

Software Engineering CSC 648/848 Spring 2021

Gator Connection

A One Stop Website for SF State Gators

Team 05

Team Lead/Github Master: Alec Stephen Tenefrancia alectene@mail.sfsu.edu

Frontend Lead/Document Master: Lakshita Chugh

Backend Lead: Angelo Gloria Reyes

Frontend member: Bikram Tamang

Backend member: Benjamin Patrick Kao

Frontend member: Carmen Denisse Paisano

Backend member: Jiaxin Yu

Milestone 2

Milestone/Version	Date
M2V1	04/01/21

Table of Contents

Data Definitions V2	3
Functional Requirements V2	6
Priority 1 Requirements	6
Priority 2 Requirements	8
Priority 3 Requirements	9
UI Mockups and Storyboards	11
High Level Database Architecture and Organization	38
Business Rules	38
Entity, Attribute, Relationship, Domain Descriptions	40
Entity Relationship Diagram (ERD)	42
Database Model/Enhanced Entity Relationship (EER)	44
Database Management System (DBMS)	45
Media Storage	45
Search/Filter Architecture and Implementation	45
High Level APIs and Main Algorithms	46
API Endpoints	46
High Level UML Diagrams	49
High Level Application Network and Deployment Diagrams	51
Project Risks	52
Project Management	54
List of Contributions	56

Data Definitions V2

1. Unregistered User/ Guest

These are the people who aren't registered with the school system yet. They will be treated as guests, and would not be eligible for any of our student/staff services. This is an administrative task.

2. Registered User

A person who is registered with us using their email would have an account with us. He/She will follow the Create Account path. His/Her data is already registered with us, he/she will have the authority to create their password here.

3. Account

An account contains identification data about the registered user that created the account. The data identifies what type of registered user the user is, as well as helps authentication registered users for restricted features such as posting announcements.

- a) **Student:** A student account gives registered users the ability to search housing listings, post restaurant reviews, and make purchase requests for posted items and housing listings.
- b) **Admin:** An admin account gives registered users the power to post public announcements. Admin accounts are split into more specific roles as to more easily group announcements together when filtering/searching.
 - i) **Sports/Athletics:** A registered user who has this type of account is in charge of a specific sport that he/she registered with.
 - ii) **Organization/Club:** A registered user who has this type of account is in charge of a specific organization that he/she registered with.
 - iii) **School Department:** A registered user who has this type of account is a part of SFSU staff.
- c) **Super User:** A super user account gives registered users the power to approve restaurant additions, administrative account creation requests, and registered user account upgrade requests.

4. Notifications

Receiving notifications about housing, submissions, etc.

- a) **Housing Notifications:** Registered users will receive email notifications about housing such as if they receive a request for a listing.
- b) **Submission Notifications:** Registered users will receive email notifications about submissions such as if their submission has been approved or denied

5. Rate

This gives users the ability to rate various student organizations, restaurants. It'll prove to be a useful tool for fellow users.

- a) **Student Organization Rating:** Ratings that users can give organizations from 1-10. The average of all ratings will be shown on the organization.
- b) **Restaurants Rating:** Ratings that users can give restaurants from 1-10. The average of all ratings will be shown on the restaurants.

6. Review

Posts shown under restaurants made by registered users and above. Unregistered users can view these but they cannot rate or post a review.

7. Email

Being able to send emails to all of the users that are registered. This would be students and administration.

- a) **Housing emails:** A User who posted a listing of a house for sale will receive emails from people who are interested to purchase the listing.
- b) **Registration Verification email:** A User who creates an account will receive an email to their SFSU email saying that their account has been created
- c) **Reset password email:** An email which a user can request to reset their password

8. Announcements

Section designated to announcements that different departments make, along with the health centers, athletics, and organizations/clubs. These announcements are directly related to SFSU campus and only administrative users can post announcements.

- a) **Athletic Announcements:** Announcements made from athletic directors
- b) **Organization Announcements:** Announcements made from organizations

- c) **School Announcements:** Announcements made from the school

9. Restaurants

Section dedicated to restaurants around campus. Users can view, rate, and write reviews for that specific restaurant. Unregistered users can view these but they cannot rate or review.

10. Ecommerce/Listing

The main section where users can view the selected items for sale or housing available and make purchase requests for them. Registered users can post things for sale. Unregistered guests cannot view these items.

a) Items:

The item(s) listed for sale in the Ecommerce/Listing by a registered seller or for purchase by a registered buyer. Users can message the seller if they are interested in the item.

b) Housing:

The listing(s) of housing listed for sale in the Housing section. Users interested in a listing can fill out a form which includes their name, school year, SFSU email, and a little information about themselves. After filling out the form, the system will send out an email to the person who posted the listing, and it will be up to them if they wish to respond back.

Functional Requirements V2

Priority 1 Requirements

- Guest User
 1. A guest user shall be able to search for restaurants/food.
 2. A guest user shall be able to sort restaurant listings by rating.
 3. A guest user shall be able to search/navigate through announcements.
 4. A guest user shall be able to create an account using his/her unique SFSU email account.
 5. A guest user shall be able to navigate to a map of SFSU.
 6. A guest user shall be able to search for items on sale by category, price, etc.
- Registered User
 7. A registered user shall have all of the same permissions as guest users.
 8. A registered user shall be able to log in to his/her account using his/her unique SFSU email.
 9. A registered user shall be able to log in to his/her account using many devices.
 10. A registered user shall be able to log out of the website.
 11. A registered user shall be able to stay logged in if they have not logged out.
 12. A registered user shall be able to post reviews of restaurants/food.
 13. A registered user shall be able to make purchase requests for items.
 14. A registered user shall be able to search for items by category, price, etc.
 15. A registered user shall be able to see and search housing listings.
 16. A registered user shall be able to post a housing listing.
 17. A registered user shall be able to make a request for a housing listing.
 18. A registered user shall be able to edit a housing listing title he/she posted.
 19. A registered user shall be able to edit a housing listing description he/she posted.
 20. A registered user shall be able to edit a housing listing price he/she posted.
 21. A registered user shall be able to change a housing listing image he/she posted.
 22. A registered user shall be able to close a housing listing that he/she posted.
 23. A registered user shall be able to post many pending items for sale with pictures.
 24. A registered user shall be able to edit a post about an item for sale that he/she posted.
 25. A registered user shall be able to take down an item that he/she put up for sale.
 26. A registered user shall be able to create a restaurant request to add a restaurant to the restaurants listing by providing the name of the restaurant and location.

- 27. A registered user shall be notified if their restaurant location has been approved or denied, with a message explaining why
- 28. A registered user shall receive a notification about a house listing being closed if he/she had made a request to the house lister.
- 29. An upgraded registered user shall receive a notification when their administration privileges are taken away and why.
- 30. A registered user shall have a unique registered id.
- 31. A registered user shall be able to fill out a form for a housing request.
- Admin User
 - 32. An administrative user shall have all of the same permissions as registered users.
 - 33. An administrative user shall be able to post announcements.
 - 34. An administrative user shall be able to remove an announcement he/she had posted.
 - 35. An administrative user shall have a unique admin id.
- Super User
 - 36. A super user shall be able to log into the website.
 - 37. A super user shall be able to log out of the website.
 - 38. A super user shall have all of the same permissions as administrative users.
 - 39. A super user shall be able to approve/deny and close account upgrade requests,
 - 40. A super user shall be able to approve/deny and close administrative account creation requests.
- Sale Item
 - 41. A sale item shall be posted by a registered user.
 - 42. A sale item shall have a unique item id.
 - 43. A sale item shall have a title for the item.
 - 44. A sale item shall have a message describing the item.
 - 45. A sale item shall have one picture attached to it.
 - 46. A sale item shall be purchased by a registered user.
 - 47. A sale item shall be taken down by the registered user that posted it for sale.
 - 48. A sale item shall have a price.
- Housing Listing
 - 49. A housing listing shall be posted by a registered user.
 - 50. A housing listing shall have a unique housing id.
 - 51. A housing listing shall have a title.
 - 52. A housing listing shall have a message describing the item.
 - 53. A housing listing shall have a price.
 - 54. A housing listing shall have at least 1 picture attached to it.
 - 55. A housing listing shall be taken down by the registered user that posted it.
 - 56. A housing listing shall be requested by many registered users.
 - 57. A housing listing shall have a form to fill out for interested users.

- 58. A housing listing shall have a situation(available/close).
- Housing Listing Form
 - 59. A housing listing form shall be filled out by a registered user.
 - 60. A housing listing form shall fill out their full name.
 - 61. A housing listing form shall fill out their school year.
 - 62. A housing listing form shall fill out their school email.
 - 63. A housing listing form shall fill out their phone number.
 - 64. A housing listing form shall have a section dedicated to “an about” me.
 - 65. A housing listing form shall have an expected day they want to move in.
- Restaurant
 - 66. A restaurant shall have a name.
 - 67. A restaurant shall have a location.
 - 68. A restaurant shall have a unique restaurant id.
 - 69. A restaurant shall have a rating.
- Restaurant Request
 - 70. A restaurant request shall be created by one and only one registered user and above.
 - 71. A restaurant request shall have the name of the restaurant.
 - 72. A restaurant request shall have the location of the restaurant.
 - 73. A restaurant request shall be confirmed/denied and closed by one and only one super user.
- Organization
 - 74. An organization shall be able to post announcements.
- Announcement
 - 75. An announcement shall be posted by one and only one administrative user or organization.
 - 76. An announcement shall be able to be viewed by one or many users.
 - 77. An announcement shall be able to be taken down by the one who posted it.

Priority 2 Requirements

- Registered User
 - 1. A registered user shall be able to rate organizations.
 - 2. A registered user shall be able to change their rating of the organization.

- 3. A registered user shall be able to sort and search for restaurants by ratings/reviews.
- Sale Item
 - 1. When a sale item is taken down, all registered users who made a request for the item shall be notified of the closure.
- Housing Listing
 - 1. When a housing listing is taken down, all registered users who made a request shall be notified of the closure.
- Housing Listing Form
- Restaurant
 - 1. A restaurant shall be able to have their rating changed.
 - 2. A restaurant shall have many reviews written by registered users.
- Restaurant Request
 - 1. When a restaurant request is confirmed/denied the registered user who made the request will be notified.
- Organization
 - 1. An organization shall be able to be rated by many registered users.
 - 2. An organization shall be able to have their ratings changed.
- Announcement
 - 1. An announcement shall be able to be reviewed.
 - 2. An announcement shall be able to be rated.

Priority 3 Requirements

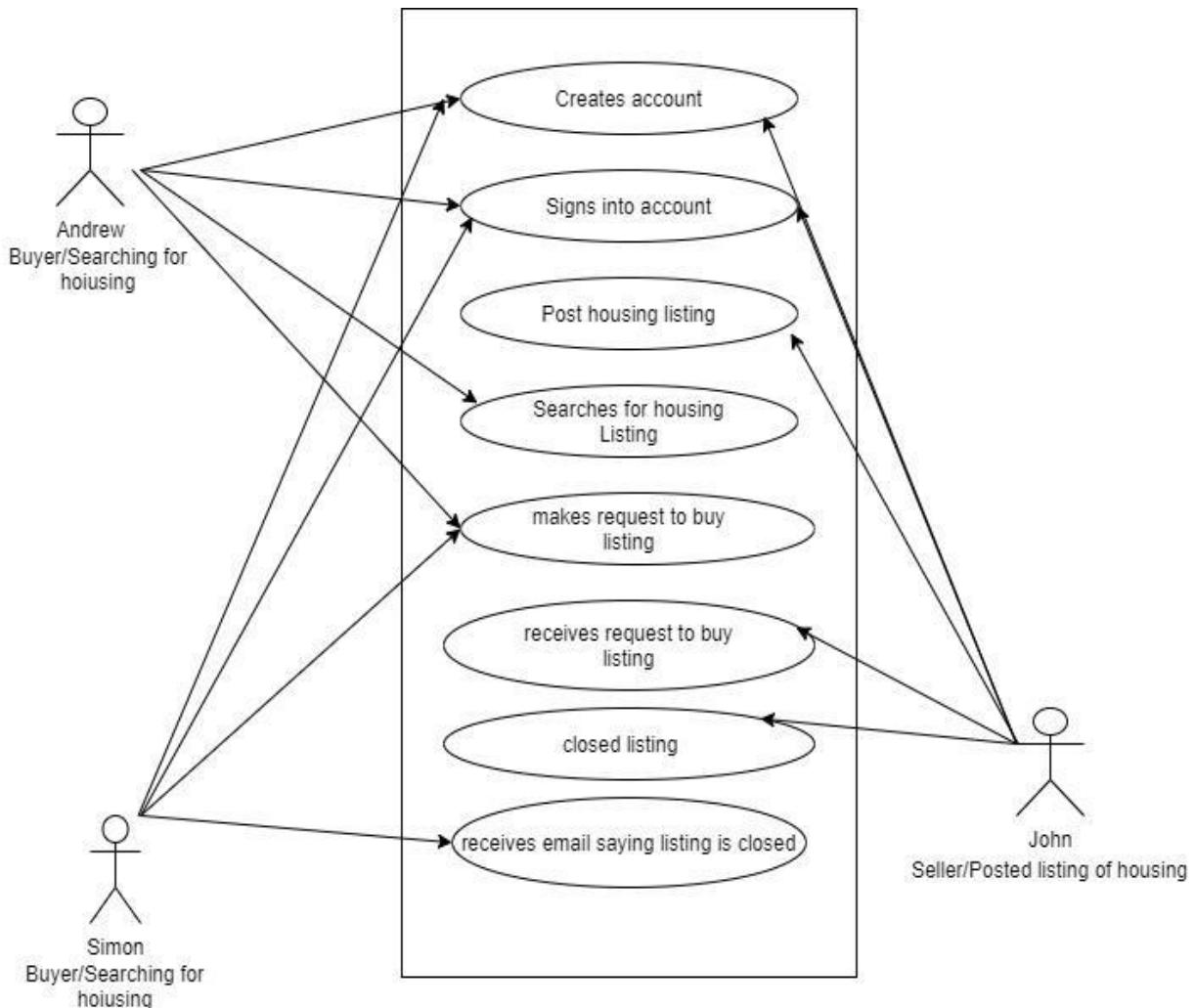
- Registered User
 - 1. A registered user shall be able to post reviews under event announcements.
 - 2. A registered user shall be able change any reviews they have under event announcements.
 - 3. A registered user shall be able to rate a housing listing.
 - 4. A registered user shall receive a notification if their housing post has been approved.
 - 5. A registered user shall receive a notification if their housing post has been denied.
 - 6. A registered user shall receive a notification if their post has been approved.
 - 7. A registered user shall receive a notification if their post has been denied.
 - 8. A registered user shall be able to rate any of the item listings on sale.
 - 9. A registered user shall be able to change their rating of the item.
 - 10. A registered user shall be able to report housing listings and provide reasoning for the report.
 - 11. A registered user shall be able to request to upgrade his/her account to an administrative account with a valid organization, role, and organization email.

12. A registered user shall receive a notification about purchase requests made on his/her account.
 13. A registered user shall receive a notification about purchase requests made on items he/she posted for sale.
 14. A registered user shall receive a notification about a house listing request for a house listing that he/she posted that includes information about the person who requested, such as his/her name and email address.
 15. A registered user who requested an account upgrade shall receive a notification when the request has been approved and the date of when their administration privileges will be taken away.
 16. A registered user who requested an account upgrade shall receive a notification when the request has been denied.
- Super User
 1. A super user shall be able to approve pending posts for restaurant reviews.
 2. A super user shall be able to reject pending posts for restaurant reviews.
 3. A super user shall be able to approve pending posts for items for sale.
 4. A super user shall be able to reject pending posts for items for sale.
 5. A super user shall be able to approve pending housing listings for sale.
 6. A super user shall be able to reject pending housing listings for sale.
 - Sale Item
 1. A sale item shall be able to be approved by a super user.
 2. A sale item shall be able to be rejected by a super user.
 - Housing Listing
 1. A housing listing shall be able to be approved by a super user.
 2. A housing listing shall be able to be rejected by a super user.

