Wi-Find

Spring 2024

Group 9: Walid Abdullahi, Kevin Morales Funes, Kelsi Hill, Lisa Nguyen, Thomas Sigler

February 14th, 2024

Section 1: Planning, Schedule, and Peer Evaluation

Name	Email	Task	Duration (Hours)	Dependency	Due Date	Evaluation
Walid Abdullahi	Wabdullahi0529@gmail.com	Use Cases 13-15 and Requirements 13-15, Revision.	2	N/A	2/13	100%
Kevin Morales Funes	Kevingustavo.kmf@gmail.com	Use cases 7-9, requirement cases 7-9, Reviewed previous diagrams and problem statement, Wi-Fi Use Case Diagram	3	N/A	2/13	100%
Kelsi Hill	2016k.hill@gmail.com	Updated Context Diagram, Use Case Diagram User Registration, Use Cases 10-12 and Requirements 10-12	5	N/A	2/13	100%
Lisa Nguyen	Nguy3n.lisa@gmail.com	Database Management Plan/Diagram /Choice, Use Cases 4-6 and Requirements 4-6, Modified Problem Statement	8	N/A	2/13	100%
Thomas Sigler	ts309435@gmail.com	Document Setup and Outline, Use	6	N/A	2/13	100%

		Cases 1-3 and Requirements 1-3, Github Setup and Screenshot, Revised Activity Diagram, Use Case Diagrams Ticket Submission and Payment Processing				

Section 2: Revised Problem Statement

Our product, at a high level, is a platform that connects users who want to rent out their internet connection (renters) to users who want to temporarily rent internet connections (rentees); the product is an e-commerce platform for a specific subset of products.

Those who want secure Wi-Fi on-the-go but do not need constant access to Wi-Fi will benefit from this product. Furthermore, users who have specific Wi-Fi criterias or need internet access but only have extremely limited and exorbitant viable options will find this product desirable. On the renter's side, this product will open a tangible opportunity to those who want to make extra money without breaking a sweat via renting out their Wi-Fi to others.

Our product presents a possible solution that is both elegant and profitable to the issue of free public Wi-Fi which is often unsecured as well as the issue of obtaining individualized Wi-Fi needs at an affordable rate.

Obvious alternatives are currently available that are viable competitors to our product. Many restaurants, cafes, and hotels now offer free Wi-Fi to their patrons; these competitors however, often do not enforce secure Wi-Fi. Internet Service Providers (ISPs) are the next upgrade from free public wifi and offers permanent Wi-Fi for a monthly fee; they are the largest competitor to our product but do have drawbacks. ISPs are far more expensive than what we expect our product will provide to consumers. They often have fine print contracts that binds users who may desire pay-as-you-use. Furthermore, ISPs availability varies by location, so a user's power to choose a rate and quality that suits them is limited if available ISPs in the area are those with pricey plans. For example, 5G with certain providers are not possible if their 5G tower is not in the area. Fiber also suffers from the same issue of physical limitations (no fiber underground in areas where it is impossible to dig them). DSL (like NetZero) is much more affordable for slow internet but is not available in areas with no telephone poles. Breaking past ISPs that are limited physically or constructionally would be satellite internet such as ViaSat or StarLink, both exorbitant alternatives that each have their respective drawbacks. ViaSat is expensive and only as fast as DSL while StarLink's internet speed is sensitive to obstructions and vary depending on location. The final possible alternative is to commute to an internet cafe or establishments similar to it which could cost much more in both time and money just for commuting.

This project is compelling and worthy of developing since it could present a viable and competitive alternative to using free, unsafe public Wi-Fi, committing to an exorbitant internet service plan or trading invaluable time to get to an internet access point. Furthermore, our product could enable people who are traveling or are far from home to feel comfortable using a Wi-Fi connection for confidential and important actions over the Internet.

The top level objective is that our product shall provide a hub or centerpoint where clients can list their Wi-Fi up for rent, and other clients can visit this hub to rent Wi-Fi that suits their requirements. The means of creating this hub will be via a website. The website shall have geolocation functionality so renters can show where their Wi-Fi is and rentees can find available Wi-Fi based on location. The website should have a secure way for monetary transaction systems. There shall be a notification system for status and reminders for both renters and rentees.

What makes our product different compared to what is currently on the market is that our product conglomerates many options with varying characteristics for accessing internet service into one location so that clients who need quick, temporary Wi-Fi are able to gain access with ease. This in turn opens profit opportunities for clients, who are not an internet service provider company but have internet service, to temporarily rent out Wi-Fi that the client is not using to its maximum capacity.

The target customers are for those seeking to make profits as in clients who want to rent their internet service at a particular location temporarily. Our target customers are also for those seeking Wi-Fi as in c ents who want an alternative form of internet service temporarily to fulfill their current individualized need such as higher security, fast upload speed, uninterrupted internet access, fast download speed, high user capacity amount, etc. There could be many creative individualized needs. Our constraint for who can use our product is that all clients must be 18 years or older.

The scope of our product is to enable greater Wi-Fi quality that an individual is seeking given what he or she is willing to pay; furthermore, we envision that our product will contribute another diverse option in the pool of profit opportunities; clients who want to rent out their Wi-Fi easily can do so which in turn also increases diversity for individuals seeking Wi-Fi. The clients seeking internet access will have greater options. For clarification, the product is not selling Wi-Fi to clients; the product is merely a platform clients can use to do their business.

Our perceived competitors for clients seeking Wi-Fi are the existing ways of gaining internet access--hotspot, 5G, Satellite, Fiber, free public Wi-Fi (Starbucks, libraries, universities, supermarkets, offices). The perceived competitors for clients seeking to rent out Wi-Fi for extra money are possibly Craigslist listings or sites similar to it and face-to-face contact between renter and rentee.

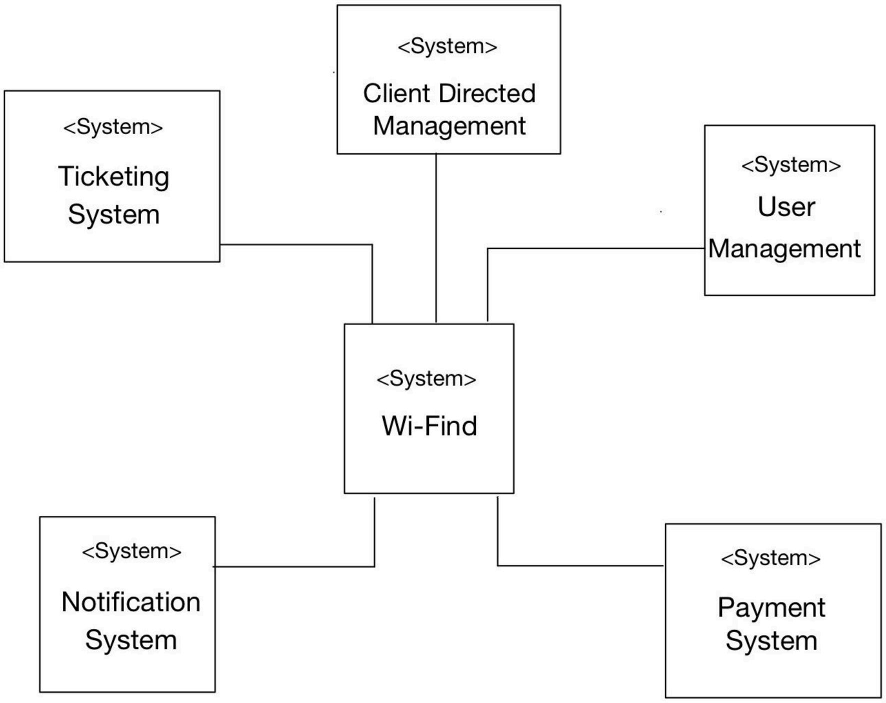
What is novel about our approach is that our hub allows both types of clients to get their needs in ease and break past their current limitations. For clients wishing to rent Wi-Fi, they may be limited physically since not all locations (such as new development areas or remote regions) have structures that enable internet service by a particular provider. For example, fiber, one of the best choices for high speed internet, is not available in every location especially if houses are far spread as laying underground fiber lines is costly. Coverage of Hotspot services such as T-Mobile have dead zones where no service is available or service is so poor that nothing loads. Public Wi-Fi has limitations beyond the monetary cost including cost of time to commute required to the destination, unmeasurable cost from speed and security limitations and cost of planning to accommodate limited time for access like business hours only. Clients may also obtain more buying power and avoid getting stuck in an internet service provider's limited plans. For clients wishing to rent out their Wi-Fi, the hub enables them to make profit by conglomerating the demand pool to just one location therefore increasing their number of possible clientele and formalizing transactions between renter and rentee better.

