

Snip Itz



Group members: Simron Dhali, Alex Haag, Ryan Menas, Carter Trowitch

System Requirement

Purpose

The purpose of this system is to provide an easy to use and efficient means of managing schedules for commission-based employees (specifically hair stylists). The system will allow flexible methods of scheduling including automatic scheduling and request based scheduling. The system will also include a remotely hosted website which will be the interface for all of these operations.

Sponsor

The sponsor for this project is Kelly Knecht, owner of Savvy Hair Gallery and Spa. The mission statement is to "Keep Stylists Behind the Chair." The goal is to minimize the amount of time the employees or owner must take to set, rearrange, or fix schedules giving them more time with clients and allowing more appointments to be made each day.

Scope

- The system will hold schedules for employees to fill
- The system will allow clients to make appointments online
- The system will allow clients to cancel appointments online
- The system will allow employees of owners to manage schedules
- The system will apply a relevant time block for appointments
- The system will allow for breaks and time off in the schedule
- The system will allow all users to access using the website
- The system will allow certain employees to receive 'walk-in' clients
- The system will allow clients to select preferred employee
- The system will select employees if none are preferred
- The system will allow for the creation of employee accounts
- The system will allow for the creation of customer accounts
- The system will send out a confirmation email before an appointment has been set

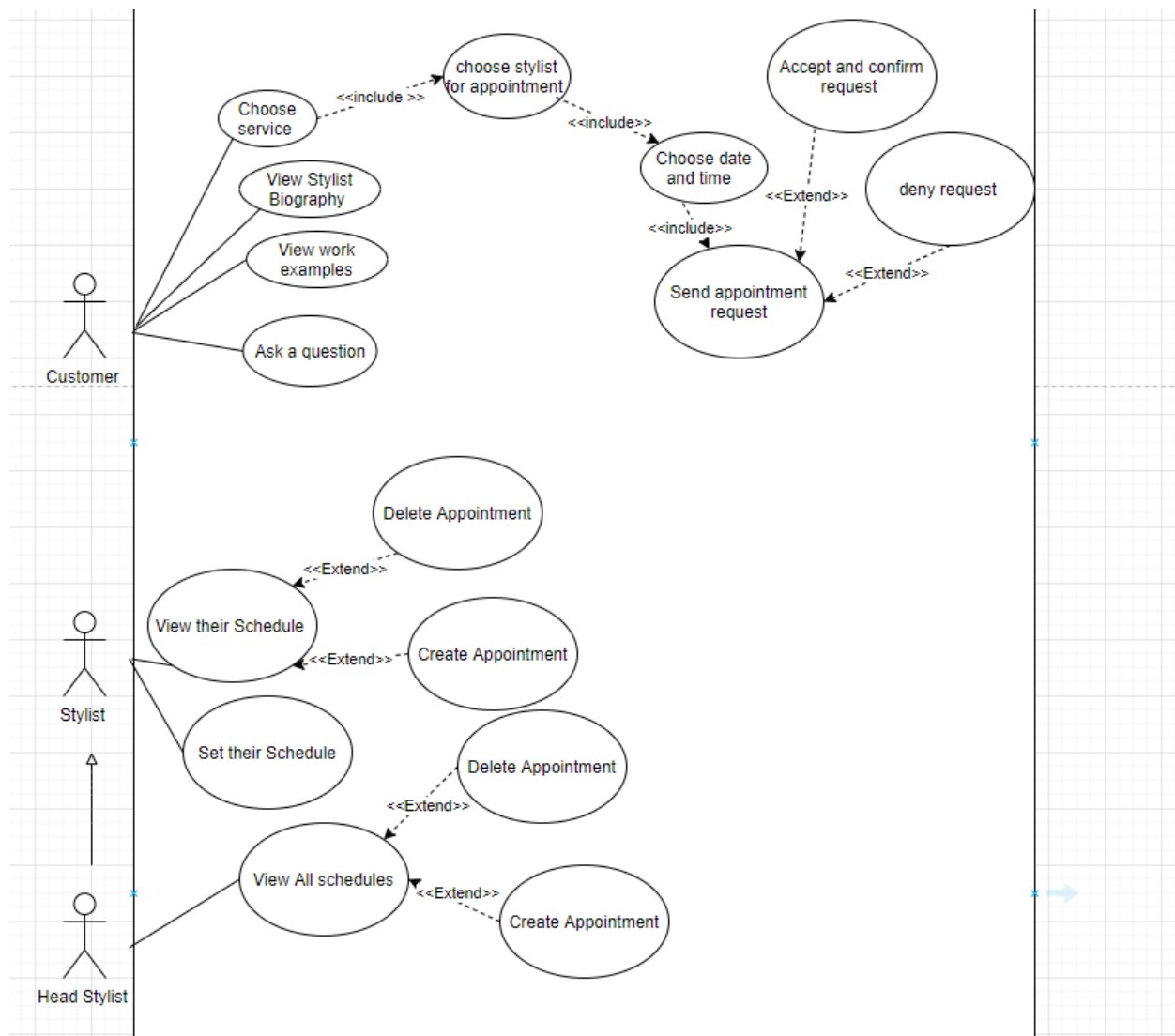
User Characteristics

Users will be the owner (head stylist) who will have universal privilege over all functions within the system. The employees (stylists) who will have administrative privileges over their own schedules. Lastly there will be customers who will be able to create accounts and schedule appointments for themselves. Customers will request a time slot for an appointment, and it will be confirmed by the stylist or owner then a confirmation message will be sent to the customer to set the appointment.

Size Constraint

This system will use a front end website, back end php to communicate with a server that will hold all of the owner, employee, customer, and schedule information. The size of this directly correlates to the price of running this system. Currently, we estimate that to maintain this software with the ,current customer base it will cost approximately \$30 a month. This may vary because currently they have one location

Use case diagrams



Initiator	Goal	Participants	Name
Customer	Pick the service that they request from the salon (ex: hair cut, hair coloring)	Database, php, front end	Choose Service
Customer	Pick the stylist that they would like to perform the service.	Database, php, front end, Choose Service	Choose Stylist for appointment
Customer	Pick the date and the time frame to have their appointment.	Database, php, front end, Choose Service, Choose Stylist for appointment	Choose date and time