UI Mockups and Storyboards

Case 1: Freshman Student looking for housing at SFSU

Actors: Andrew(Buyer, Searching for housing), Simon (Buyer, Searching for housing), John(Seller, Posted listing of housing)



CREATE ACCOUNT

ANDREW

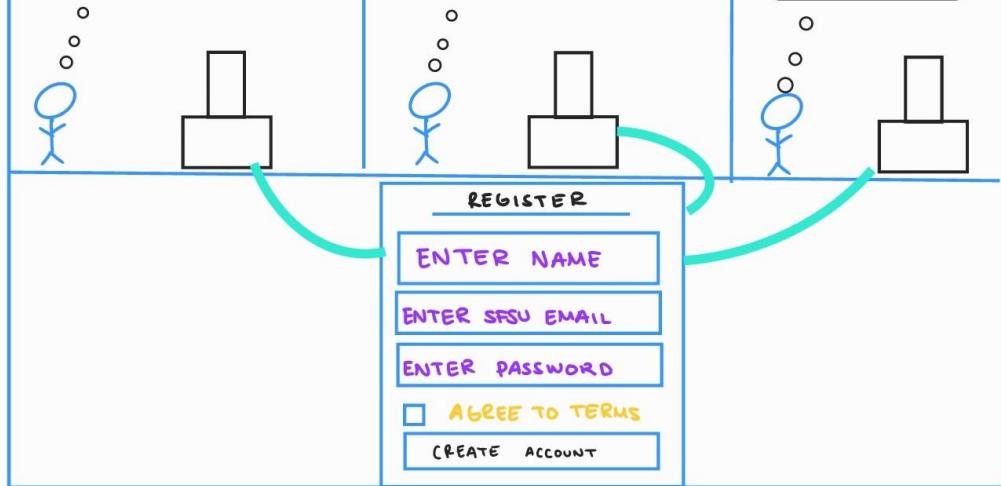
Simon

John

TIME TO MAKE AN ACCOUNT ON GATOR CONNECTIONS TO SEARCH FOR HOUSING SO IT GUARANTEES I LIVE WITH SFSU STUDENTS!

TIME TO MAKE AN ACCOUNT ON GATOR CONNECTIONS TO SEARCH FOR HOUSING SO IT GUARANTEES I LIVE WITH SFSU STUDENTS!

TIME TO MAKE AN ACCOUNT ON GATOR CONNECTION SO I CAN SELL MY HOUSING LISTING!



LOG IN

ANDREW

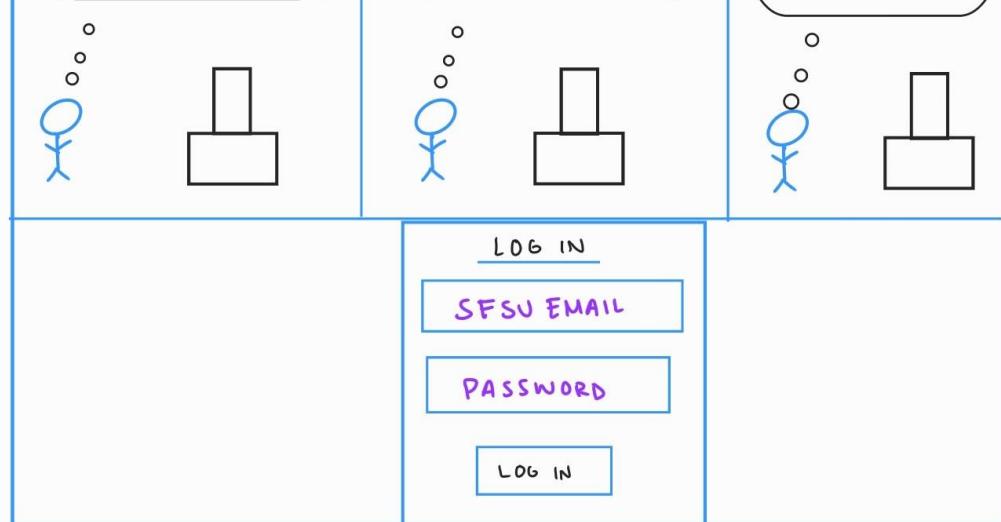
Simon

John

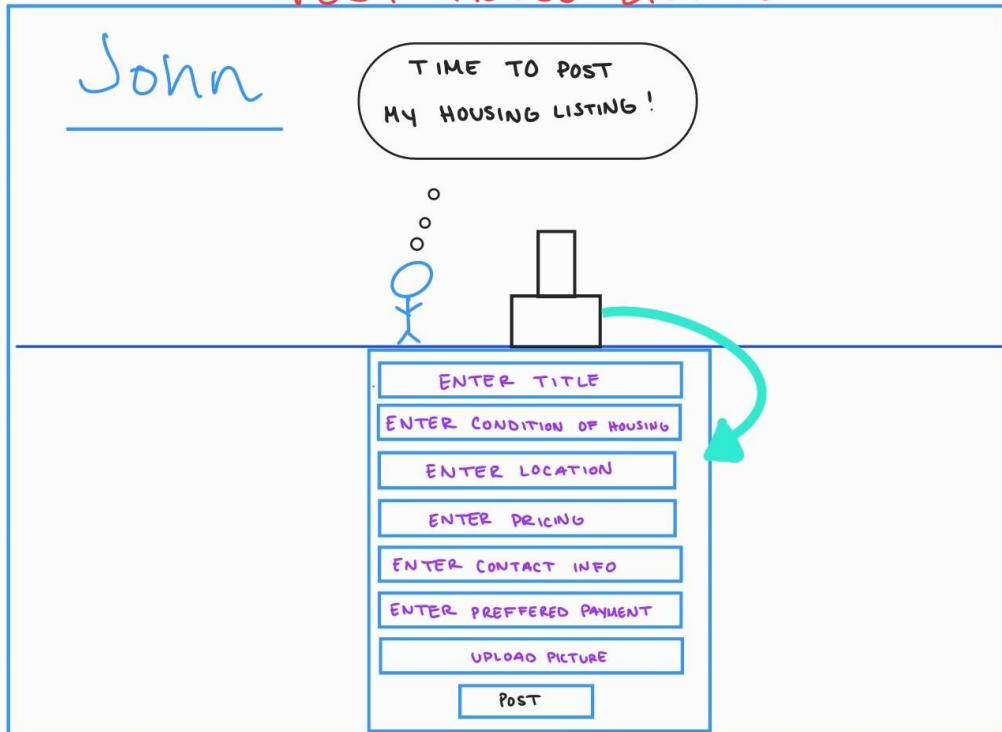
NOW THAT I HAVE AN ACCOUNT, I CAN LOG IN

NOW THAT I HAVE AN ACCOUNT, I CAN LOG IN

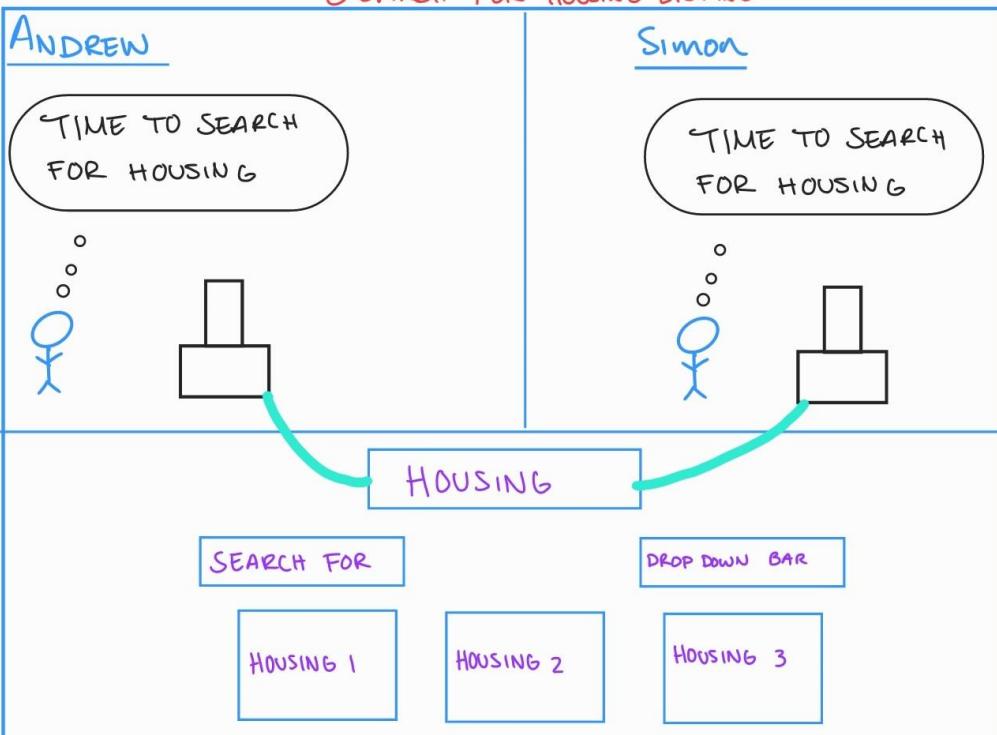
NOW THAT I HAVE AN ACCOUNT, I CAN LOG IN



POST HOUSE LISTING



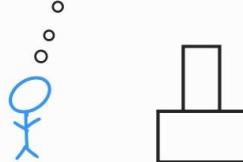
SEARCH FOR HOUSING LISTING



MAKE REQUEST TO BUY LISTING

ANDREW

THIS HOUSING LISTING FROM
JOHN LOOKS NICE, I'LL MESSAGE
HIM ABOUT IT!



Simon

THIS HOUSING LISTING FROM
JOHN LOOKS NICE, I'LL MESSAGE
HIM ABOUT IT!



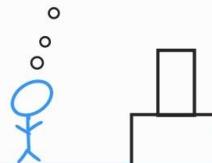
JOHN'S HOUSE

SF
\$800
555-555-555
VENMO FOR RENT
PICTURE OF
HOUSE

RECEIVE REQUEST TO BUY LISTING

John

I HAVE A TEXT FROM
ANDREW AND SIMON REGARDING
THE HOUSING, I THINK I'LL GO
WITH ANDREW!



PHONE :



ANDREW: HEY I'M INTERESTED
IN THE HOUSE ...

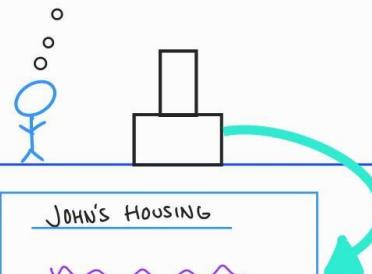
SIMON: HEY I'M INTERESTED
IN THE HOUSE ...



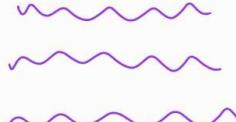
CLOSED LISTING

John

TIME TO CLOSE
THE LISTING SINCE I'M
GOING WITH ANDREW



JOHN'S HOUSING

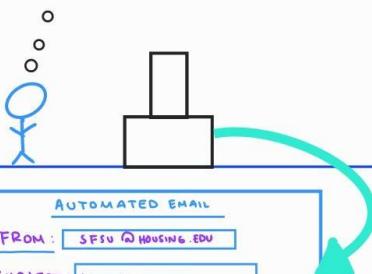


TAKE DOWN POSTING?

RECEIVES EMAIL SAYING LISTING IS CLOSED

Simon

THAT SUCKS THAT JOHN'S LISTING
IS CLOSED, TIME TO SEARCH
FOR ANOTHER



AUTOMATED EMAIL

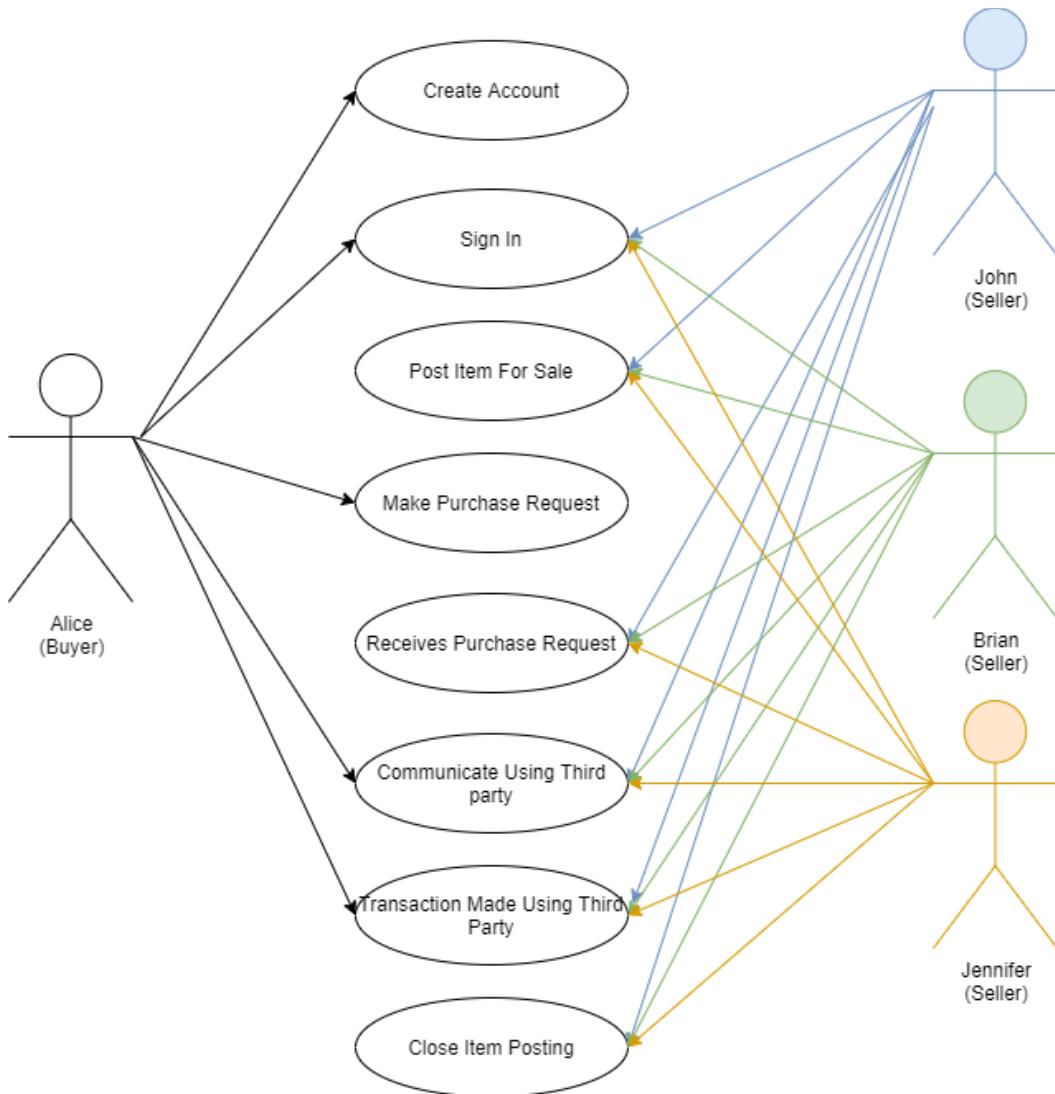
FROM : SFSU@HOUSING.EDU

SUBJECT: LISTING CLOSED

BODY: HELLO SIMON,
WE REGRET TO INFORM
YOU THAT JOHN'S LISTING
IS CLOSED.

Case 2: Student at SFSU looking for cheaper textbooks

Actors: Alice(Buyer), John(Seller), Brian(Seller), Jennifer(Seller)



CREATE ACCOUNT - ALICE



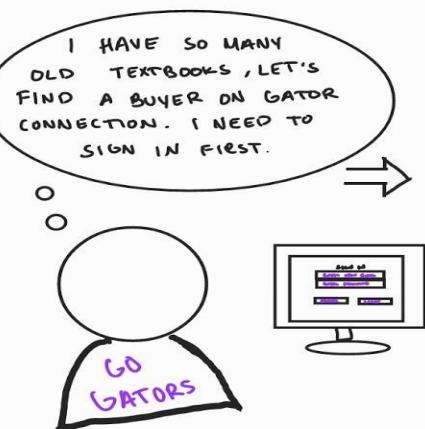
A hand-drawn illustration of a registration form titled "REGISTER". It contains three input fields: "ENTER NAME", "ENTER SFSU EMAIL", and "CREATE PASSWORD". Below these fields is a checkbox labeled "AGREE TO TERMS OF PRIVACY AND SERVICE" with a checkmark. At the bottom is a "CREATE ACCOUNT" button.

SIGN IN - ALICE



A hand-drawn illustration of a sign-in form titled "SIGN IN". It contains two input fields: "ENTER SFSU EMAIL" and "ENTER PASSWORD". At the bottom are two buttons: "CANCEL" and "LOGIN".

SIGN IN - JOHN, BRIAN, JENNIFER



A hand-drawn illustration of a sign-in form titled "SIGN IN". It contains two input fields: "ENTER SFSU EMAIL" and "ENTER PASSWORD". At the bottom are two buttons: "CANCEL" and "LOGIN".