The system for our product can be built using an existing web stack that best matches the project's requirements and lifecycle. Existing webstacks include technologies and software that are compatible between frontend, backend, database, etc. The Google Maps API has geolocation functionality as well as libraries for calculation of points and area. Using this will allow renters to pin where their Wi-Fi location is and have the range calculated and shown on the map. Rentees can view rentable Wi-Fi's near them or around an address or coordinate they enter. Popular enterprise databases such as postgreSQL have easy-to-use encryption for data management and server storage for protecting sensitive user information such as passwords, purchases and cash exchange information. For payments, a third party established platform such as Stripe can be used as well. Reliable open source application security tools can be used to help make the hub more secure against malicious attacks.

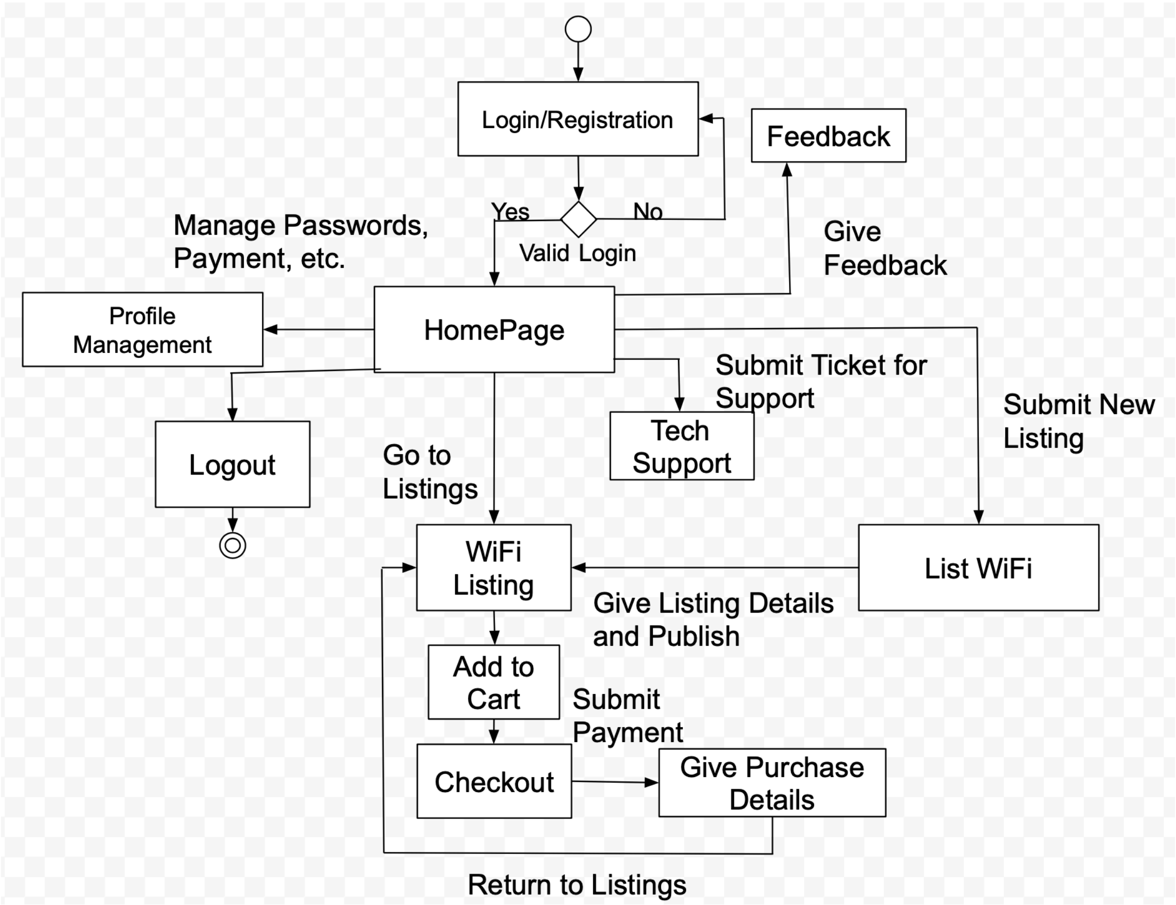
From a technical point of view, implementing appropriate validations for Wi-Fi, transactions, identity, etc may be tricky thus reaching an acceptable solution will be interesting.Another interesting point is figuring out a strategy and implementation to maintain an acceptable update that reflects to all users viewing the site.

There will be a client login and an admin login. The client login will consist of features pertaining to renting and renting out Wi-Fi. Clients should be able to do both if they wanted to (such as rent out their home Wi-Fi while they are on vacation in an area that has poor service leading them to simultaneously rent Wi-Fi from someone else in that area). A separate admin login will be used by the website developers, administrators and moderators. Depending on their role and permission level, actions and information will be limited to just what is necessary for completing the task or maintaining the site.

Section 3: Revised Context Diagram



Section 4: Revised Activity Diagram



Section 5: System Requirements:

- 5.1: Use Cases

  Use Case Number: 1

  - Use Case Name: User Login
  - Actors: User, Administrator, Database
  - Description:
    - On connection to the site, the user will be given a login page.
    - The user will enter an email and password(includes).
    - Recommended option is an excludes option
    - After the inputs are complete, the user will have to hit submit
    - If successful, the user will end up on the destination page
    - If unsuccessful, the user will be given a try again message
    - The information will be validated with the database
    - Administrators can change the look and feel of the website and anything in the database, such as email and password.
  - Alternate Path: There is only 1 path to the destination site, unless given the error to try again.
  - Exception Path: If the user doesn't have an account, they will be prompted to sign up.
  - Pre-Condition: Valid account must be made.

- Post-Condition: Reaching the destination page through the browser after the pre-condition has been met.

Use Case Number: 2

- Use Case Name: Payment Processing
- Actors: User, Administrators, Database, Third Party Companies
- Description:
    - On clicking checkout, User will be sent to a page asking for payment information
    - The user will fill in their card information, their card number, CVC, address, and name
    - When this information is filled out, the user will have to click a submit button
    - The information will be validated for legitimate cards with third party companies
    - The purchase information will be stored in the database
    - Administrators can change the look and feel of the website, along with any information in the database.
    - The user will be given information about their purchase and then sent to the destination page.
- Alternate Path: The user will be able to select already stored payment information rather than filling out their information manually. The result, the information page and return to
- Exception Path: If the payment information is not validated or incomplete, the user will be prompted to fix the information.
- Pre-Condition: Valid payment information must be entered and submitted
- Post-Condition: Reaching the destination page through the browser after the pre-condition has been met.

Use Case Number: 3

- Use Case Name: Ticket Submission
- Actors: User, Administration, Database
- Description:
    - On clicking Ticket Submission button, the user will be sent to the ticket submission page
    - The user will fill out the submission information
    - When the information is filled out, the user must press the submission button.
    - The ticket will be submitted to the data base.
    - Administrators will be notified of new tickets submitted to the database.
    - An email containing the ticket number and information will be sent to the email address associated with the account.
    - The database will store the ticket information.
    - The user will be sent to the destination page upon submission.
- Alternate Path: There is no alternate path, the only exception will be the user clicking the home button.

