCode.isEmpty()) {
        hosmypageinfoVO.setHos_code(hos_code);
        hosmypageinfoDAO.newuploadimg(hosmypageinfoVO);
    } else {
        hosmypageinfoVO.setHos_code(hos_code);
        hosmypageinfoService.uploadimg(hosCode, hosmypageinfoVO);
    }
} catch (Exception e) {
    e.printStackTrace();
}
}
```

/hos_MypageInfo/imgUpload.do

- Extract image files received through requests into a list format, and if the files are not empty, save the images to the web server.
- If the directory corresponding to the hospital code does not exist in the savePath, create it and save the images.
- Delete all existing image files and replace them with a maximum of three newly uploaded images.
- Use a for loop to save the images in Himg1/Himg2/Himg3 according to the order.
- If there is already a thumbnail in the existing image, update the table using the uploadimg function through the searchuploadimg function. If there is no existing thumbnail, insert into the table using the newuploadimg function.