Customer	When they complete all the previous requirements they will put a request for an appointment to be made.	Database, php, front end, Choose Service, Choose Stylist for appointment, choose date and time	Send appointment request
Customer	This is where the appointment will either be accepted with a confirmation or denied and sent back to the start.	Database, php, front end, Choose Service, Choose Stylist for appointment, choose date and time, Send appointment request	Accept and Confirm request/ Deny request
Customer	The customer can view different stylist biographies/ about pages	Front end	View stylist biography
Customer	The customer can view different stylist works previously done	Front end	View work examples
Customer	This is where a customer can put a question and it will be sent to Kelly's email	Front end	Ask a question
Stylist	This is where the stylist can view what appointments they have made	Database, php, front end	View their schedule
Stylist	This is where the stylist can set what times are	Database, php, front end	set their schedule

	available to be scheduled for an appointment		
Stylist	This is where the stylist can add an appointment to their schedule	Database, php, front end, view their schedule	create appointment
Stylist	This is where the stylist can delete an appointment to their schedule	Database, php, front end, view their schedule	delete appointment
Head Stylist	This is the supervisor role that can see all of the other stylist schedules	Database, php, front end	View all schedules
Head Stylist	This is where the stylist can add an appointment to any schedule	Database, php, front end, view their schedule	create appointment
Head Stylist	This is where the stylist can delete an appointment to any schedule	Database, php, front end, view their schedule	delete appointment

Class Diagrams

In object-oriented modeling, class diagrams are the most important component. They are used to display the various objects in a system, as well as their characteristics, operations, and interrelationships. In particular ways, classes are linked to one another. Relationships in class diagrams, in particular, include various kinds of logical connections. Let us go more in depth.

Association : is a broad term that refers to any logical relationship or connection between classes. Customers and stores, for example, could be connected.

Directed Association : A line with an arrowhead represents a directional connection. A container-contained directional flow is represented by the arrowhead.

Reflexive Association : When a class has several tasks or obligations, this happens. A stylist, eyebrow expert, nail artist, or waxer, for example, is a member of the salon staff. There may be a managed by partnership in two examples of the same class if the waxer is handled by the stylist.

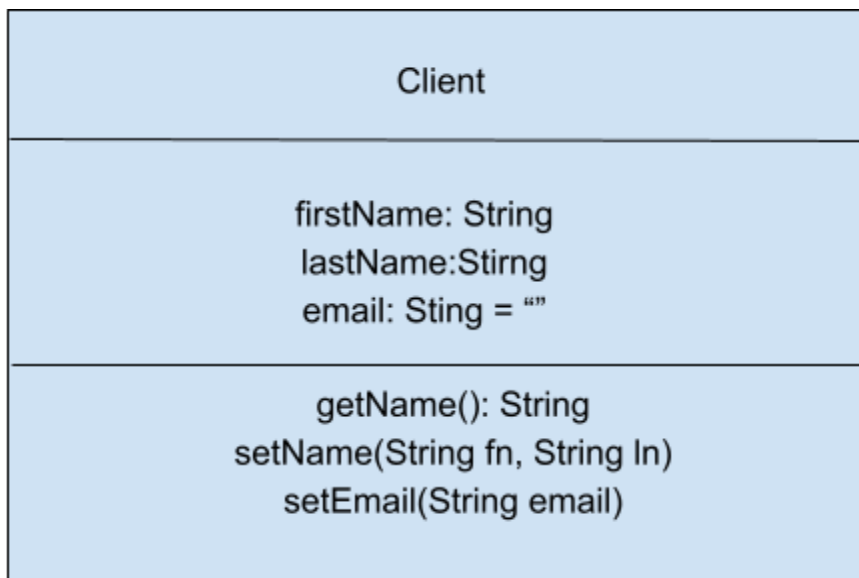
Multiplicity: When the cardinality of one class in relation to another is represented, is the active logical association. One fleet, for example, may include multiple salons, with each salon containing zero to many customers. In the diagram, the notation 0..* stands for "zero to many."