- Exception Path: If the ticket sent is blank, the user will be prompted to fill in information rather than submitting a blank ticket.
- Pre-Condition: The user having an account with the website.
- Post-Condition: The user arrives at the destination page after submission.

Use Case Number: 4

- Use Case Name: User Management Portal
- Actors: Administrator, Database
- Description:
    - The administrator will have a user management portal button or tab in the administrator view of the homepage.
    - When the button is clicked, there will be a check with the database for the administrator's permission level.
    - After the permission level is confirmed, the administrator will be directed to the user management portal homepage which only shows action buttons or tabs and information that is available to that permission level.
- Alternate Path: There are no other paths to get to the user management portal homepage.
- Exception Path: If validation for checking that the person entering the portal is an administrator fails, the person is redirected to the login screen.
- Pre-Condition:
    - Administrator is logged in
    - Administrator clicked "User Management Portal" button on the homepage
- Post-Condition: The administrator will land on the user management portal tailored to his or her permission level after the pre-condition has been met.

Use Case Number: 5

- Use Case Name: Administrator Ticket Management
- Actors: Administrator, Database, User
- Description:
    - The administrator with a permission level that allows ticket management will have a ticket management button or tab available in his or her user management homepage.
    - When the button or tab is clicked, the administrator will be directed to a page or interface that has a list or table containing all tickets from the database and its attributes.
    - There will be options to sort and filter the table where the administrator can sort by columns and ascending or descending order as well as filter for keywords.
    - There will be a button to toggle the view to kanban board style.
    - Depending on the permission level of the administrator, a dropdown will be available for tickets the administrator selects as well as a button to assign the tickets to someone.
    - The dropdown contents in the dropdown will depend on the permission level; one permission level only allows self assignment of unassigned tickets, and another permission level will allow the administrator to assign

    - another administrator a ticket or self assign a ticket regardless of the ticket's initial assignment status.
    - Clicking the assign button after choosing one option from the dropdown will update the ticket's assignment attribute in the database.
    - Administrators will be able to change their assigned ticket status by selecting the ticket and clicking update ticket status button.
    - Updates will be reflected in the database.
    - Tickets that go from unassigned to assigned status will automatically have an email sent to the user that made the ticket, so the user is informed of ticket acknowledgement.
- Alternate Path: There are no other paths to get to ticket management.
- Exception Path: If error with database occurs, administrator is redirected to user management homepage
- Pre-Condition:
    - Administrator is logged in
    - Administrator's permission level allows ticket management
    - Administrator clicked "Ticket Management" button or tab in the user management portal homepage
- Post-Condition:
    - After preconditions are met, the administrator will land on the ticket management interface.
    - Changes in the ticket management interface will be reflected into the database.
    - Tickets that go from unassigned to assigned status will trigger a notification email to the user that submitted the ticket.

Use Case Number: 6

- Use Case Name: Administrator Remove Inactive User
- Actors: Administrator, Database, User
- Description:
    - The administrator with a permission level that allows inactive user management will have a manage inactive user button or tab available in his or her user management homepage.
    - When the button or tab is clicked, the administrator will be directed to a page or interface that has a list or table containing all users from the database that meet the definition of inactive.
    - There will be options to sort and filter the table where the administrator can sort by columns and ascending or descending order as well as filter for keywords.
    - The administrator can select users from the list or table and click 'delete'.
    - Clicking the button will send a notification email to the user to warn the user of deletion if the user does not log on in a stipulated amount of time.
    - Deletion of the selected user will occur after the user does not log on in the stipulated time and will be reflected in the database.
- Alternate Path: There are no other paths to get to inactive user management.

- Exception Path: If error with database occurs, administrator is redirected to user management homepage
- Pre-Condition:
    - Administrator is logged in
    - Administrator's permission level allows inactive user management
    - Administrator clicked "Check for Inactive Users" button or tab in the user management portal homepage
- Post-Condition:
    - After preconditions are met, the administrator will land on the inactive user management interface.
    - Changes in the interface will be reflected into the database.
    - Inactive users will receive an email about imminent deletion before deletion occurs.

Use Case Number: 7

- Use Case Name: Renting
- Actors: User, Administration, Database
    - Description: When clicking on the "Rent" button, the user will be sent to a rent page of different wifi listings.
    - The User will have the option to browse different wifi levels and prices that best suit their needs, rental dates, and device settings.
    - The user will then select a time period that they would like to rent out their wifi for, similar to a calendar.
    - User will click "Rent now" after choosing a date and have the Wifi listing added to a cart.
    - Users will go through payment processing.
    - Then the user will agree to the rental policy and extra charges if not taken care of.
    - User will confirm the rental.
    - The user will then receive a confirmation email stating it was successful.
- Alternate Path: There is no alternate path, if the user chooses not to go with anything, they will only be able to click the home button.
- Exception Path: If the Wifi listing the user chooses is not available, then the user will be prompted to choose another one.
- Pre-Condition: A valid account must be made beforehand.
- Post-Condition: The user will be sent a confirmation email and sent back to the destination screen.

Use Case Number: 8

- Use Case Name: Wifi Listing
- Actors: User, Database
- Description:
  - User will log into their account and will navigate to the listing section
  - The user will create a new listing, inputting their device information.
  - The user will then add their rental date, availability, terms, and any additional fees.

    - User will agree to listing terms and conditions.
    - User will click "publish now" button and the listing will be added to the website.
- Alternate Path: There is no alternative path, if the user does not list anything when prompted, then they will just click the home button or other tabs.
- Exception Path: If the user chooses to enter nothing or partial information, the user will be prompted to either fill it out completely or discard the listing. .
- Pre-Condition: Users will be logged in with their credentials.
- Post-Condition: User successfully lists their wifi to rent on their website.
- 5.2: Requirement Number

Use Case Number: 9

- Use Case Name: Wifi Managing Listings
- Actors: User, Administration, Database
- Description:
    - User will log into their account and will navigate to listing management section in their dashboard
    - Users will view a list of their existing listing in their management section.
    - Users will be able to select which listing from their list they wish to update or edit.
    - They will click on their listing to access the edit screen to make their changes.
    - If desired, the user will be able to update their rental terms, device information, minimum durations, rental dates, and more.
    - Users will also, if need be, update their listing to unavailable.
    - Users will be able to review and save changes to their listing.
- Alternate Path: There is no alternative path, if the user wishes to not edit or change anything, then they will choose the home button.
- Exception Path: If the user chooses to enter nothing, then the listing will not change. If the user does not save their current changes, they will be prompted and asked to either save their changes or discard them.
- Pre-Condition: Users will be logged in with their credentials and have access to their listing management from their dashboard.
- Post-Condition: The changes the user inputs will be reflected on the Wifi listing on the website.

Use Case Number: 10

- Use Case Name: User Registration
- Actors: User, Administrator, Database
- Description:

  o User accesses the registration page provided by the website.
  o The registration form is displayed, prompting user to enter required information such as a username, email, and password.
  o User fills out the registration form with accurate information.
  o User clicks the "Submit" or "Register" button to proceed.

o The system validates the entered information for correctness and completeness.

o If any errors are detected, the system displays error messages indicating the fields that need to be corrected and prompts the user to resubmit the form.

o If the information provided is valid, the system stores the user's registration details in the database.

o User receives a confirmation message or email indicating successful registration.

- Alternate Path: May include optional steps such as profile customization.
- Exception Path: If the entered information does not meet the required format or criteria the system prompts the user to revise the information.
- Pre-Condition: User must be on the website.
- Post-Condition: User registration is successfully completed, and the user's account details are stored in the system

Use Case Number: 11

- Use Case Name: Payment confirmation
- Actors: User, Administrator, Database
- Description:

o User initiates a payment transaction through the website.

o After payment is processed, a confirmation message is generated.

o The confirmation message includes details such as the transaction ID, payment amount, date and time of payment, and a summary of services.

o The user receives the confirmation message and reviews the payment details for accuracy.

o If the user has any questions or concerns regarding the payment, they can contact customer support for assistance.

o The transaction details are logged in the database for record-keeping purposes.