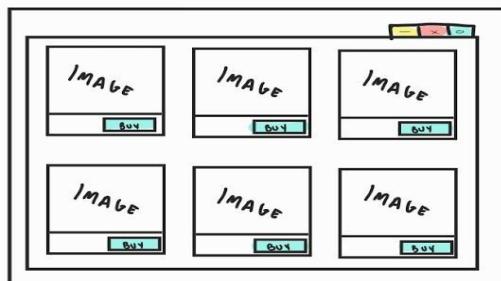
POST ITEM - JOHN, BRIAN, JENNIFER



CREATE ITEM SALE POST

ENTER TITLE
ENTER CONDITION
ENTER OTHER DETAILS
ENTER PRICING
UPLOAD PICTURE
CANCEL CREATE POST

MAKE PURCHASE REQUEST - ALICE



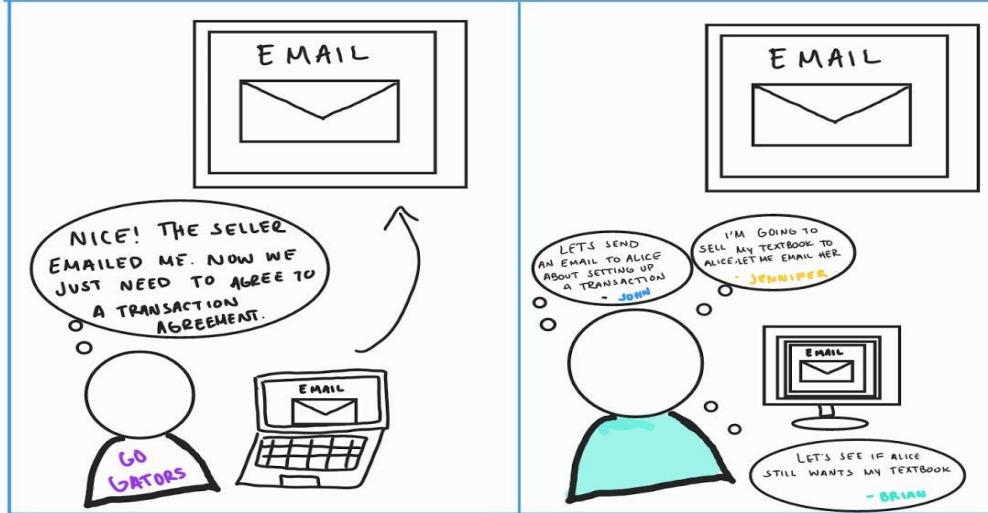
RECEIVING PURCHASE REQUEST - JOHN, BRIAN, JENNIFER



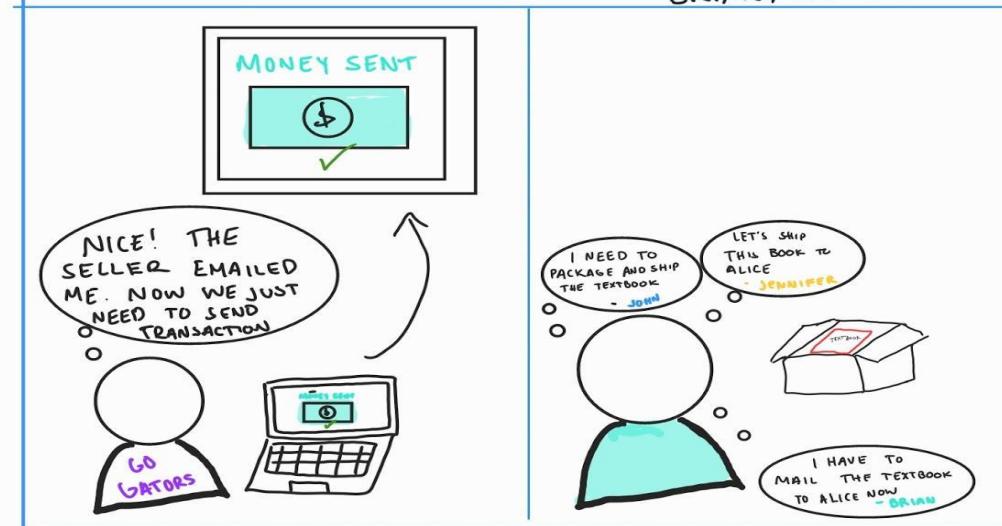
PURCHASE REQUESTS

IMAGE POST TITLE REQUEST
BUYER: ALICE
SEE CONTACT INFO

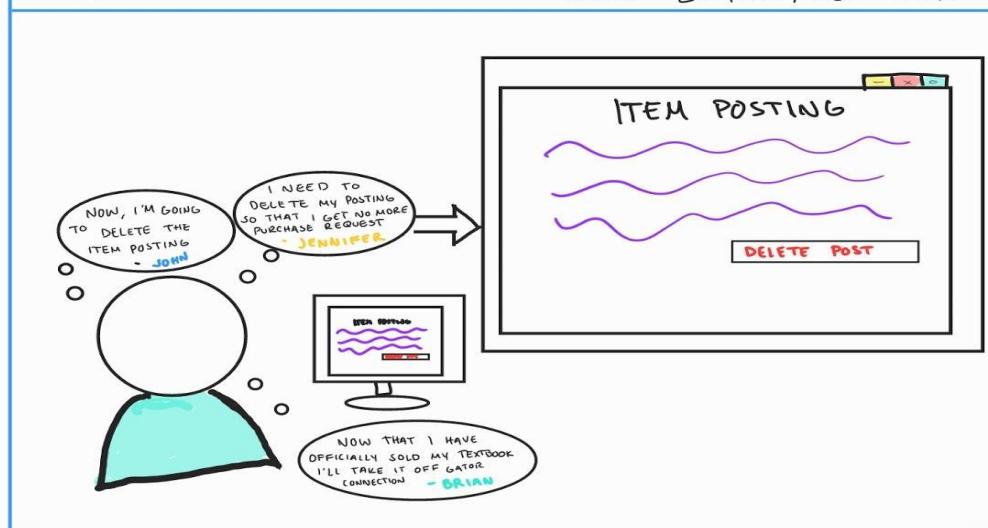
THIRD PARTY COMMUNICATION - ALICE, JOHN, BRIAN, JENNIFER



COMPLETE TRANSACTION - ALICE, JOHN, BRIAN, JENNIFER

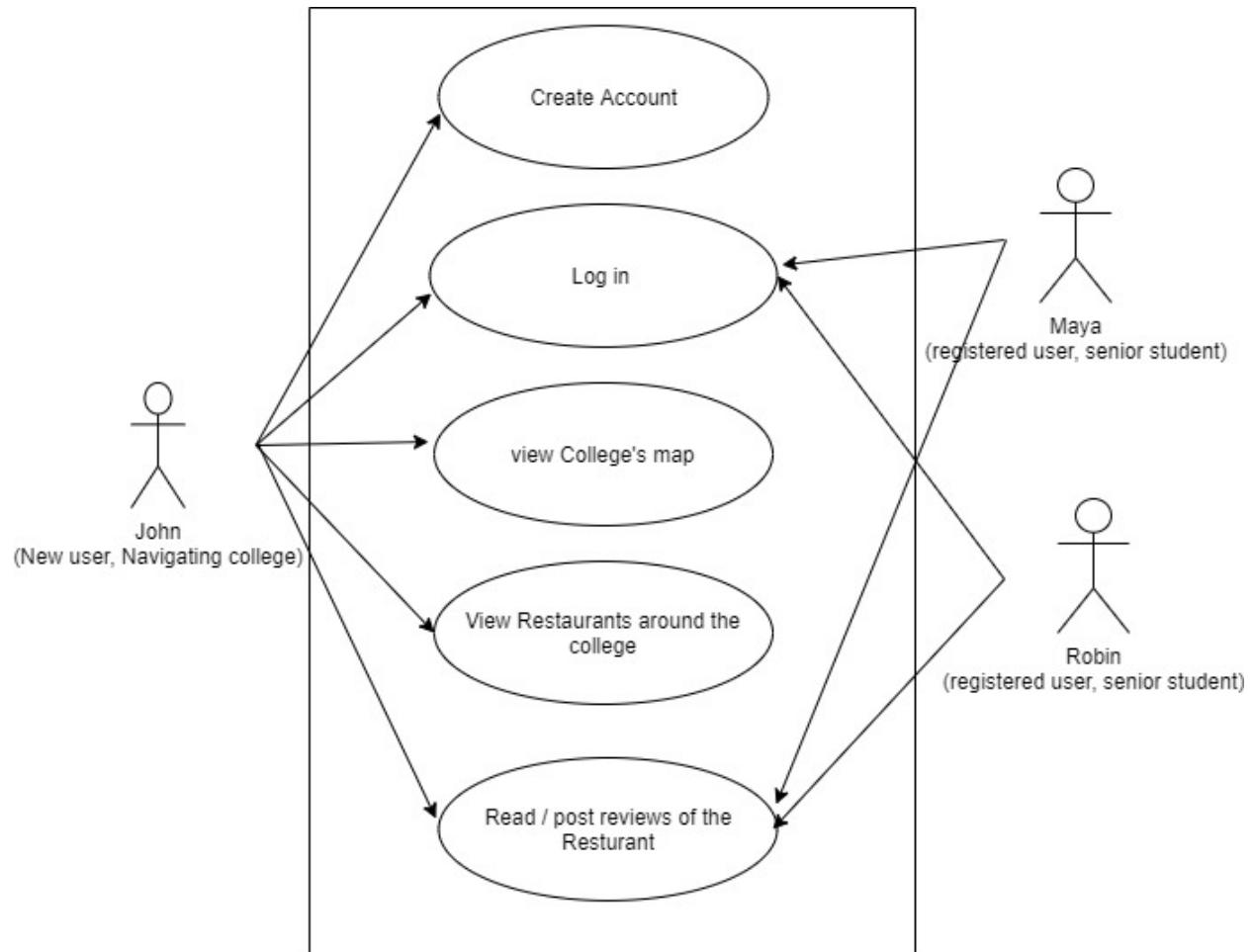


CLOSE POSTING - JOHN, BRIAN, JENNIFER

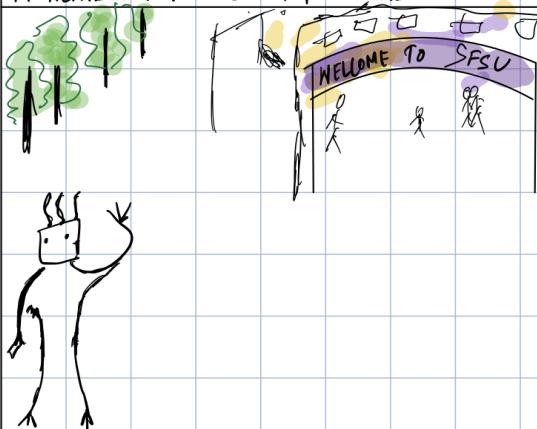


Use Case 3: Student at SFSU looking for food on campus, with his two friends outside SFSU accompanying him

Actors: John(New User), Maya(Unregistered User), Robin(Unregistered User)

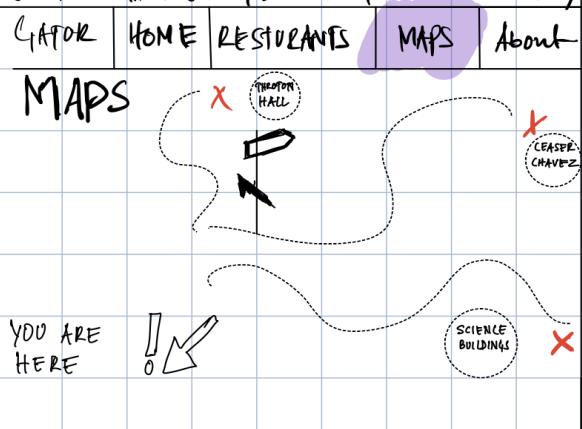


JOHN, A FRESHMAN AT SFSU, WANTS TO BE FAMILIAR WITH THE CAMPUS & ITS SURROUNDINGS



AFTER A LITTLE WHILE,
JOHN GETS HUNGRY

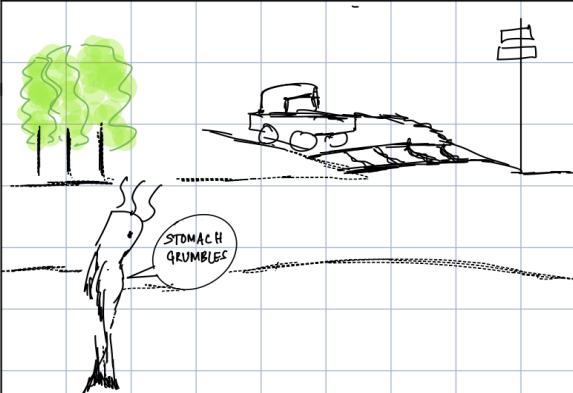
THROUGH THE GATOR CONNECTION WEBSITE, JOHN CAN SEE ALL THE CAMPUS BUILDINGS AND ITS SURROUNDING



HE AGAIN VISITS GATOR CONNECTION WHERE HE CAN SEE RESTAURANTS AROUND CAMPUS HOPEFULLY TO FIND SOMETHING GOOD TO EAT

AFTER A LITTLE WHILE,
JOHN GETS HUNGRY

HE AGAIN VISITS GATOR CONNECTION WHERE HE CAN SEE RESTAURANTS AROUND CAMPUS HOPEFULLY TO FIND SOMETHING GOOD TO EAT



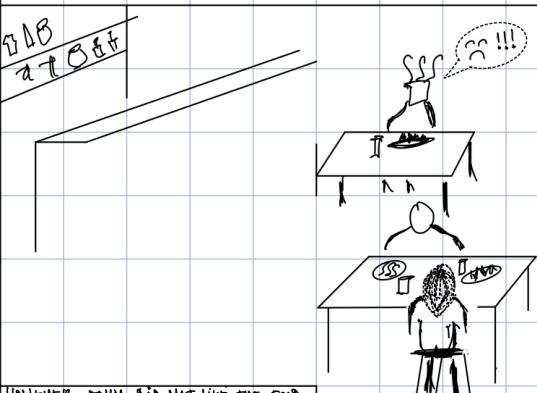
HOME RESTAURANTS MAPS HOUSINGS ITEMS

KRABBY PATTY ★★★★★

ICHIRAKU RAMEN ★★★★★

CENTRAL PERK ★★★★★

APRTER LOOKING AT SOME REVIEWS POSTED BY SFSU STUDENTS
JOHN GOES TO A RESTAURANT LOOKING AT ITS HIGH RATINGS



HOWEVER JOHN DID NOT LIKE THE FOOD
& HE THOUGHT ABOUT LEAVING A REVIEW

- X
GATOR CONNECTION.

Sorry, only registered users are able to post a review.

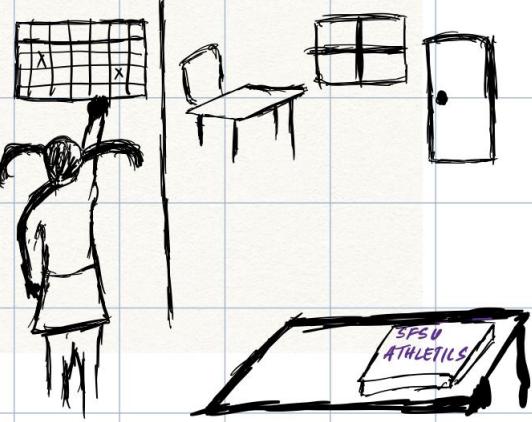
[Sign me up!](#)

JOHN then signs up using SFSU email and leaves a review to the restaurant.