- Alternate Path: There is only 1 path to the destination site.
- Exception Path:
- Pre-Condition: A payment transaction has been initiated and successfully processed
- Post-Condition: User receives payment confirmation message containing details of the transaction

Use Case Number: 12

- Use Case Name: User Feedback
- Actors: User, Administrator, Database
- Description:

o The User interacts with the feedback submission interface provided by the website.

o A Form or dialogue box is presented prompting the user to enter their feedback.

o User provides feedback by typing text or selecting options from predefined categories.

o The feedback is submitted,processed, and stored in the database.

o Feedback may undergo categorization, or prioritization based on importance.

○ A confirmation message is displayed to the user acknowledging successful submission of feedback.

- Alternate Path: May prompt users for feedback at specific touchpoints, such as after completing a transaction.
- Exception Path: If there are technical issues or errors during the submission process, the system may display an error message and prompt the user to retry submitting their feedback.
- Pre-Condition: User is on the website and wishes to provide feedback.
- Post-Condition: User feedback is successfully submitted and stored in the system.

Use Case Number: 13

- Use Case Name: System Maintenance
- Actors: Administrator, System
- Description:
    - Conducting system maintenance to ensure optimal platform performance and reliability.
    - Software updates, updating software components to the latest versions or patches for bug fixes, performance improvements, and security enhancements.
    - Database optimization, improving database performance and efficiency through activities like indexing, defragmentation, and query optimization.
    - Server upgrades, upgrading hardware components or migrating to newer server technologies to enhance performance and scalability.
    - Applying security patches, updating the system with security patches released by software vendors to address known vulnerabilities and enhance security.
- Alternate Path: If the scheduled maintenance window is interrupted due to unforeseen circumstances, administrators reschedule the maintenance activity to minimize service disruptions.
- Exception Path: If critical issues arise during maintenance, such as software update failures or database corruption, administrators halt the process and initiate recovery procedures to restore system functionality.
- Pre-Condition: Administrator authorization and resources for maintenance, users informed in advance of planned downtime.
- Post-Condition: Successful completion of system maintenance, ensuring continued reliability and performance.

Use Case Number: 14

- Use Case Name: User Account Deactivation
- Actors: Administrator, Database
- Description:

  - ○ Deactivating inactive or dormant user accounts to maintain security and manage platform access.
  - ○ Identifying inactive accounts, Identifying user accounts that have been inactive or dormant for a certain period.
  - ○ Changing their status to inactive or suspended, changing the status of inactive accounts to inactive or suspended to prevent unauthorized access.
  - ○ Notifying users, notifying users of the deactivation of their accounts.
  - ○ Providing reactivation instructions if necessary,

- Alternate Path: If an inactive account is mistakenly identified as active, administrators review the account status and verify user activity before deactivation.
- Exception Path: If an account deactivation request fails due to technical issues or database errors, administrators investigate the issue and resolve it promptly to ensure account security and functionality.
- Pre-Condition: Administrator authorization for account deactivation, periodic review of inactive accounts.
- Post-Condition: Successful deactivation of inactive user accounts, reducing security risks and managing platform access effectively.

Use Case Number: 15

- Use Case Name: Extend Wi-Fi Service
- Actors: User, System
- Description:
  - ○ Enables customers to extend their local Wi-Fi service beyond the initial rental period.
  - ○ Customers can request an extension of their Wi-Fi rental, specifying the desired extension duration.
  - ○ The system validates the request, checks for resource availability, and updates the rental period accordingly.
- Alternate Path: If a customer's extension request cannot be processed immediately due to resource constraints, the system notifies the customer and suggests alternative options or a revised extension timeline.
- Exception Path: If the customer's Wi-Fi service extension request fails validation, such as due to invalid extension duration or unavailable resources, the system notifies the customer and provides guidance on resolving the issue.
- Pre-Condition: Customer has access to request Wi-Fi service extension, system resources available for extension.
- Post-Condition: Successful extension of Wi-Fi service rental period as requested by the customer, ensuring continued access to Wi-Fi connectivity.

- 5.2: Requirements

  Requirement Number: 1
  - Use Case Number: 1
  - Introduction: Easy feel and look for a login to go to the destination page, using a username and a login.
  - Inputs: Username and password
  - Requirement Description: The user must be able to sign up, given all requirements are met
    - All fields must be complete
    - Requirements from the system are met
    - Database validation is met
  - Outputs: Routed to destination page

  Requirement Number: 2
  - Use Case Number: 2
  - Introduction: Easy use and secure payment system
  - Inputs: Card Number, CVC, Expiration Date, Address, Name on Card
  - Requirement Description:
    - All fields must complete
    - The credit card information validation must be met
    - Database storage is completed
  - Outputs: Confirmation of payment and payment details

  Requirement Number: 3
  - Use Case Number: 3
  - Introduction: Quick, Easy, and versatile Ticket System
  - Inputs: Ticket Information
  - Requirement Description:
    - Information filled in
    - Submission button clicked
    - Ticket information storage in database
    - Ticket information emailed to customer
  - Outputs: Ticket Number emailed to customer and details of ticket

  Requirement Number: 4
  - Use Case Number: 4
  - Introduction: Easy organized look and feel page for administrators to manage users, view tasks and manage tickets. Ability to view and manage depends on the administrator's permission level.
  - Inputs: Administrator credentials
  - Requirement Description:
    - Given administrator is logged in,
    - Administrator’s permission level is checked then user management welcome page shows only actions that the permission level allows.

    - The administrator can click options available to see more.
- Outputs:
    - If action is clicked, administrator is navigated to the respective view.
    - Else, log administrator out if no activity is detected

Requirement Number: 5

- Use Case Number: 5
- Introduction: List of tickets with current status and assigned administrator is shown in a table that can be filtered and sorted. List can be viewed in kanban board style as well. Depending on ticket status, administrator can assign ticket to self or assign to other administrators (only if that administrator has permission)
- Inputs: Administrator credentials, optional keywords for filtering table
- Requirement Description:
    - Given administrator is logged in and has appropriate permission level,
    - List or table is shown containing all tickets and their attributes from the database.
    - Administrator can filter and sort list or table and can type in keywords to search within table
    - View of list or table can be toggled to kanban board style.
    - Depending on permission level, the administrator can click a drop down to assign self an unassigned ticket.
    - Depending on permission level, an administrator can further assign a ticket to another administrator or reassign a ticket to self.
- Outputs:
    - If changes are made to tickets, respective ticket information in database is updated
    - If unassigned tickets become assigned, email notification to user is sent.
    - If idle (no activity detected), administrator is logged out

Requirement Number: 6

- Use Case Number: 6
- Introduction: Interface for administrator to handle deleting inactive users and reminding users that have an unclaimed balance made from renting out wifi. Administrators can view users' last login in a table, email and unclaimed balance on a table or list that the administrator can filter and sort.
- Inputs: Administrator credentials, optional keywords for filtering table
- Requirement Description:
    - Given administrator is logged in and has appropriate permission level,
    - List or table of users that meet a given inactivity time (such as last login date was over 6 months) is shown.
    - List or table can be filtered and sorted, and administrators can type in keywords to search within table
    - Administrator can click a button to email inactive users that the user's account will be deleted if last login date is not updated by a certain time.
    - Administrators can click a button to email inactive users that have an unclaimed balance from renting out wifi.

- Outputs:
  - If action buttons were clicked, users are sent emails.
  - Database entries are updated if deletion criterias were met.
  - If idle (no activity detected), administrator is logged out.

Requirement Number: 7

- Use Case Number: 7
- Introduction: This requirement outlines how users will rent Wifi from the website.
- Inputs:
  - User credentials
  - Selection of Wifi to rent
  - Specification of rental term
  - Rental agreements
  - Payment information
- Requirement Description: This system allows users to browse and select their wifi they wish to rent. Users will have to provide the rental date they wish to go with and payment information.
- Outputs:
  - Confirmation details in the form of email
  - Rental details as well as device information
  - Access to rented wifi to use during rental period

Requirement Number: 8

- Use Case Number: 8
- Introduction: This requirement outlines the process users will go through to list their wifi to rent on the website.
- Inputs:
  - User login
  - Wifi device details
  - Rental terms
  - Agreement to list terms/conditions.
- Requirement Description: The system should provide a listing interface to input their wifi details to list onto the website as well as their rental terms and agree to listing agreements.
- Outputs:
  - A confirmation message informing the user of a successful listing
  - Published listing for users browsing for available internet.

Requirement Number: 9

- Use Case Number: 9
- Introduction: This requirement showcases the functionality of managing their listings on the website. Users should be able to edit listing terms, change rental dates or take down their listings whenever they desire.
- Inputs:
  - User credentials
  - Access to listing management section in users dashboard
  - Selection of specific listing or edit

- Requirement Description: This system should be able to allow users to edit their listings whenever, update rental dates, device info, and more. On top of this as well, it also should allow the user to remove their listing from the website due to either damage, currently in use, or other reasons.
- Outputs:
    - A confirmation stating the changes and edits have been made and saved.
    - The wifi listing is updated to whatever the user has chosen to change.

Requirement Number: 10

- Use Case Number: 10
- Introduction: User-friendly registration process
- Inputs: Personal information (name, dob, etc), email address, password
- Requirement Description:
    - Mandatory fields in the registration form must be filled out by the user
    - Validated data securely stored within the system's database
- Outputs: Account details stored in the system's database, allowing them to log in.

Requirement Number: 11

- Use Case Number: 11
- Introduction: Payment confirmation involves verifying user-provided payment details and obtaining user confirmation to finalize a transaction.
- Inputs: clicking a "Confirm Payment" button or similar action
- Requirement Description:
    - Payment is confirmed, action serves as the input to initiate the finalization of the payment process.
- Outputs: Message or email containing details of the transaction.

Requirement Number: 12

- Use Case Number: 12
- Introduction: This process collects user input
- Inputs: Text input, ratings, or selection from predefined categories
- Requirement Description:
    - Validate submitted feedback to ensure it meets basic requirements
    - Store feedback in the system's database
- Outputs: Confirms successful submission of feedback to the user, through an on-screen notification or a confirmation message

Requirement Number: 13

- Use Case Number: 13
- Introduction: System Maintenance
- Inputs: System maintenance triggers, administrator credentials
- Requirement Description:
    - The system must undergo routine maintenance to ensure optimal performance and security.
    - Software updates, database optimization, server upgrades, and security patching.
    - Administrators are responsible for initiating and overseeing maintenance activities.

- Outputs: Confirmation of maintenance completion, system stability, and security enhancements.

Requirement Number: 14

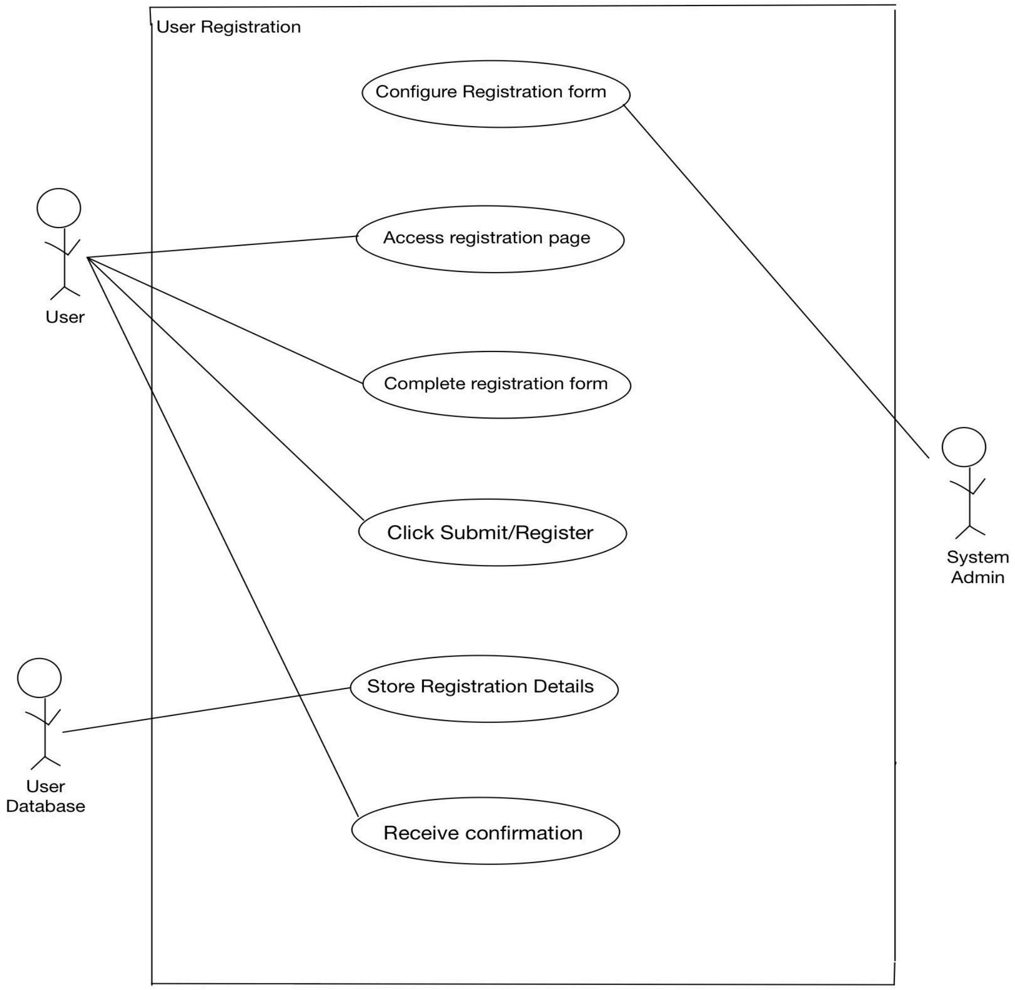
- Use Case Number: 14
- Introduction: User Account Deactivation
- Inputs: Administrator credentials, inactive user accounts
- Requirement Description:
    - Inactive user accounts must be deactivated to enhance system security and manage access.
    - Administrators review and identify inactive accounts, change their status to inactive or suspended, and notify users accordingly.
    - Deactivated accounts may be reactivated upon user request or system validation.
- Outputs: Notification to users regarding account deactivation, enhanced system security.