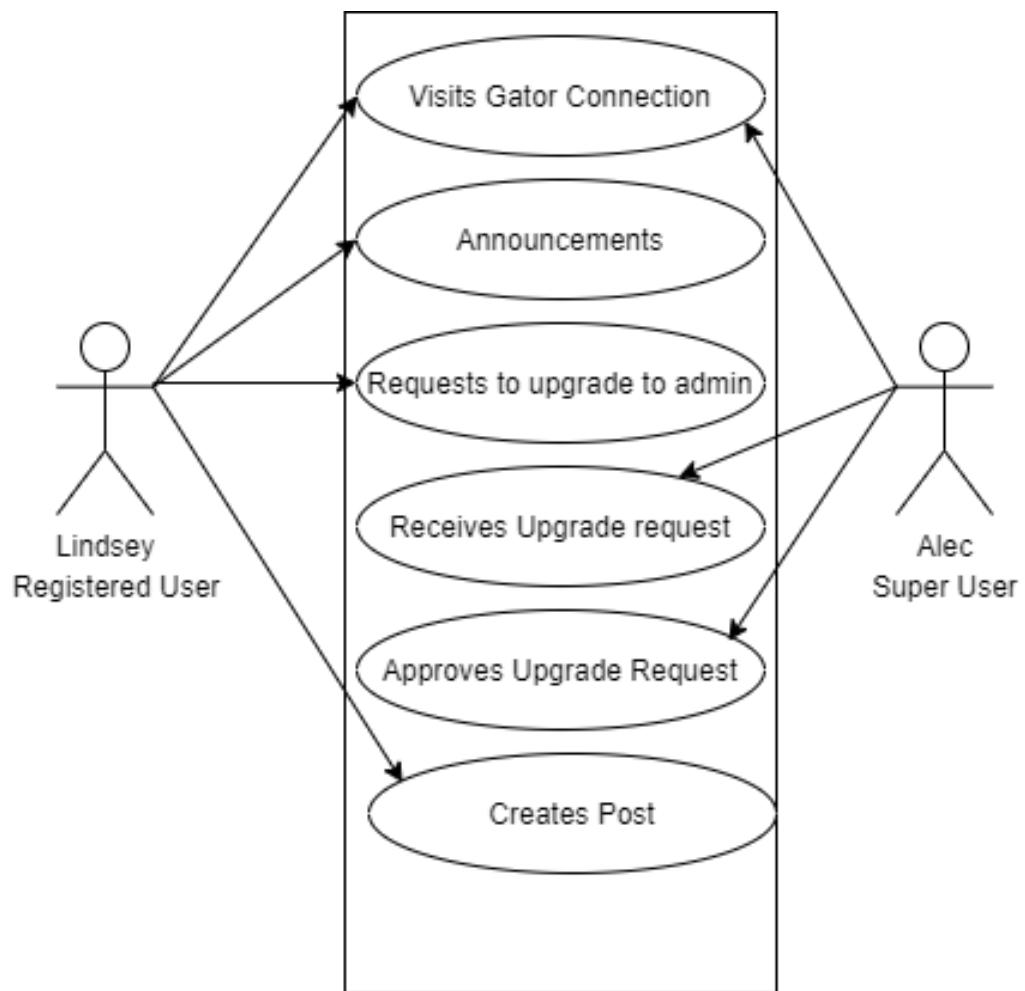
Case 4: Athletic Director at SFSU wants to announce upcoming events

Actors: Stephanie(User, Athletic Director)

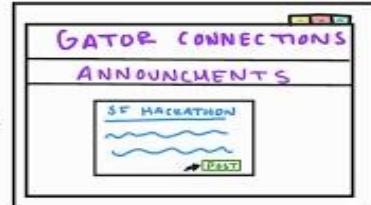


<p>Stephanie Hawkins is a athletics director in the SFSU</p>	<p>She realized Most of the students use GATOR connection & decides to post the announcement there.</p>
	<p>GATOR CONNECTION ANNOUNCEMENTS</p> <ul style="list-style-type: none"> MAKE NEW ANNOUNCEMENT EDIT ANNOUNCEMENTS DELETE ANNOUNCEMENTS
<p>SHE WANTS TO ANNOUNCE the UPCOMING Basketball GAME hosting at SFSU</p>	
<p>However, making announcements takes special privilege and she gets an error.</p>	<p>Stephanie successfully creates an administrative account and could successfully post an announcement & everyone can see it.</p>
<p>GATOR CONNECTION</p> <p>ANNOUNCEMENTS</p> <p>Sorry, but announcements can be made only with administrative account if you are a SFSU director please create a administrative account.</p> <p>Create a administrative account</p>	<p>GATOR CONNECTION</p> <p>HOME SHOP ANN</p> <p>ANNOUNCEMENTS</p> <p>ADVERTISEMENTS</p> <p>Stephanie's announcement is shown on the home page Carousel.</p>

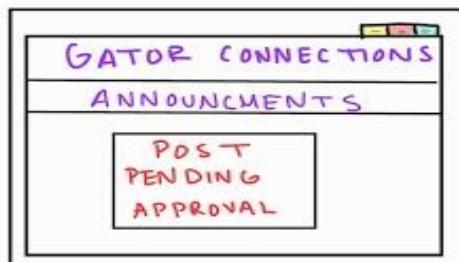
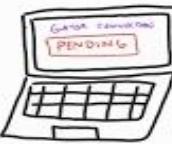
Case 5: President of SF Hacks wants to post about her upcoming events
Actors: Lindsey(Registered User), Alec(Super User)



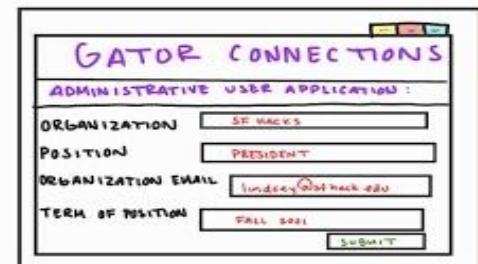
LINDSEY WANTS TO POST SF HACK'S SOCIAL MEDIA, ANNOUNCEMENTS, AND FUTURE EVENTS ON GATOR CONNECTIONS.



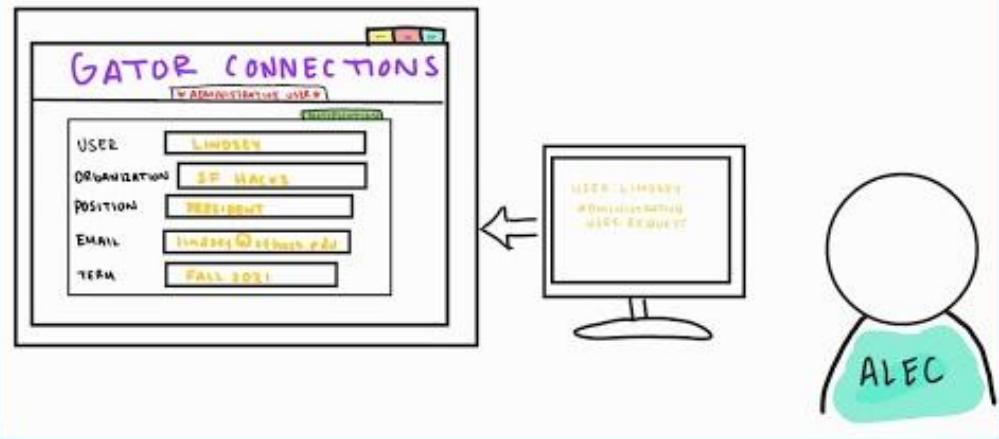
LINDSEY TRIES TO POST AN ANNOUNCEMENT BUT SHE REALIZES SHE DOESN'T HAVE FULL ACCESS AS ADMINISTRATIVE USER, THEREFORE HER POST IS PENDING.



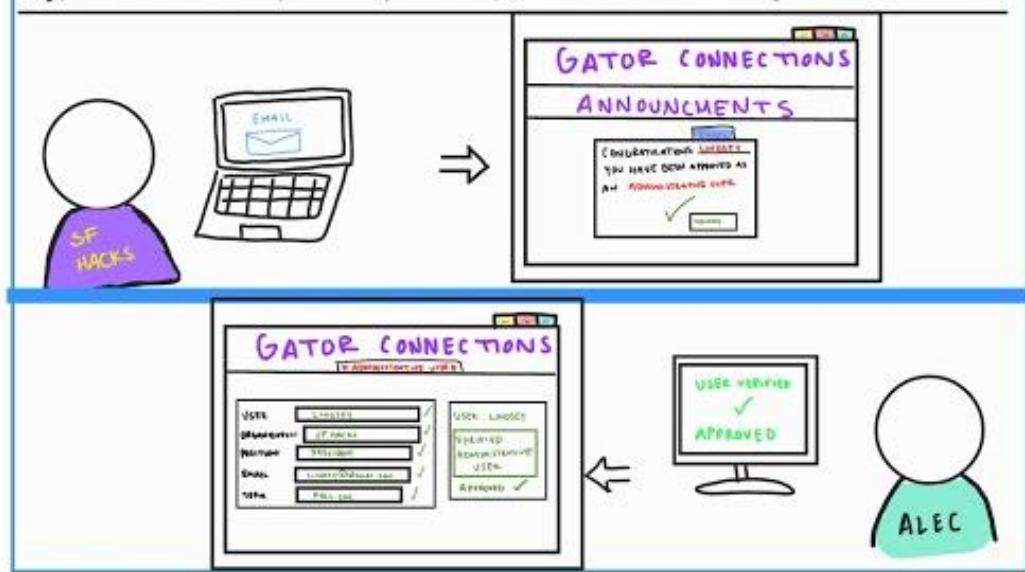
LINDSEY DECIDES THAT SINCE SHE IS PRESIDENT OF SF HACKS SHE CAN APPLY AS AN ADMINISTRATIVE USER SO HER POSTS DON'T HAVE TO GO THROUGH A PENDING PROCESS



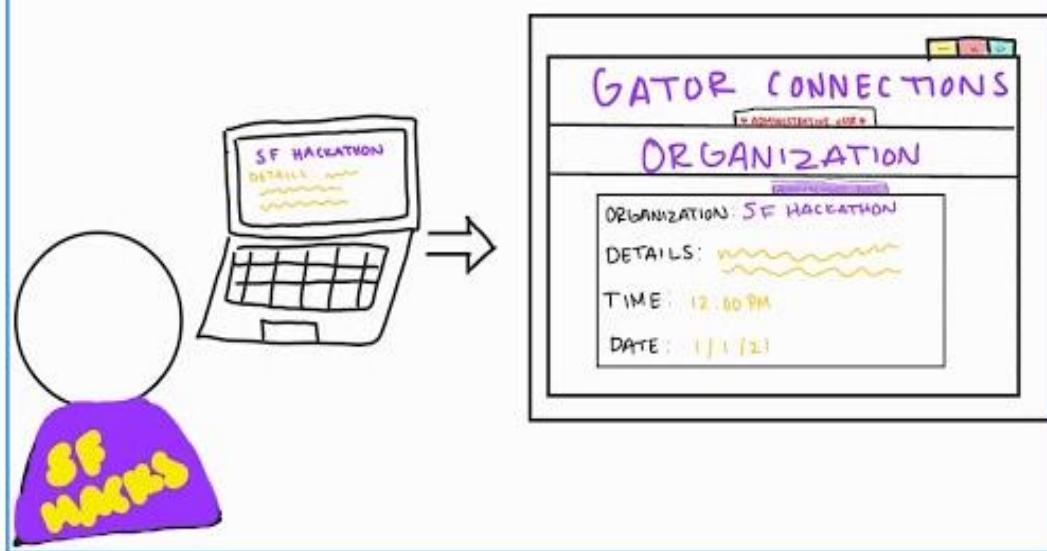
WHILE LINDSEY IS WAITING FOR OFFICIAL CONFIRMATION, SUPER USER ALEC HAS RECEIVED HER REQUEST.



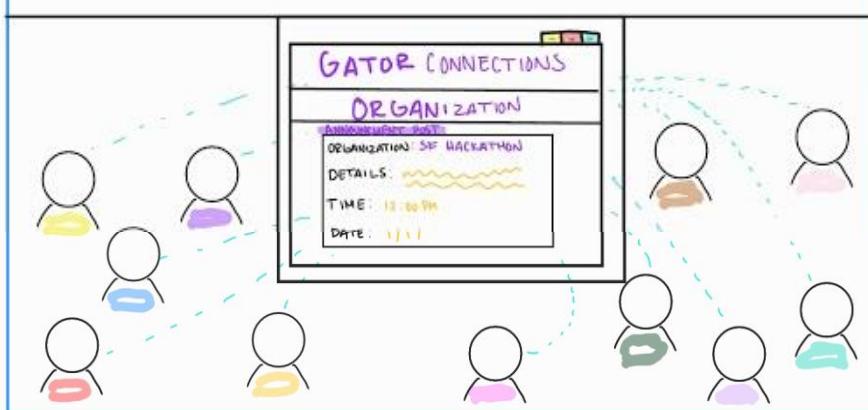
ALEC VERIFIES LINDSEY'S INFORMATION AND APPROVES HER REQUEST AND LINDSEY RECIEVES AN EMAIL NOTIFYING HER APPROVAL TO UPGRADE HER ACCOUNT.



NOW THAT LINDSEY HAS ADMINISTRATIVE PRIVILEGES AS PRESIDENT SHE POSTS AN UPCOMING SF HACKATHON EVENT WITH DETAILS.

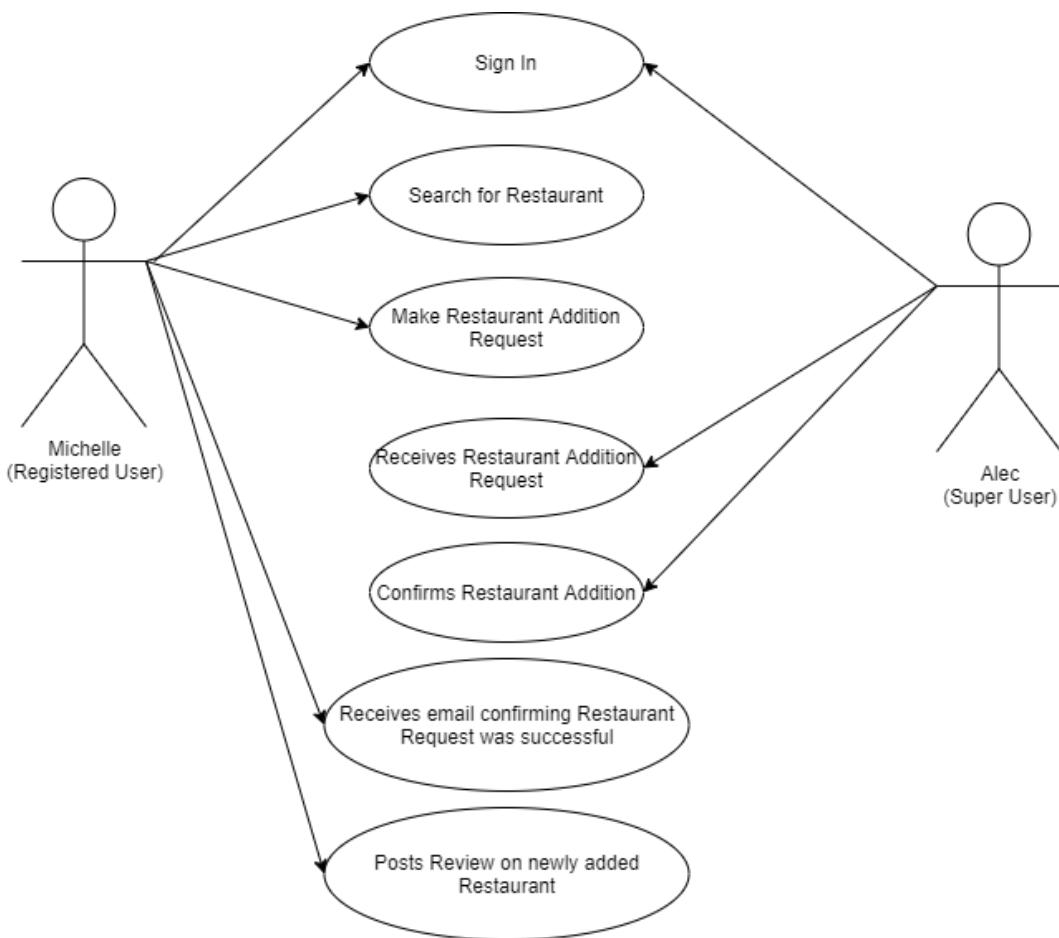


AFTER POSTING HER POST, LINDSEY WENT TO THE "ORGANIZATIONS" SECTION OF GATOR CONNECTIONS AND HER ANNOUNCEMENT WAS AVAILABLE FOR EVERYONE TO SEE RIGHT AWAY.

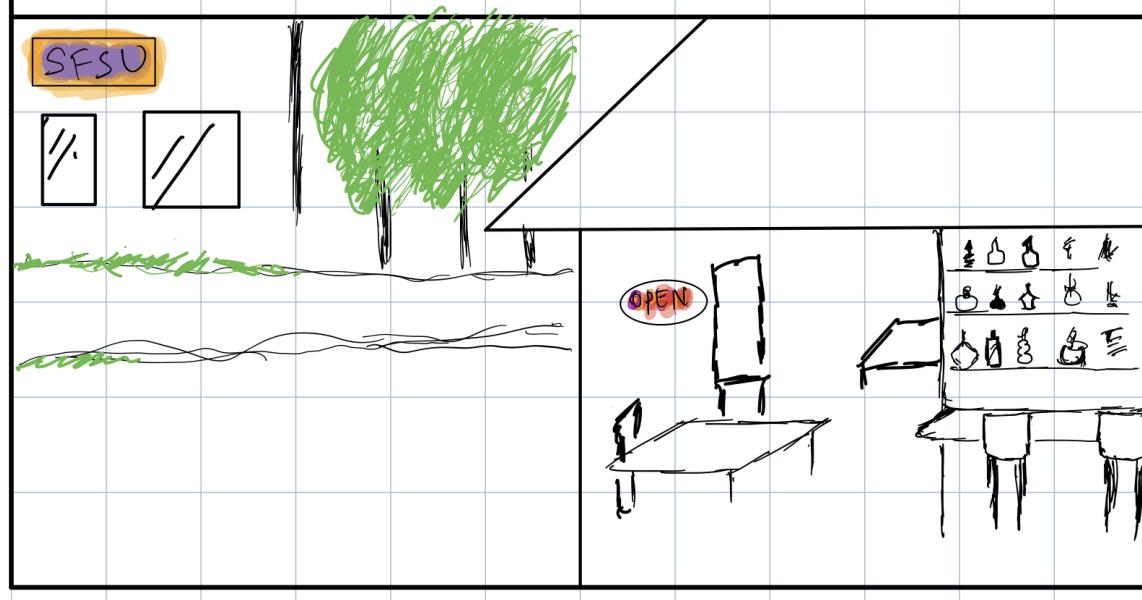


Case 6: Student at SFSU wants to write a review about a restaurant she liked

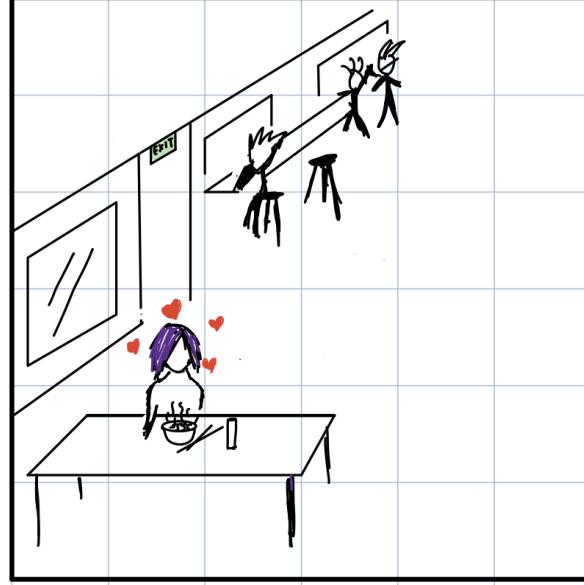
Actors: Michelle (Registered User), Alec (Super User)



A NEW RESTURANT NEAR SFSU WAS OPENED RECENTLY



Michelle, a SFSU student loved the food & thought about leaving a positive review in Gator Connection.

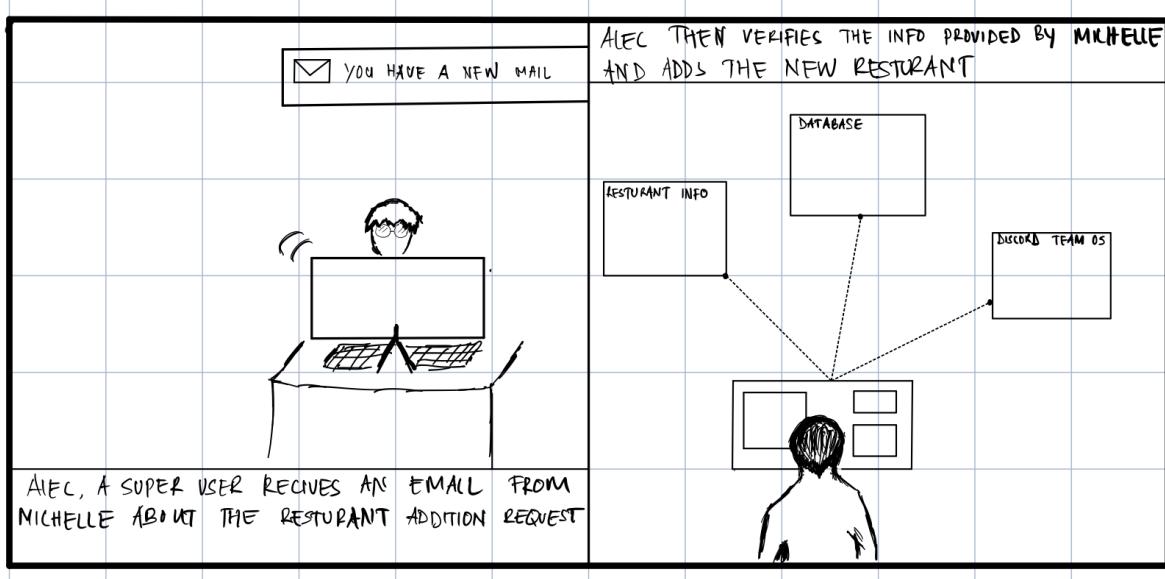


HOME SHOP REVIEWS

Wu tang...

SORRY, BUT WE COULD NOT FIND THE RESTURANT YOU ARE LOOKING FOR. IF YOU THINK THIS IS A ISSUE ON OUR END PLEASE SEND US AN EMAIL.

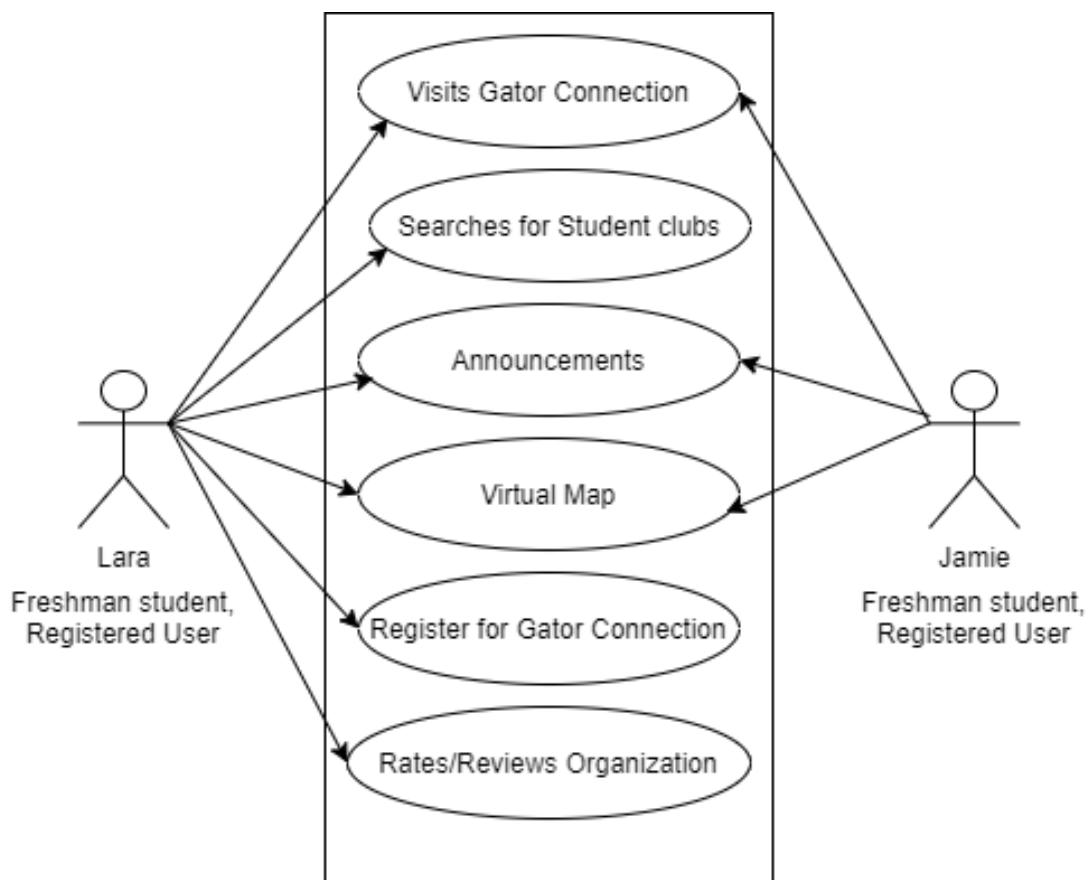
MICHELLE SENDS THE ADMIN AN EMAIL WITH THE NAME & ADDRESS OF THE RESTURANT.



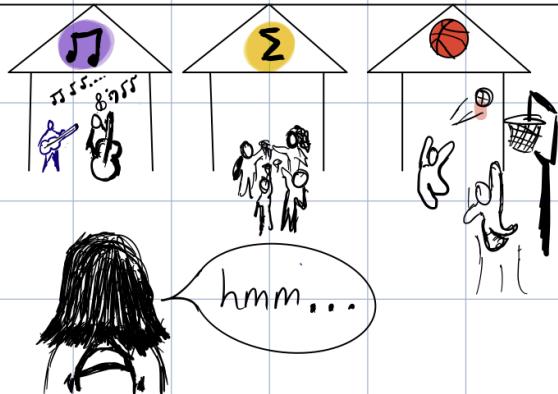
<p>MICHELLE GETS AN EMAIL SAYING HER REQUEST HAS BEEN APPROVED</p>	<p>HOME SHOP RESTAURANTS REVIEWS</p>
<p>WUTANG ★★★★☆ NO REVIEWS FOUND ADD A REVIEW</p>	<p>★★★★★ MUST TRY YUM YUM !!!</p>
<p>SHE CHECKS GATOR CONNECTION & FINDS THE NEW RESTAURANT LISTED.</p>	<p>MICHELLE SUCCESSFULLY POSTS A RAVING REVIEW WITH A PICTURE</p>

Case 7: Two Freshman at SFSU want to join a club but want more information

Actors: Lara(Registered User), Jamie(Registered User)



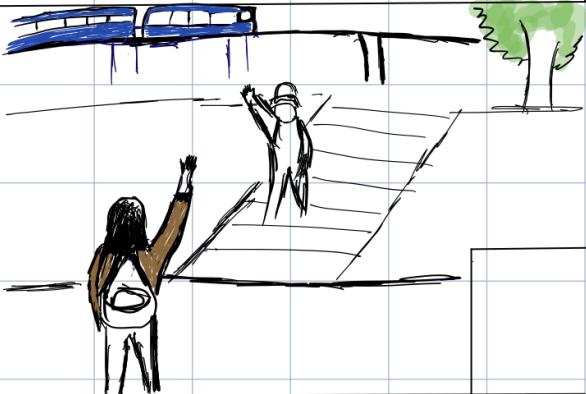
LARA, a Freshman at SFSU is deciding which club to join at the campus.



LARA Checks the GATOR CONNECTION Announcement to see if there is any event that may peak her interest.

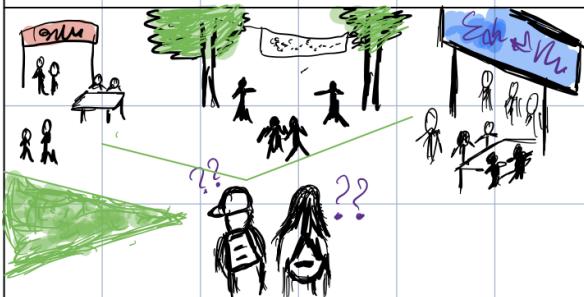
LARA MEETS HER HIGH SCHOOL FRIEND JAIME WHO IS ALSO ATTENDING SFSU

JAYME ALSO VISITS GATOR CONNECTION AND THEY BOTH REGISTER FOR AN EVENT THEY LIKE



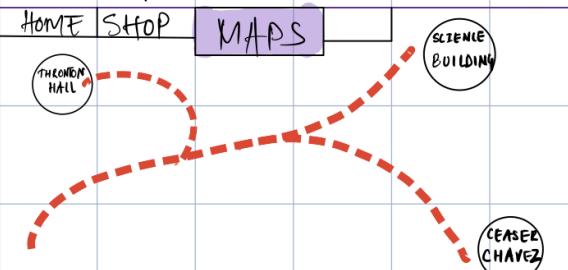
LARA tells JAIME about the events going and they both decide to join an event together

AFTER REGISTERING AND JOINING THE EVENT BOTH GET EMAIL CONFIRMATION



HOWEVER THEY DON'T KNOW WHERE THE EVENT IS HOSTED.

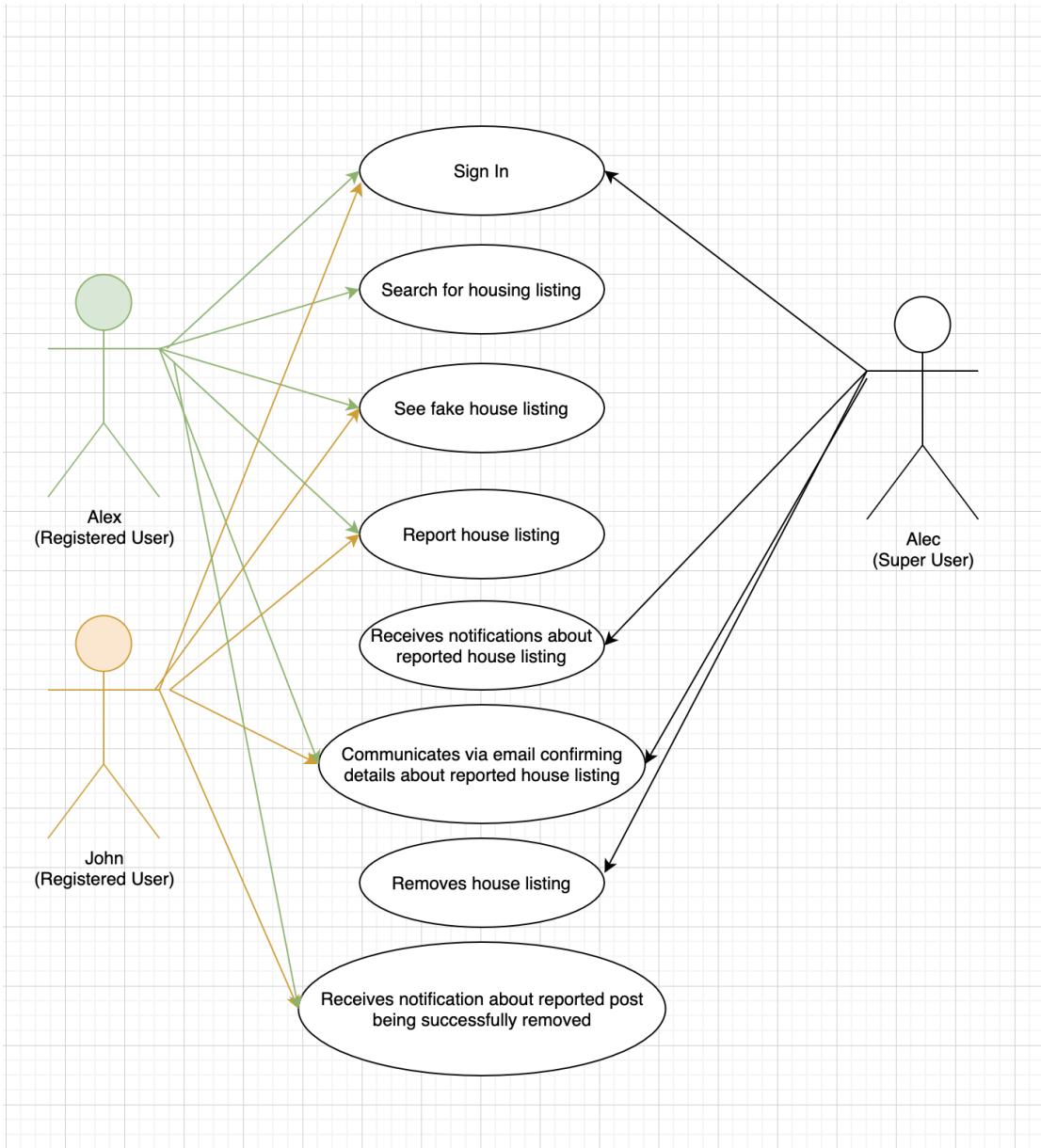
THEY GO TO GATOR CONNECTION ONE MORE TIME WHERE THEY CAN SEE THE MAP OF THE CAMPUS



BOTH SUCCESSFULLY ATTENDS THE EVENT

Case 8: A registered user reports a false housing listing

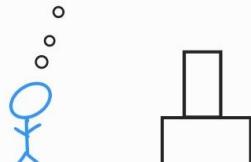
Actors: John (Registered User), Alex (Registered User), Alec (Super User)