Requirement Number: 15

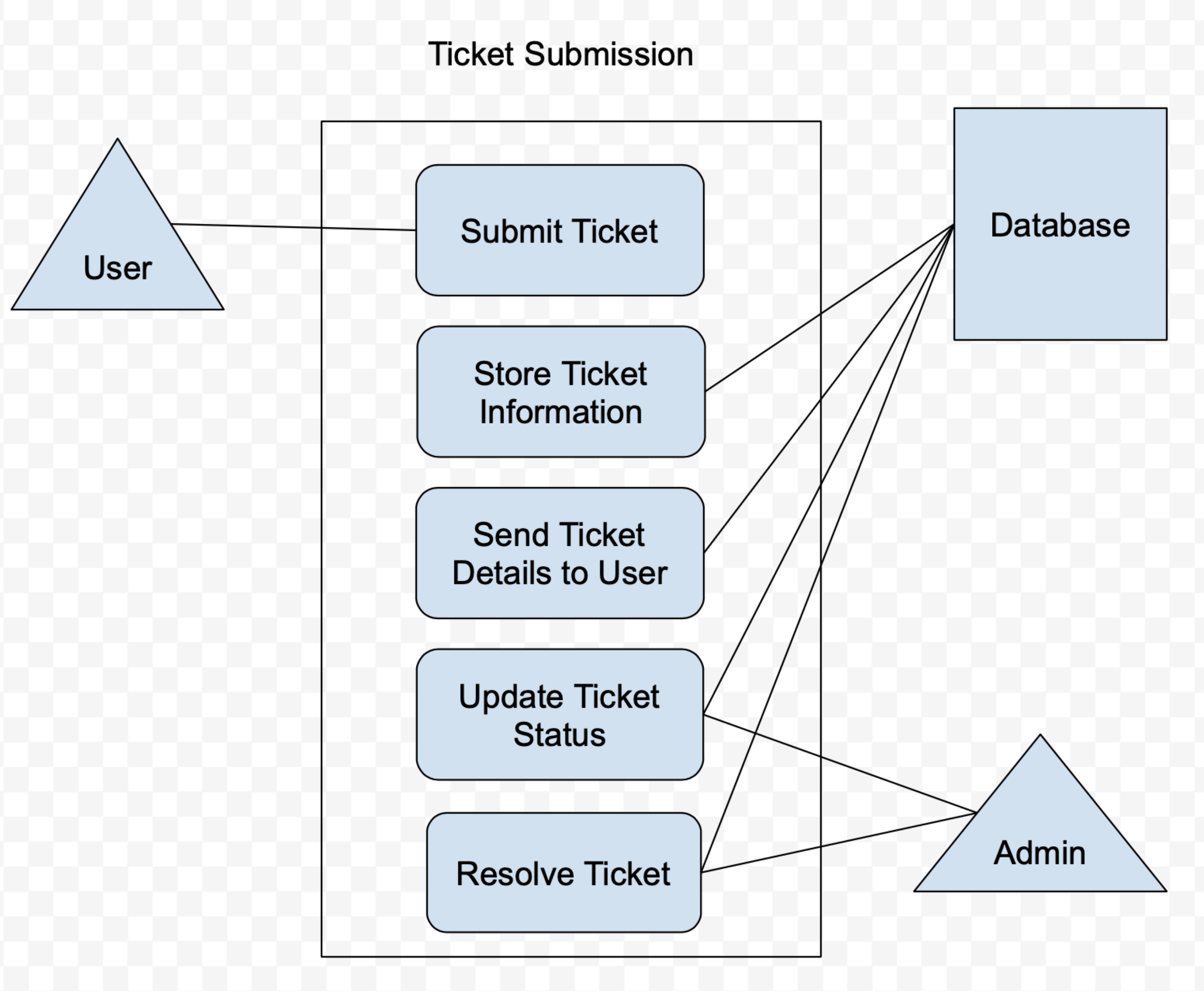
- Use Case Number: 15
- Introduction: Extend Local Wi-Fi Service
- Inputs: Customer's request for extended Wi-Fi service, system resources
- Requirement Description:
    - The system must allow customers to extend their local Wi-Fi service beyond the initial rental period.
    - Customers should have the option to request an extension of their Wi-Fi rental, specifying the desired extension duration.
    - The system will validate the request, check for resource availability, and update the rental period accordingly.
- Outputs: Confirmation of Wi-Fi service extension, updated rental details, continued access to extended Wi-Fi service.