LOG IN

ALEX

IN THE SEMESTER I
WANT TO GET A NEW
PLACE TO LIVE



LOG IN

SFSU EMAIL

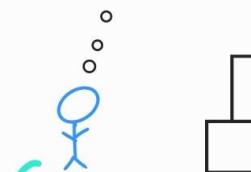
PASSWORD

LOG IN

SEARCH FOR HOUSING

ALEX

LET'S GO SEARCH
FOR HOUSING



HOUSING

SEARCH FOR

DROP DOWN BAR

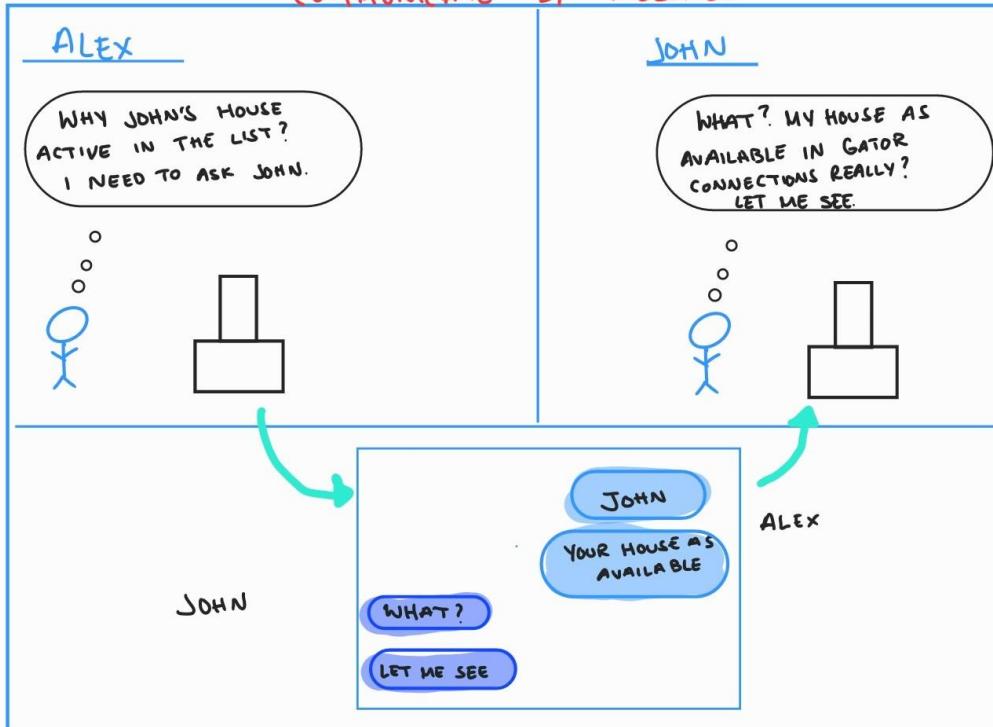
HOUSING 1

JOHN

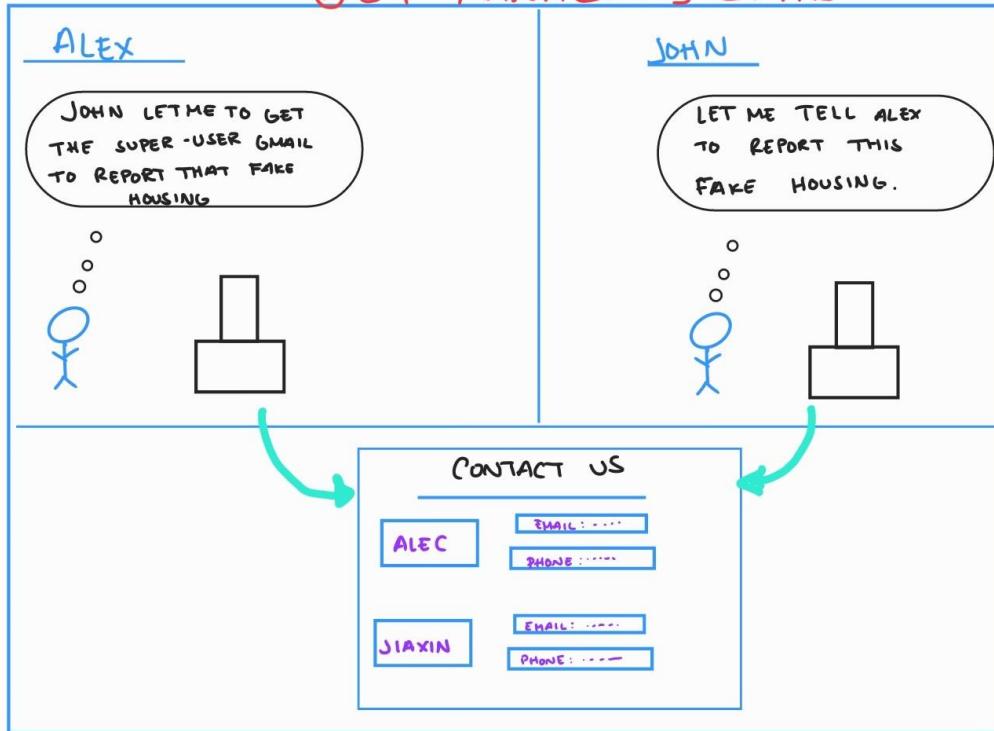
HOUSING 2

HOUSING 3

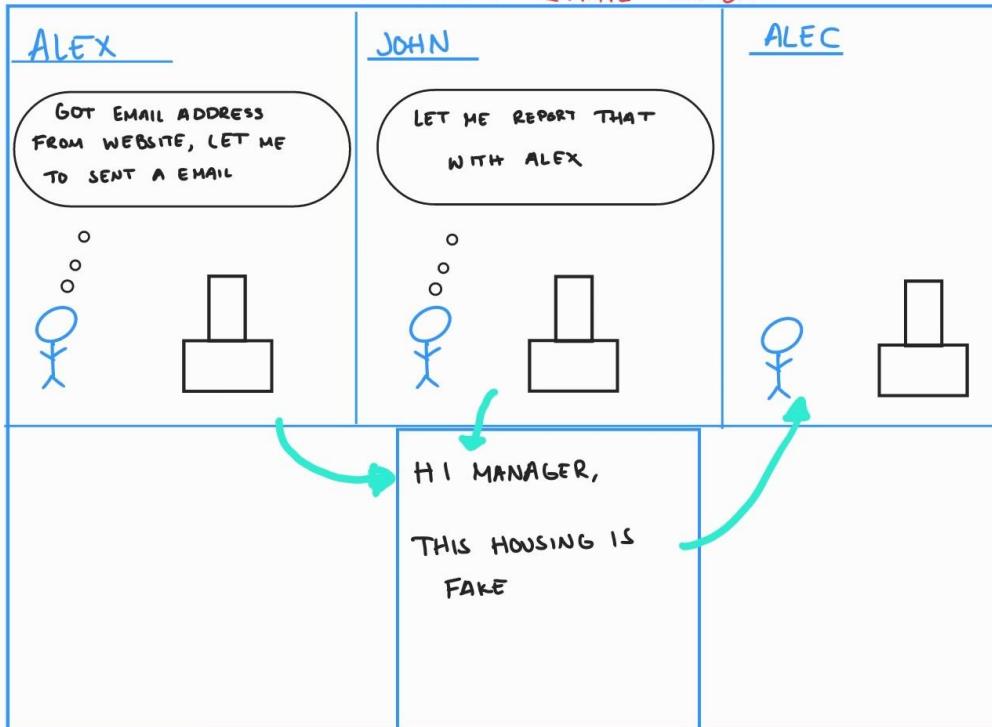
COMMUNICATE BY MESSAGE



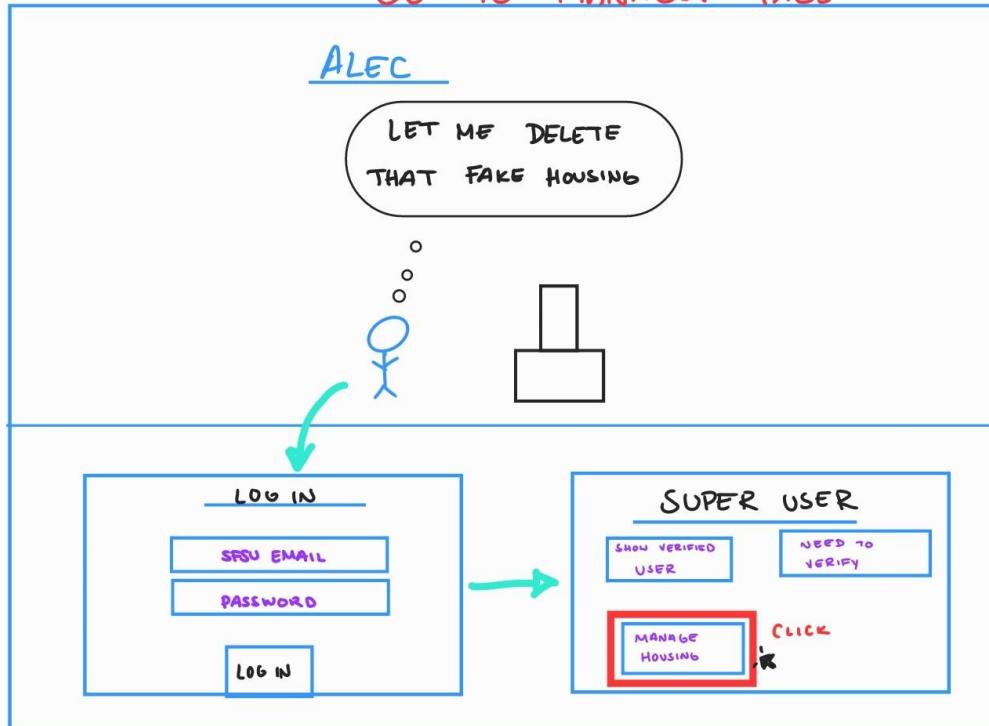
GET MANAGER'S EMAIL



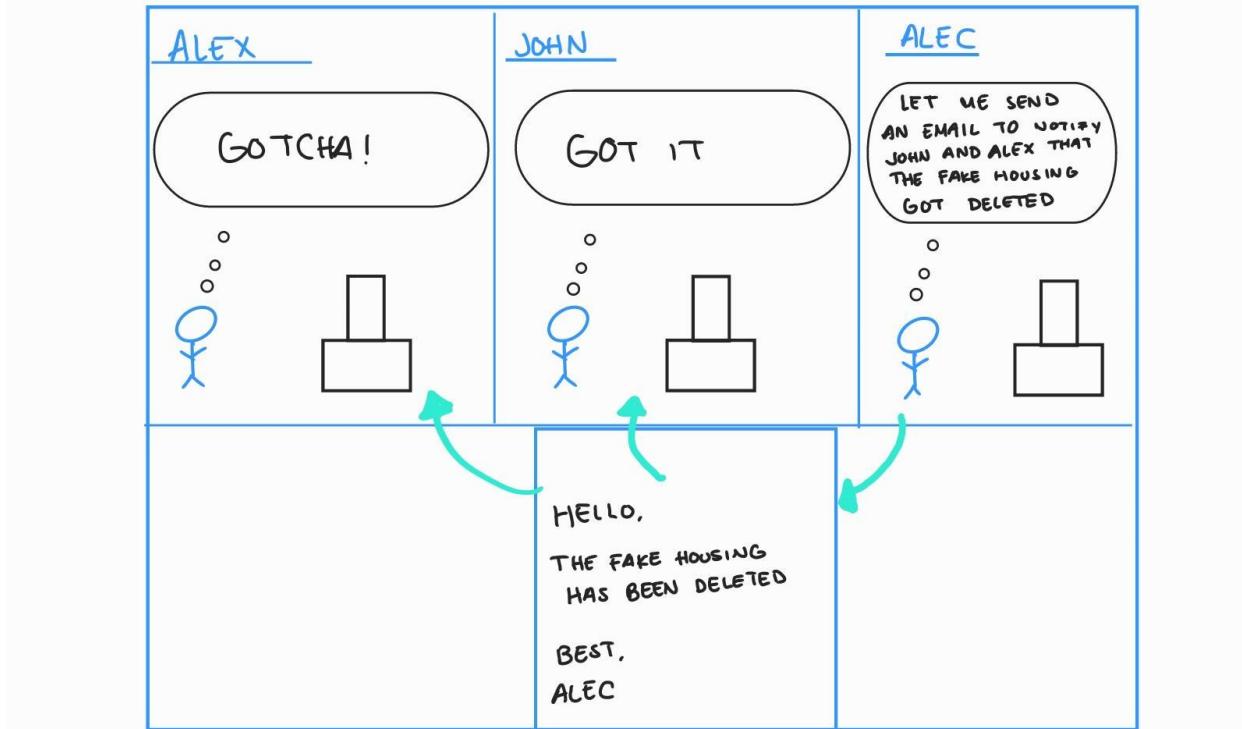
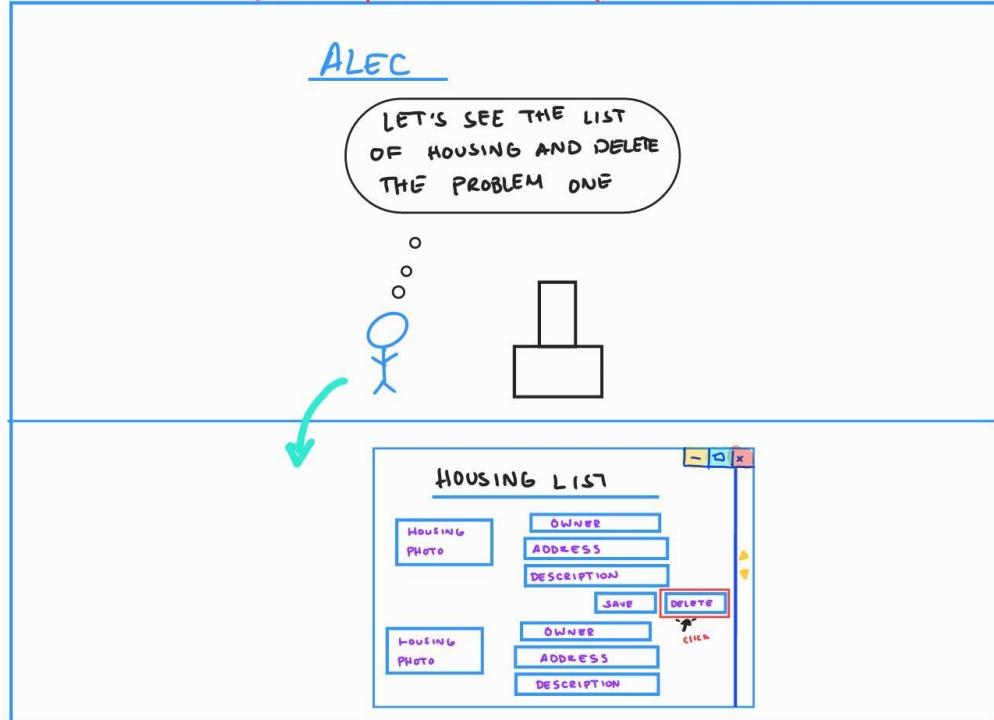
SEND EMAIL REPORT



GO TO MANAGER PAGE



CHECK HOUSING IN MANAGER PAGE



High Level Database Architecture and Organization

Business Rules

1. Registered User
 - a. A registered user shall be able to create zero or many housing listing requests.
 - b. A registered user shall have zero or many housing listing request notifications.
 - c. A registered user shall be able to fill out one housing listing form for one housing listing request.
2. Housing Listing Request
 - a. A housing listing request shall be created by one and only one registered user.
 - b. A housing listing request shall be for one housing listing post.
 - c. A housing listing request shall create one housing listing request notification.
 - d. A housing listing request shall have one housing listing form.
 - e. A housing listing request shall have a date of when the request was made.
3. Housing Listing Request Notification
 - a. A housing listing request notification shall have one housing listing form.
 - b. A housing listing request notification shall notify one and only one registered user.
 - c. A housing listing request notification shall have one and only one housing listing request.
 - d. A housing listing request notification shall have one and only one unique notification id.
 - e. A housing listing request notification shall have a date of when the notification was created.
4. Housing Listing Post
 - a. A housing listing post shall be posted by one and only one registered user.
 - b. A housing listing post shall have a unique housing id.
 - c. A housing listing post shall have a title.
 - d. A housing listing post shall have an item description.
 - e. A housing listing post shall have a price.
 - f. A housing listing post shall have many images.
 - g. A housing listing post shall be able to be removed by one and only one registered user that posted the housing listing.
 - h. A housing listing post shall be requested by one or many registered users.
 - i. A housing listing post shall have one housing listing form.

5. Housing Listing Form

- a. A housing listing form shall be for one housing listing post.
- b. A housing listing form shall be filled out by one and only one registered user.
- c. A housing listing form shall have the registered user's full name.
- d. A housing listing form shall have an about me about the registered user.
- e. A housing listing form shall have an expected day they want to move in.

6. Image

- a. An image shall belong to one and only one post.
- b. An image shall have a file path to the image.

Entity, Attribute, Relationship, Domain Descriptions

1. Super User:
 - a. A super user is a user that handles all verification of administration, organization, and athletic administration users. A super user also reviews and approves/declines posts being made for announcements, housing listings, and items for sale.
 - b. A super user's attributes are:
 - i. email: varchar
 - ii. password: varchar
2. Registered User:
 - a. A registered user is a user who has the credentials and access to be able to login and use Gator Connection's functionalities based on the type of registered user they are.
 - b. A registered user's attributes are:
 - i. reg_user_id: primary key
 - ii. sfsu_email: varchar, unique
 - i. graduation_year: int
3. Housing Listing Post
 - a. A housing listing post is the post a registered user can either view or create that holds any necessary information for the house that is for rent. Has a "HAS-A" relationship with an entity, Housing Listing Form. Also, "HAS-A" relationship with an entity, Image.
 - b. A housing listing post's attributes:
 - i. post_id: primary key
 - ii. title: varchar
 - iii. description: varchar
 - iv. price: decimal
 - v. created_at: date
4. Image
 - a. An image is associated with the post of the housing listing it belongs to. It holds the path of the image found in the project directory so that it can be displayed on the front end with its associated post.
 - b. An image's attributes:
 - i. image_id: primary key
 - ii. post_id: Housing Listing Post primary key
 - iii. image_path: varchar

5. Housing List Notification

- a. The housing list notification is an entity that alerts a user who posted a housing listing for sale a notification to show someone has expressed interest in their housing. This notification is sent out after the housing listing form has been submitted.
- b. A housing listing notification's attributes:
 - i. notification_id: primary key
 - ii. request_id: Housing Listing Request primary key
 - iii. form_id: Housing Listing Form primary key
 - iv. created_at: date

6. Housing Listing Request

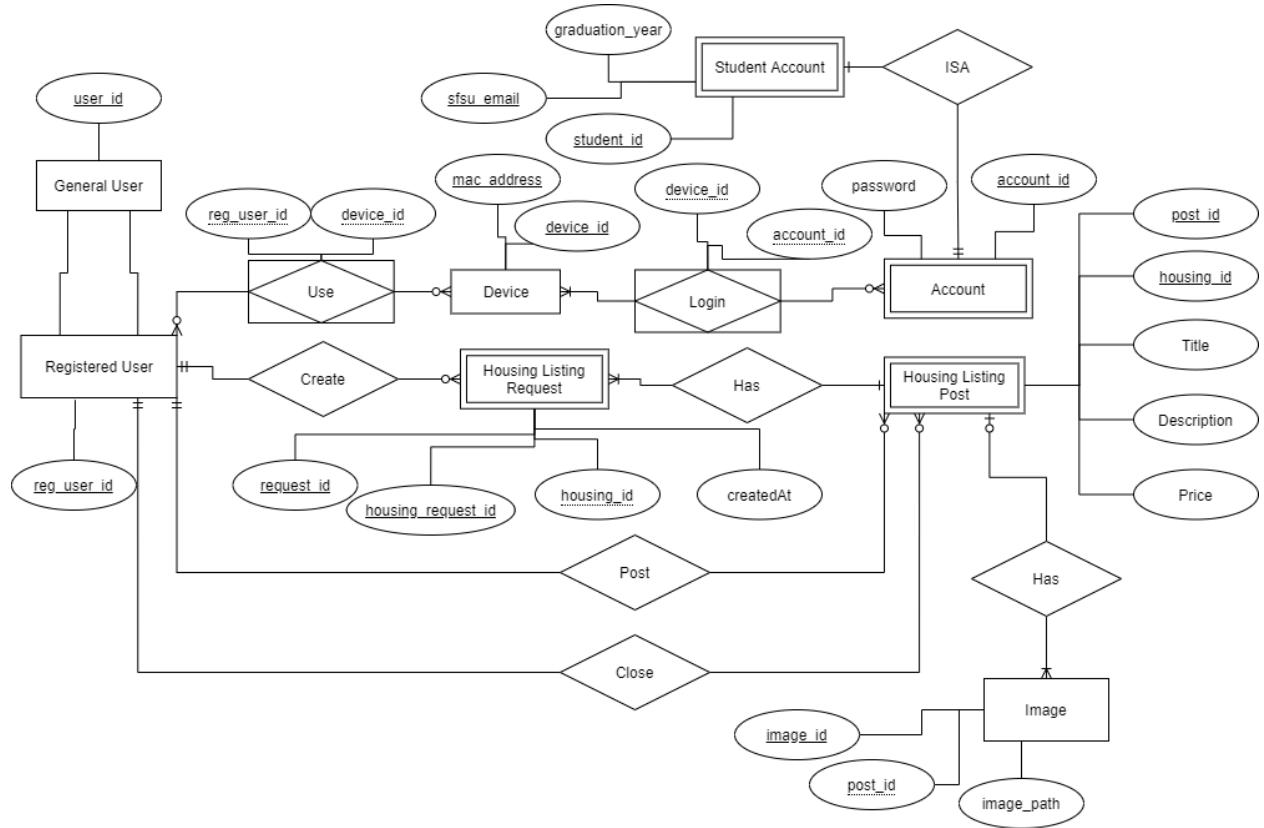
- a. The housing listing request is for the super user to approve or decline to be posted on the frontend. Once a request is made and verified, it has a “HAS-A” relationship with the Housing Listing Post.
- b. A housing listing request's attributes:
 - i. request_id: primary key
 - ii. housing_id:
 - iii. created_at: date

7. Housing Listing Form

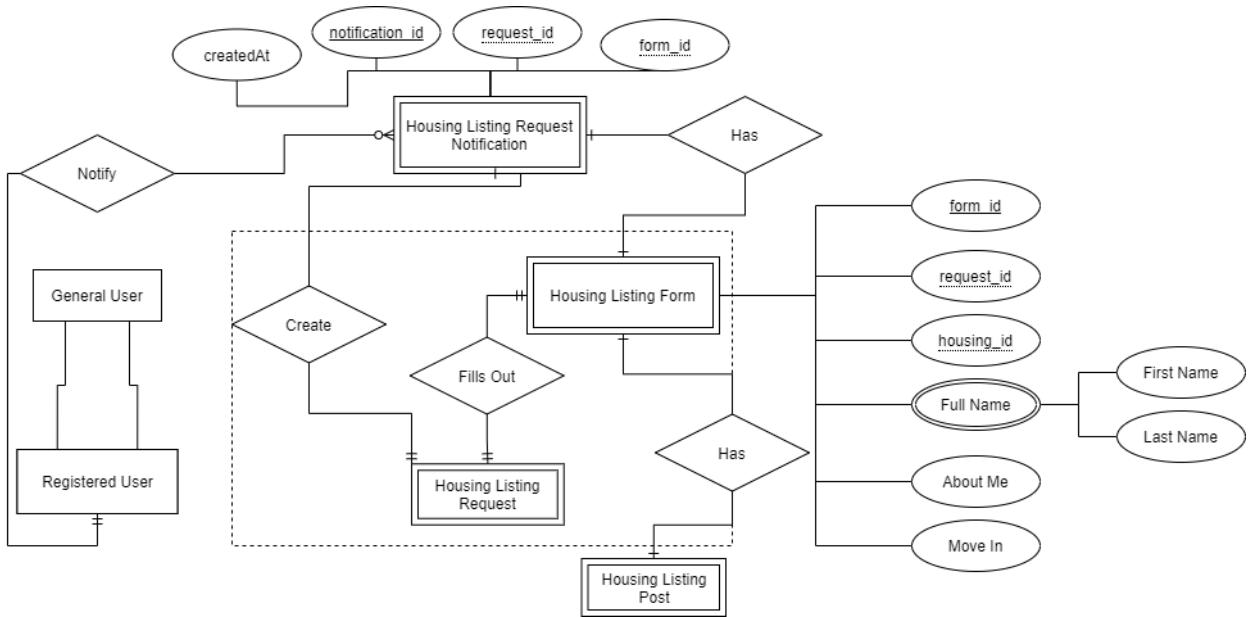
- a. A housing listing form is the form a user fills out when they find a housing listing they are interested in. The form holds all the required fields of information, and will be sent to the user who posted the listing. This entity has a “HAS-A” relationship with the entity, Housing Listing Notification.
- b. A housing listing form's attributes:
 - i. form_id: primary key
 - ii. request_id: Housing Listing Request primary key
 - iii. housing_id: Housing Listing Post primary key
 - iv. name: varchar
 - v. about_me: varchar
 - vi. move_in_date: date

Entity Relationship Diagram (ERD)

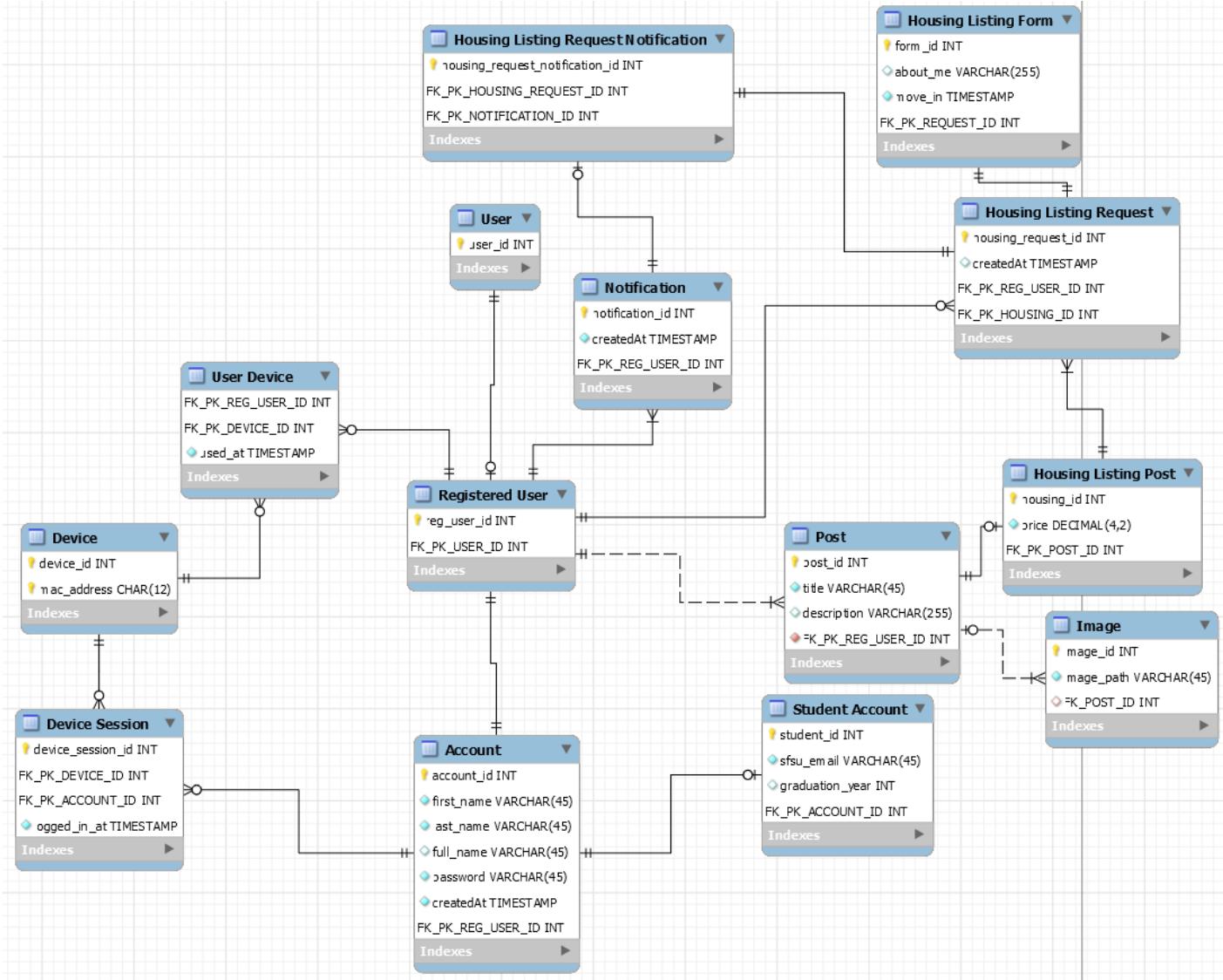
Shows login portion of ERD and relationships between Registered User, Housing Listing Request, and Housing Listing Post



Shows notification portion of ERD and relationships between notifications, Housing Listing Request, Housing Listing Form, and Housing Listing Post.



Database Model/Enhanced Entity Relationship (EER)



Database Management System (DBMS)

The DBMS that we are using to create the database is MySQL because we are creating a web application that mainly only deals in data transactions and simple queries. Another option we were considering was PostgreSQL. However, we concluded that MySQL was suitable for our application in the long run, and PostgreSQL, even though it supports more complex features, does not provide any significant advantages or extra utility for our web application needs.

Media Storage

Images will be kept in the file system in the EC2 instance because storing the images as blobs in our database does not provide any significant performance advantages. On top of this, storing the images in the EC2 instance keeps our database size small.

Search/Filter Architecture and Implementation

For our basic search functionality, we will be taking advantage of MySQL's InnoDB engine Full Text Search feature. In our database tables such as items, housing, restaurants, etc., we will create full text indexes based off of the columns in the table we want searched. Full text search allows us to search multiple columns in tables for relevant text strings due to word stemming which means our search functionality will be flexible in the results it finds. Also, we can have MySQL automatically sort search results by most relevant text-wise, which SQL's LIKE operator does not do. On top of this, based on our research, Full Text Search, on average, is faster than using SQL's LIKE operator. Therefore, our main search implementation method will be using MySQL's InnoDB engine Full Text Search. However, we will still look into SQL's LIKE operator and compare the performance and results of individual queries. Therefore, some sections of our search implementation will be decided on a case to case basis. We will create full text indexes for the combinations of table columns that we need. These full text indexes also have the added benefit of taking care of most of our filtering.

High Level APIs and Main Algorithms

API Endpoints

POST Calls:

/login_user

Using this post call will authenticate the user credentials and log them in.

/student_registration

Using this post call will register users in the mysql database.

/password_reset

Using this post call will reset the password users have for their account.

/logout_url

Using this post call will log the user out.

/item_upload

Using this post call will allow users to create an item for sale post.

/item_edit

Using this post call will allow users to edit an item post they have created.

/item_delete

Using this post call will allow users to delete an item post they have created.

/housing_upload

Using this post call will allow users to create a housing for sale post.

/housing_edit

Using this post call will allow users to edit a housing post they have created.

/housing_delete

Using this post call will allow users to delete a housing post they have created.

/housing_submit_form

Using this post call will send an email to the user who posted the listing, informing them of people who are interested in the housing.

/announcement_upload

Using this post call will allow users to create an announcement post.

/announcement_edit

Using this post call will allow users to delete an announcement post they have created.

/announcement_delete

Using this post call will allow users to delete an announcement post they have created.

/restaurant_upload

Using this post call will allow users to submit a restaurant for consideration to the restaurants section.

/restaurant_review_upload

Using this post call will allow users to create a restaurant review.

/restaurant_review_edit

Using this post call will allow users to edit a restaurant review they have created.

/restaurant_review_delete

Using this post call will allow users to delete a restaurant review they have created.

GET Calls:

/item_return_all

Using this get call will allow users to get all item posts.

/item_return_specific

Using this get call will allow users to get a specific number of item posts based on their search query.

/housing_return_all

Using this get call will allow users to get all housing posts.

`/housing_return_specific`

Using this get call will allow users to get a specific number of housing posts based on their search query.

`/announcement_return_all`

Using this get call will allow users to get all announcement posts.

`/announcement_return_specific`

Using this get call will allow users to get a specific number of announcements based on their search query.

`/restaurant_return_all`

Using this get call will allow users to get all restaurants.

`/restaurant_return_specific`

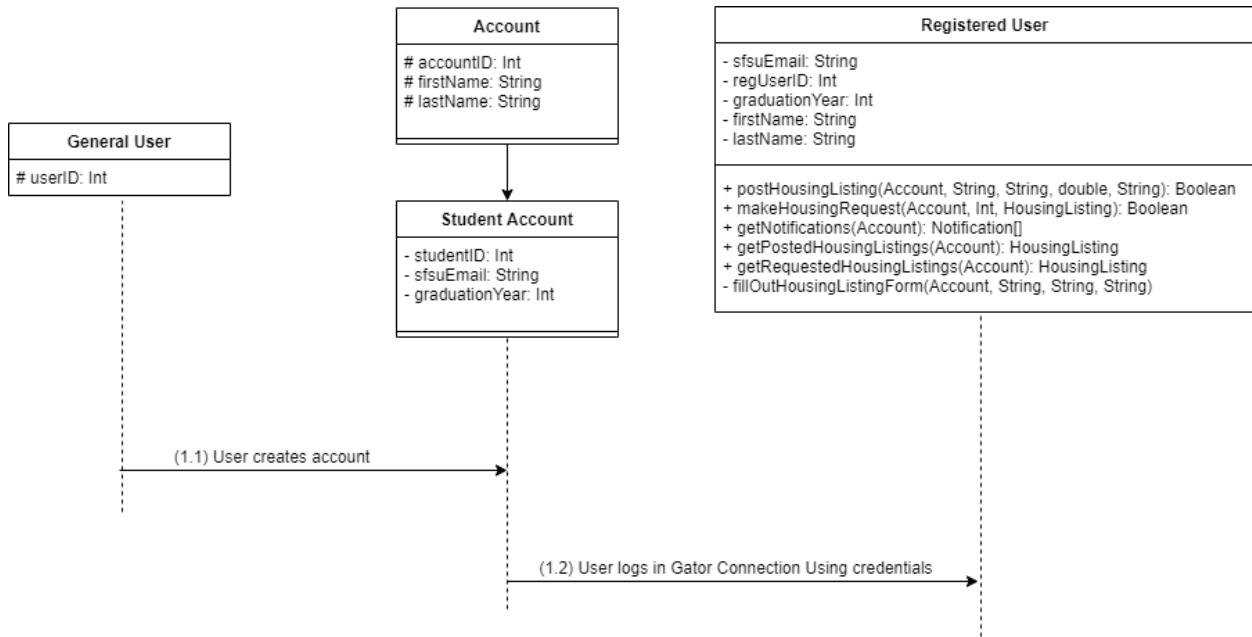
Using this get call will allow users to get a specific number of restaurants based on their search query.