- 5.3: Use Case Diagrams

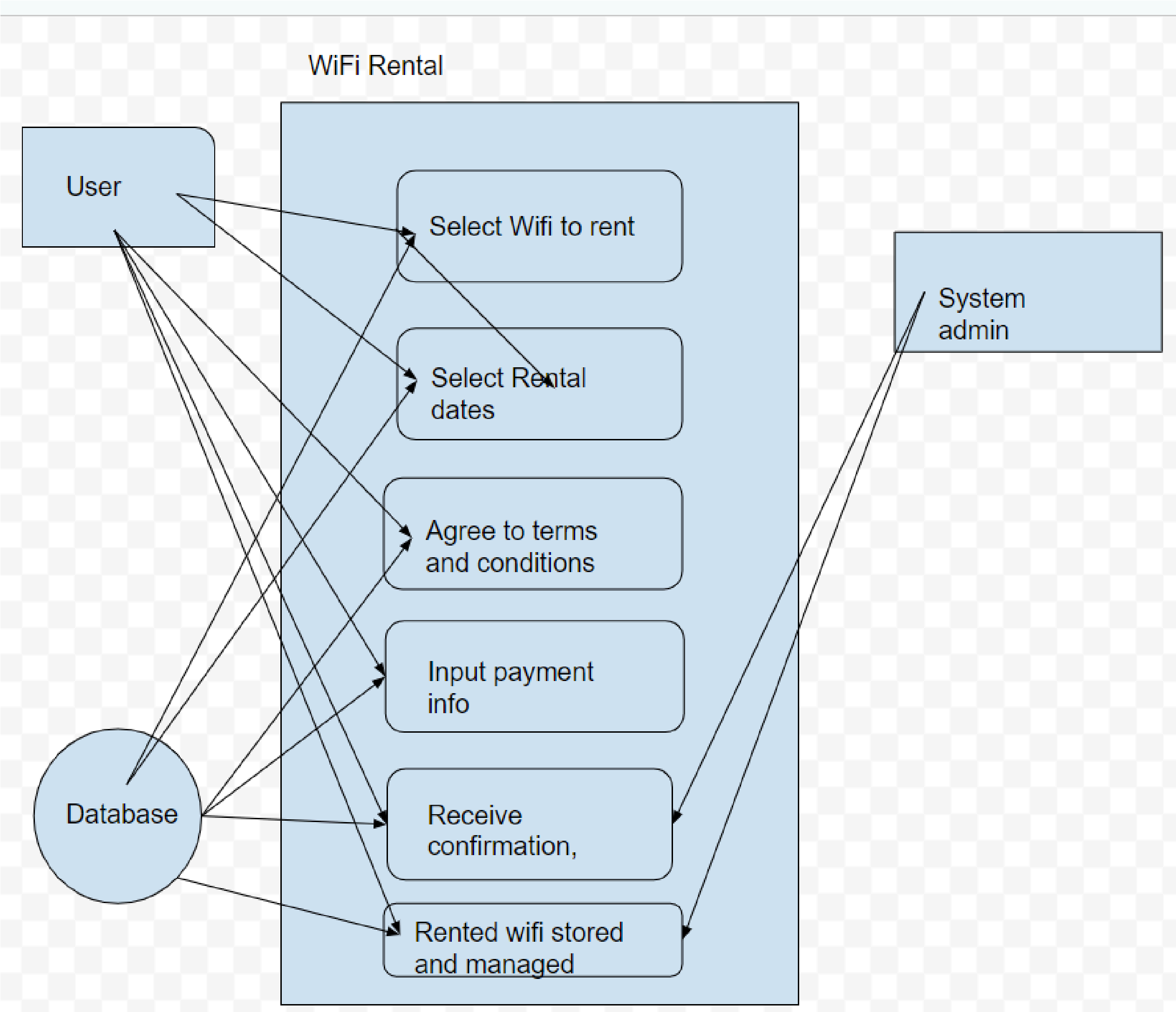
<u>User Registration</u>



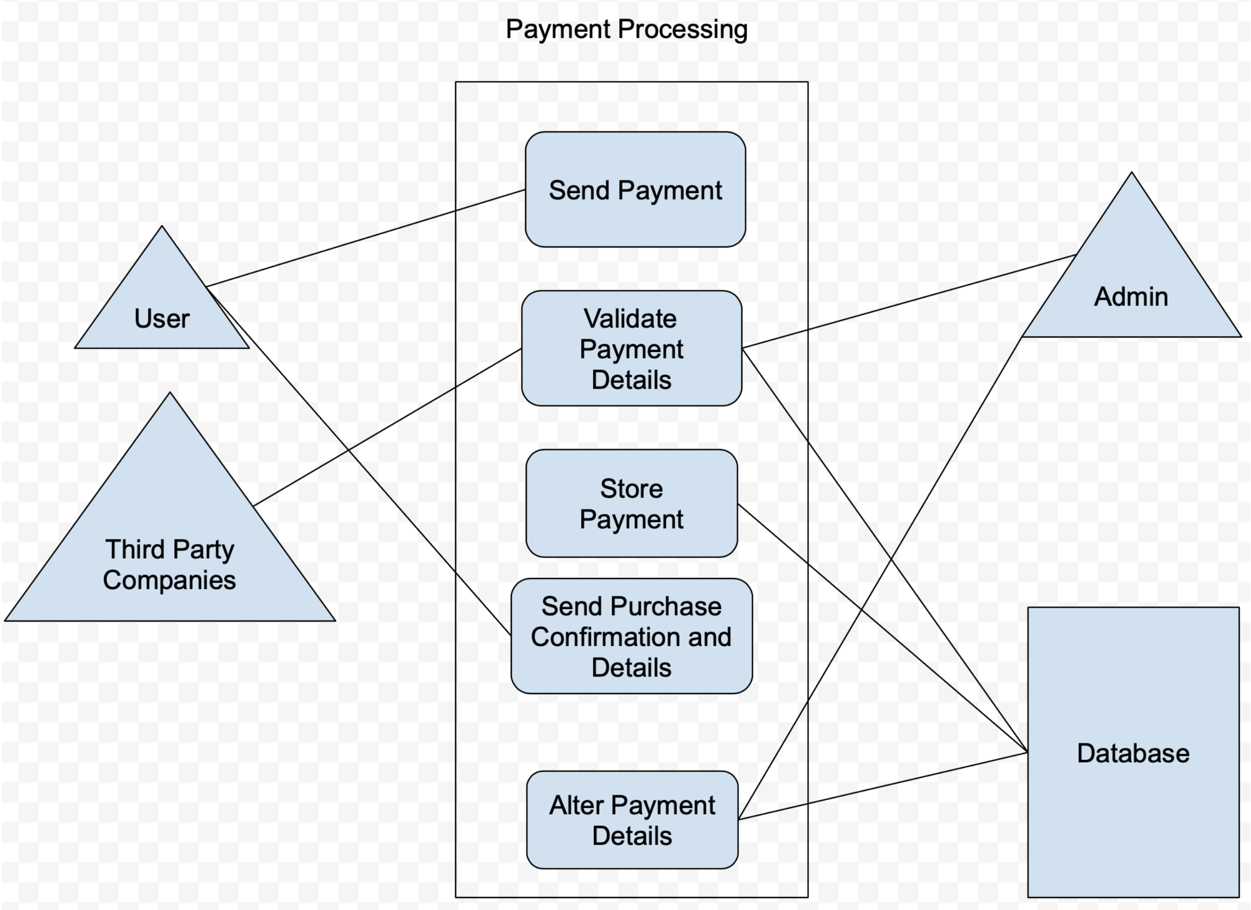
Ticket Submission


Ticket Submission
User
Submit Ticket
Store Ticket Information
Send Ticket Details to User
Update Ticket Status
Resolve Ticket
Database
Admin

Wi-Fi Rental

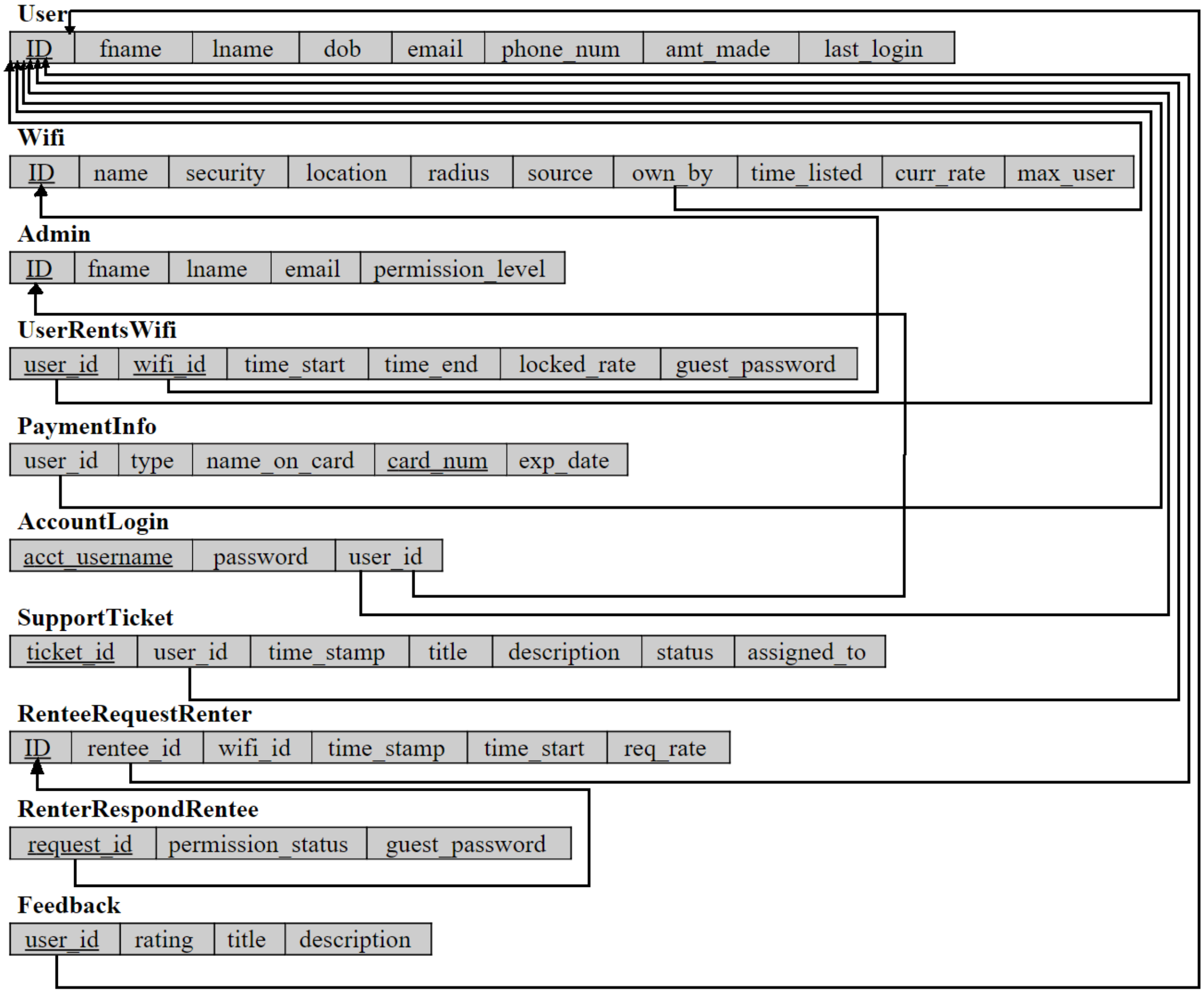


Payment Processing


Payment Processing
User
Third Party Companies
Send Payment
Validate Payment Details
Store Payment
Send Purchase Confirmation and Details
Alter Payment Details
Admin
Database

Section 6: Database Management

System Database Tables:



User

ID	fname	lname	dob	email	phone_num	amt_made	last_login

Wifi

ID	name	security	location	radius	source	own_by	time_listed	curr_rate	max_user

Admin

ID	fname	lname	email	permission_level

UserRentsWifi

user_id	wifi_id	time_start	time_end	locked_rate	guest_password

PaymentInfo

user_id	type	name_on_card	card_num	exp_date

AccountLogin

acct_username	password	user_id

SupportTicket

ticket_id	user_id	time_stamp	title	description	status	assigned_to

RenteeRequestRenter

ID	rentee_id	wifi_id	time_stamp	time_start	req_rate

RenterRespondRentee

request_id	permission_status	guest_password

Feedback

user_id	rating	title	description

Database Management System:

PostgreSQL

Schema Statements:

User(ID, fname, lname, dob, email, phone_num, amt_made, last_login)
User(ID:integer, fname: string, lname: string, dob: date, email: string, phone_num: string, amt_made: decimal, last_login: datetime)

Wifi(ID, name, security, location, radius, source, own_by, time_listed, curr_rate, max_user)
Wifi(ID: integer, name: string, security: string, location: point, radius: decimal, source: string, own_by: integer, time_listed: datetime, curr_rate: decimal, max_user: boolean)

Admin(ID, fname, lname, email, permission_level)
Admin(ID: integer, fname: string, lname: string, email: string, permission_level: string)

UserRentsWifi(user_id, wifi_id, time_start, time_end, locked_rate, guest_password)
UserRentsWifi(user_id: integer, wifi_id: integer, time_start: datetime, time_end: date time, locked_rate: decimal, guest_password: string)

PaymentInfo(user_id, type, name_on_card, card_num, exp_date)
PaymentInfo(user_id: integer, type: string, name_on_card: string, card_num: string, exp_date: date)

AccountLogin(acct_username, password, user_id)
AccountLogin(acct_username: string, password: string, user_id: integer)

SupportTicket(ticket_id, user_id, time_stamp, title, description, status, assigned_to)
SupportTicket(ticket_id: integer, user_id: integer, time_stamp: timestamp, title: string, description: string, status: character, assigned_to: integer)

RenteeRequestRenter(ID, rentee_id, wifi_id, time_stamp, time_start, req_rate)
RenteeRequestRenter(ID: integer, rentee_id: integer, wifi_id: integer, time_stamp: timestamp, time_start: datetime, req_rate: decimal)

RenterRespondRentee(request_id, permission_status, guest_password)
RenterRespondRentee(request_id: integer, permission_status: boolean, guest_password: string)

Feedback(user_id, rating, title, description)
Feedback(user_id: integer, title: string, description: string)

Relationships:

Parent Tables

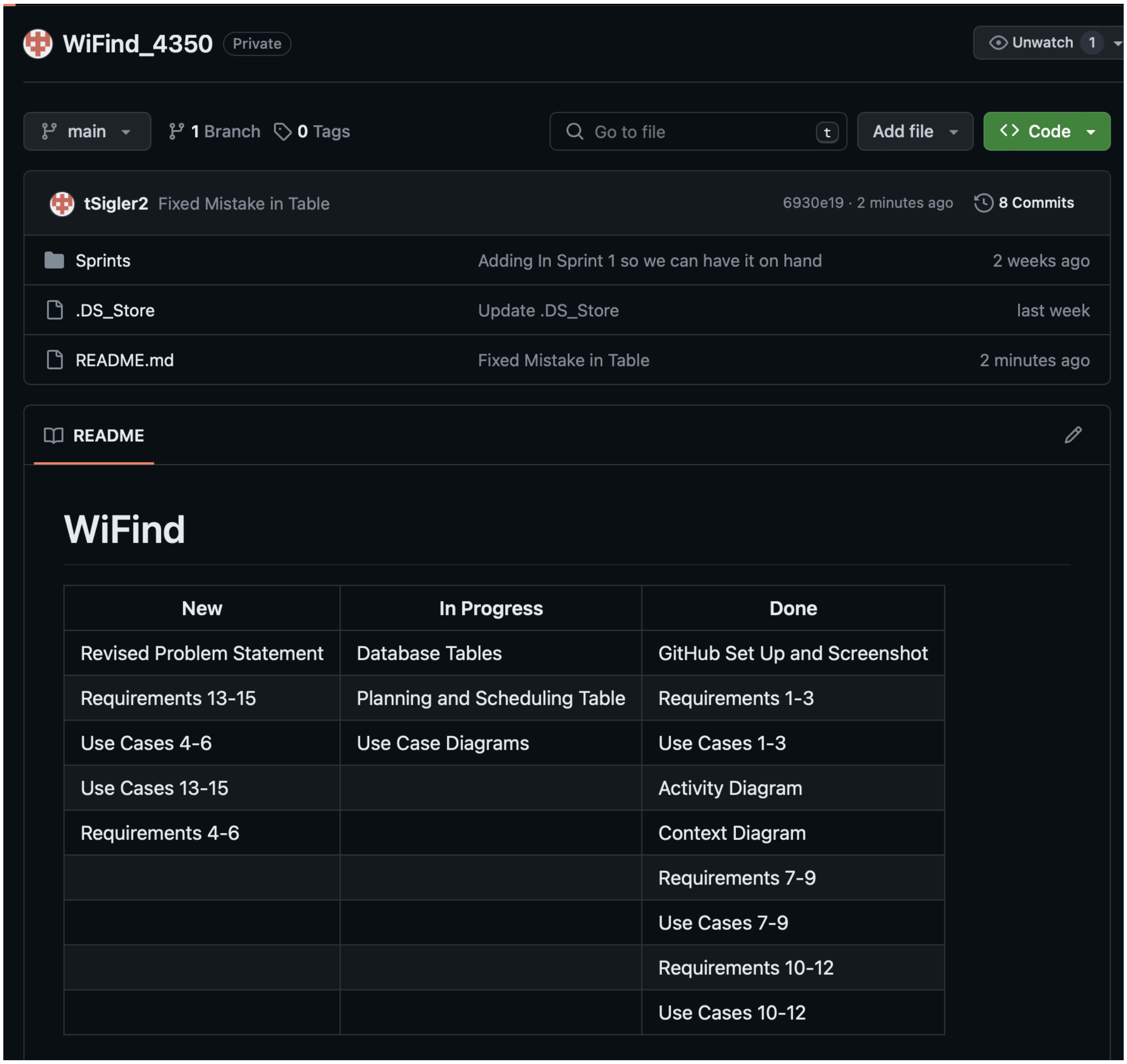
User: ID (PK), fname, lname, dob, email, phone_num, amt_made
Admin: ID (PK), fname, lname, email, permission_level

Child Tables

Wifi: ID (PK), name, security, location, radius, source, own_by (FK), time_listed, curr_rate, max_user
UserRentsWifi: user_id (FK), wifi_id (FK), time_start, time_end, locked_rate, guest_password
PaymentInfo: user_id (FK), type, name_on_card, card_num (PK), exp_date
AccountLogin: acct_username (PK), password, user_id(FK)
SupportTicket: ticket_id (PK), user_id (FK), time_stamp, title, description, status, assigned_to (FK)
RenteeRequestRenter: ID (PK), rentee_id (FK), wifi_id (FK), time_stamp, time_start, req_rate
RenterRespondRentee: request_id (FK), permission_status, guest_password
Feedback: user_id (FK), rating, title, description

Section 7: Github Link and Screenshot

https://github.com/tSigler2/WiFind_4350



tSigler2 Fixed Mistake in Table 6930e19 · 2 minutes ago 8 Commits

Sprints	Adding In Sprint 1 so we can have it on hand	2 weeks ago
.DS_Store	Update .DS_Store	last week
README.md	Fixed Mistake in Table	2 minutes ago

WiFind

New	In Progress	Done
Revised Problem Statement	Database Tables	GitHub Set Up and Screenshot
Requirements 13-15	Planning and Scheduling Table	Requirements 1-3
Use Cases 4-6	Use Case Diagrams	Use Cases 1-3
Use Cases 13-15		Activity Diagram
Requirements 4-6		Context Diagram
		Requirements 7-9
		Use Cases 7-9
		Requirements 10-12
		Use Cases 10-12