`/restaurant_review_return_all`

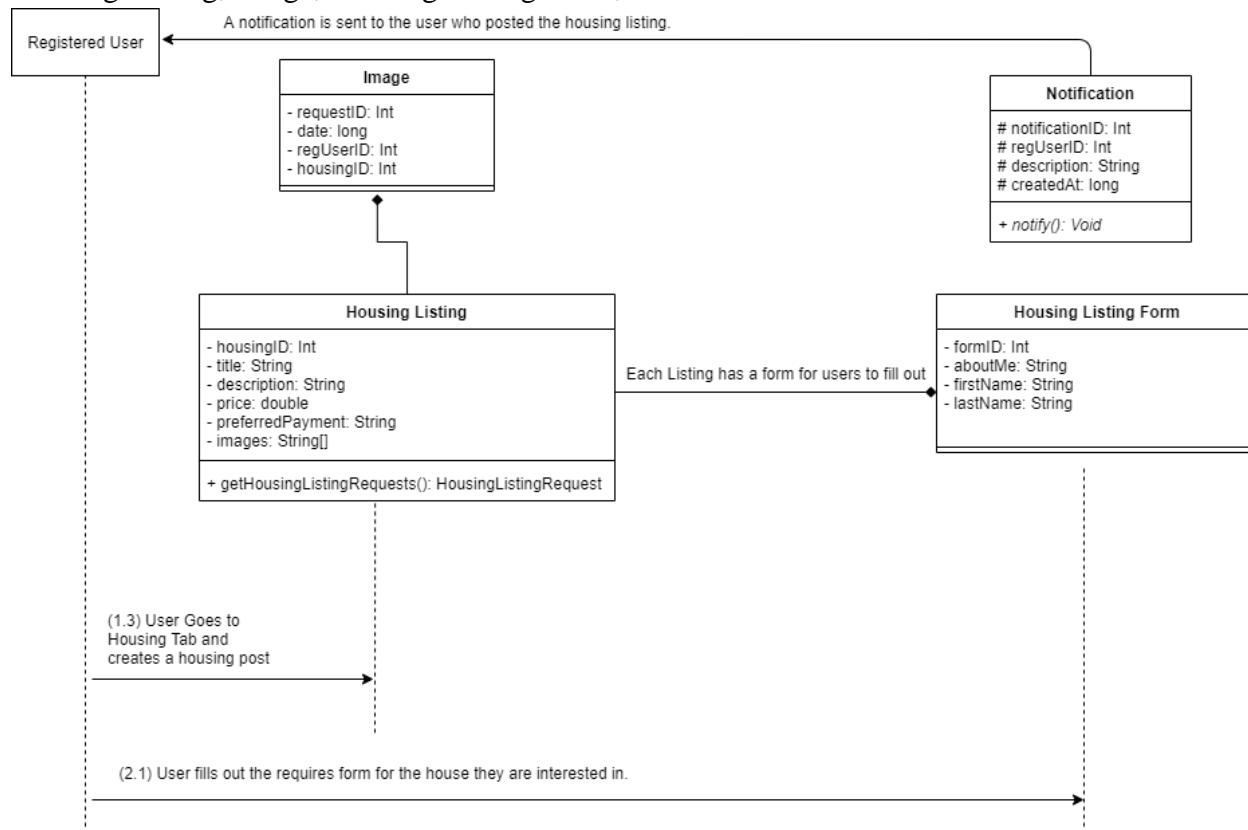
Using this get call will allow users to get all restaurant reviews for that restaurant.

High Level UML Diagrams

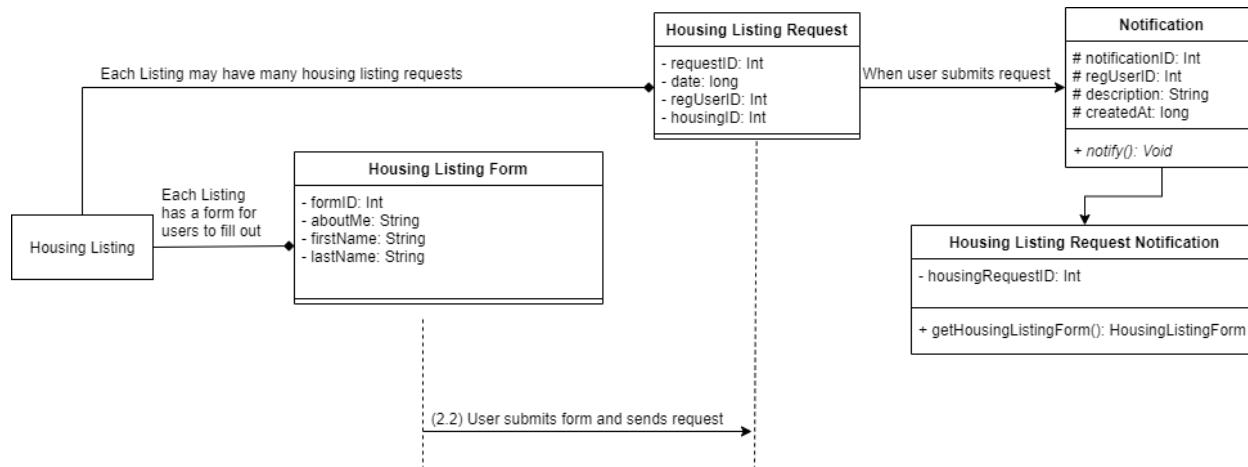
Association and Relationships between General User, Registered User, Account, and Student Account.



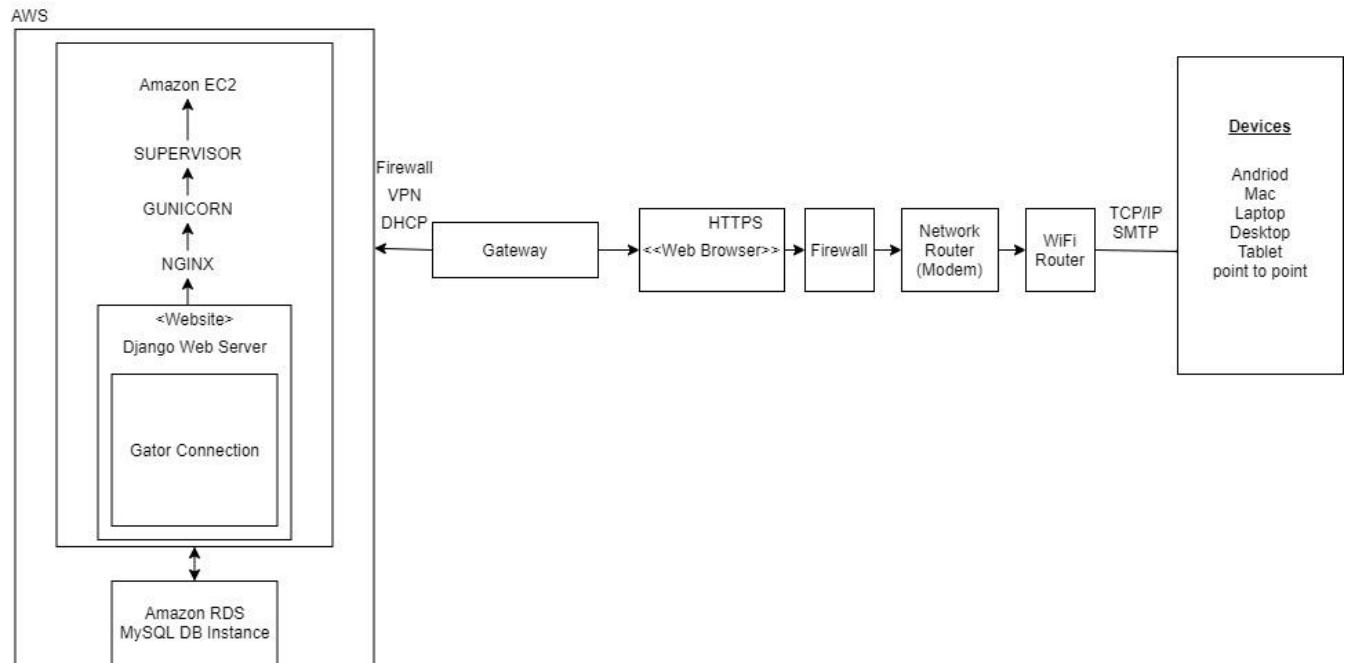
Associations and Relationships between Registered User, which was already defined above, Housing Listing, Image, Housing Listing Form, and Notification.



Associations and relationships between Housing Listing, which was already defined above, Housing Listing Form, Housing Listing Request, Notification, and Housing Listing Request Notification.



High Level Application Network and Deployment Diagrams



Project Risks

Schedule risk

Due to everyone's schedules, it's hard to get a meeting time for everyone to be present. However, with having a trello account set up for project management for all team members, it has been easier to convey task distribution. Furthermore, we have been able to meet by separating our meetings into front-end and back-end meetings, and recapping the team members who can not attend.

Skill risk

Our team has development experience, however, none of us had experience designing a website. Furthermore, while some of us were familiar with the django web framework, it was unfamiliar to most of the team, with most of us having to learn it on the fly. In addition, Python is also unfamiliar to some of the team, so some of us are still in the process of understanding it. Lastly, most of us have never done front end design at this large of a scale, so everyone on the team is learning it as we go along as well.

Technical risk

One technical risk was associated with our github repo. What was at risk was our git branches, specifically with push/pull and merge conflicts. How we keep track of merge conflicts is with the multiple branches we have for different tasks and having a Github master who handles the merge conflicts. Another technical risk was the issue of our MySQL database because we wanted the team to connect to the database locally so we would all work with the same data. Connecting to the database was challenging for the team, but we had managed to all do it. However, one thing was that we had to do was make our database public, which could expose information. However, we fixed this problem by creating security rules that allowed only the team's and EC2 instance's IP addresses to access the database. One more technical risk was how everyone has different operating systems, such as Windows, Mac, etc. In the end, we decided to use Ubuntu for our project, which is generally easier for people with Apple products. For the team members who had a Windows Operating system, they had to learn Linux commands to get used to the environment for our project.

Teamwork risk

One major teamwork risk is having good communication between the backend and frontend teams due to the team's scheduling issues. However, with our project management being handled by Trello and the team lead posting updates on the discord channel and attending every meeting to be the middleman between the front and backend, everyone is up to date on what needs to be done and what each team is working on.

Legal/Content risk

Our legal risks may vary depending on the end product of the deployment. On our server, we will have control and ownership of images on the site. With that, one risk associated with us is that we may encounter people who upload images that do not belong to them. Another risk is that we will have external links to outside resources for services we do not provide. This means that if anything happens to the user after he/she clicks on an external link, we might be held responsible. To resolve these risks, when a user registers their account, they must agree to a terms and conditions form. These terms and conditions would ensure that everything they upload belongs to them and that the user understands that we are not responsible for anything that occurs after the user exits our site via any external links. We will also warn users when they click on an external link that they are leaving our site.

Project Management

In our first couple of team meetings for M2, we addressed the sections that we believed had to be done with everyone together such as setting the priorities of our functional requirements, enhancing our data definitions, and examples of how we wanted to format our UI mockups and storyboards. Afterwards, our next team meeting, we set up Trello as our main project management tool with tasks split up into frontend tasks, backend tasks, team tasks, and completed tasks. By keeping track with Trello, backend and frontend team members would create rough drafts of the sections assigned to them. Once those team members completed the rough drafts and notified the team on Trello and Discord, either the frontend lead or backend lead would give feedback depending on which team was in charge of which section and if needed, set up meetings with the team members that created the rough drafts and edited the sections together. Otherwise, the team members would incorporate the feedback and continue the cycle of getting feedback from a frontend lead or backend lead. Once the frontend lead or backend lead deemed the section was completed, the team lead did the final quality check. The team lead also made sure to check in and help with all sections that team members were working on and gave intermittent feedback as well. For future tasks, we will continue to use Trello and Discord to keep track of tasks and our progress. Most importantly, we will continue to continue the cycle we were practicing of creating rough drafts, receiving feedback from a team lead member, and incorporating the feedback.

Gator Connection Tasks

Board | Star | Team visible | LC AR AY AT BK +2 | Invite

Milestone 2 Tasks

- Create Database model
- Finish m2 doc (Mar 30)
- Search method, filter student's email.

+ Add another card

Team Lead Tasks

+ Add a card

Front-end Tasks

- Shop-Items(development branch) (Apr 6)
- Home Page updated (Mar 25)
- Create less whitespace bw picture and welcome to gator connection string
- Add carasoul under mission statement
- Have searching underneath the carasoul
- Updated Home Page (Mar 26)

Back-end Tasks

- Send email to verification (AY)
- Display stored images on items page. (BK)
- Create tables for signup, login, material post. (Mar 14)
- Create DB models on MySQL workbench. (AR)
- Send email to reset password (AR)
- Verify only SFSU emails are registered. (AR)
- Start on MySQL tables (Due TBA)
- Function if user tries to find all.

+ Add another card

Everyone Tasks

+ Add a card

COMPLETED TASKS

- Create new Django app folder: gator_connection for new path. (Mar 6)
- Create ERD by Monday (Mar 8)
- Create html for login for signup (Mar 16)
- NavBar (Mar 18) 1
- Home Page (Mar 18)
- EnableHttps for main instance
- New version of ERD (Mar 13)
- Work on search bar function on home page. M2 branch (Mar 21)
- API Calls

+ Add another card

List of Contributions

1) Data Definitions V2:

During a team meeting, the team looked at the data definitions we had, and decided on which terms we needed to expand upon. After deciding, Alec added the extra terms to explain certain subsections, with the team revising what should be said to explain them.

2) Functional Requirements V2:

During a team meeting, the team looked at the functional requirements we had. During this time, with suggestions from other members, namely Angelo and Benjamin, we were able group up our functional requirements into the specified three priority groups, with Priority 1 being the most important, to Priority 3 being opportunistic.

3) UI Mockups and StoryBoards:

During a team meeting, Alec tasked the group to do a rough draft of a UI mockup of the Use Case they had made. After receiving the rough draft, Alec looked over the styles everyone had submitted and decided to have the Front End members, namely Carmen and Bikram to revise everyone's rough draft into a final draft version.

4) High level database architecture and organization:

During a team meeting, Angelo started the rough draft of the entity relationship diagram (ERD) for our special feature. After feedback from Jiaxin and Benjamin, the ERD was revised to be more accurate of the special feature we had chosen for our project and to follow more closely along the lines of the business rules. From the business rules, Angelo was able to extract the entities and describe them at a deeper level. On the database model, Benjamin designed the database model that we would be using for the special feature based off of the ERD. On this part, we also decided that we would use MySQL for our Database Management System(DBMS) and our images would be kept in the file systems of our EC2 instance, with Benjamin filling this part out. Furthermore, on another team meeting, Benjamin showed the team the Full Text Search feature, which we would decide to use for our search/filter architecture.

5) High Level APIs and Main Algorithms:

Alec listed down certain API's that the team was going to eventually use for the project, such as POST calls for logging in and registering, GET calls for getting user information, etc. After feedback from the team, the API's were more fleshed out to describe the functionality.

6) High Level UML Diagrams:

Benjamin was tasked to complete the UML diagram for the special feature. After a meeting and some feedback for the first version from Angelo, Benjamin was able to revise the UML diagram to fit the standards of our special feature.

7) High Level Application Network and Deployment Diagrams:

Alec started a rough draft of the Application Network diagram. After a class lecture, the team, most notably Benjamin, helped revise the Application Network diagram to portray more info of what the professor had explained about the diagram. After clarification from the professor, Alec found out that the High Level Application Network and Deployment Diagrams could be combined into one, and revised it as such.

8) Identify actual key risks for your project at this time:

During a team meeting, the team helped suggest things that were considered as risks for the project. After jotting them down, Alec organized them into 5 sections, which were schedule, skill, technical, teamwork, and legal/content risks.

9) Project management:

During a team meeting, Angelo set up Trello and invited everyone on the team so we could start assigning tasks. On Trello, Alec delegated tasks to the team, such as having UI mockups done by a certain date.

10) Detailed list of contributions:

Alec filled out this part. After assigning the tasks to everyone, Alec looked over what everyone did to accurately fill out this section.

11) Milestone2 Editor:

After volunteering to edit the document, Lakshita organized the Milestone2 document to make it ready to be turned in.

Vertical Prototype Contributions

Backend:

For the backend portion of our vertical prototype, Angelo helped set up basic API features, such as registering and logging in for users. After a revision from Benjamin for the login portion, the registering and logging in features was complete. After the team learned how to use the login feature, which would allow us to restrict certain actions from someone if they were not logged in, Jiaxin was tasked to start implementing the search function. With that, Angelo expanded upon the search feature which would then allow users to search for a user based by their email. Alec then helped on the searching feature, so users would then be able to filter by sfsu email, and by the first or last name associated with that email. Benjamin then improved upon Alec's filtering feature by having the feature use Full Text Search to improve upon the searching feature. After improving upon the filtering feature, Benjamin then reorganized our backend code structure to make everything easier to find. Angelo then set up the DB table "student" on MySQL workbench. When a user would register, their info would be stored there. Furthermore, on the registering and logging in functions, Alec added error handling to make sure that the program would continue in case of an error.

Frontend:

The front end team was in charge of designing the vertical requirement home page. Lakshita, Carmen, and Bikram added features such as having a mission statement at the top of the screen, and a carousel slide at the bottom to show SFSU pictures. Other features implemented were having certain features only appear if a user is logged in, having messages appear to notify a user that once they have completed a certain action, such as logging in or registering. Furthermore, these messages have replies for actions that fail or succeed. In the case of an action succeeding, they will be greeted with a green box saying something like "Thank you for registering". On the flip side, if they fail, they will be greeted with something like "Please try again". Another feature added was structuring how the data coming from the database would look like on the page. On the navbar features, Alec designed the rough prototype, then had Bikram implement features on the navbar such as the navbar would highlight a specific section when it was on that page. Furthermore, Carmen designed our logo which would show on the top left of the page. On the design of the login and registering boxes, Alec designed the rough prototype for them, and after feedback from the team, he was able to revise it to make it look much cleaner than it was before.