Aggregation: refers to the formation of a specific class as a result of the aggregated or constructed as a collection of other classes. The class "salon," for example, is made up of one or more clients, as well as other materials. The contained classes in aggregation are not overly reliant on the container's lifecycle. Clients, for example, will continue to be clients even though the salon is closed. Draw a line from the parent class to the toddler class with a diamond shape near the parent class to illustrate aggregation in a diagram.

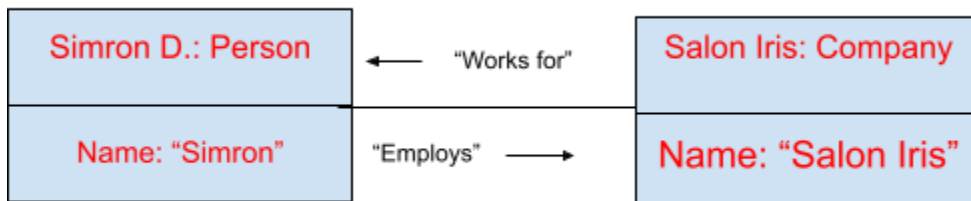
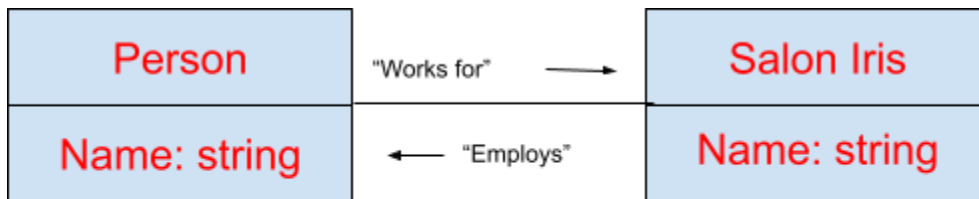
Composition: The composition relationship resembles the aggregation relationship in many ways. with the exception of its primary goal of stressing the contained class's

reliance on the container class's life cycle. That is, when the container class is demolished, the contained class is obliterated as well. The side pocket of a shoulder bag will also vanish once the shoulder bag is destroyed. Use a directional line connecting the two classes in a UML diagram, with a filled diamond shape adjacent to the container class and the directional arrow to the contained class, to illustrate a composition relationship.

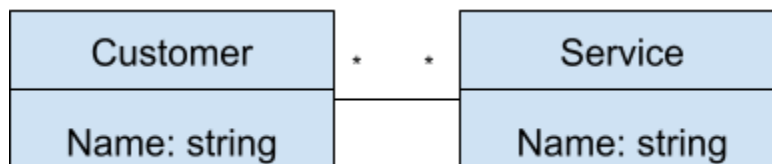
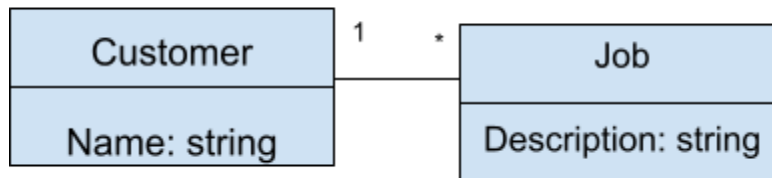
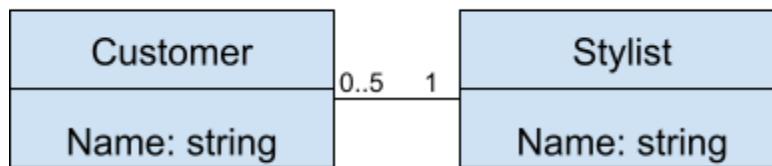
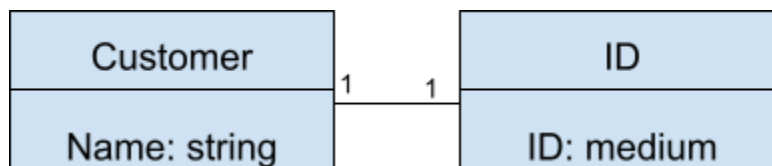
Class Diagrams-Class



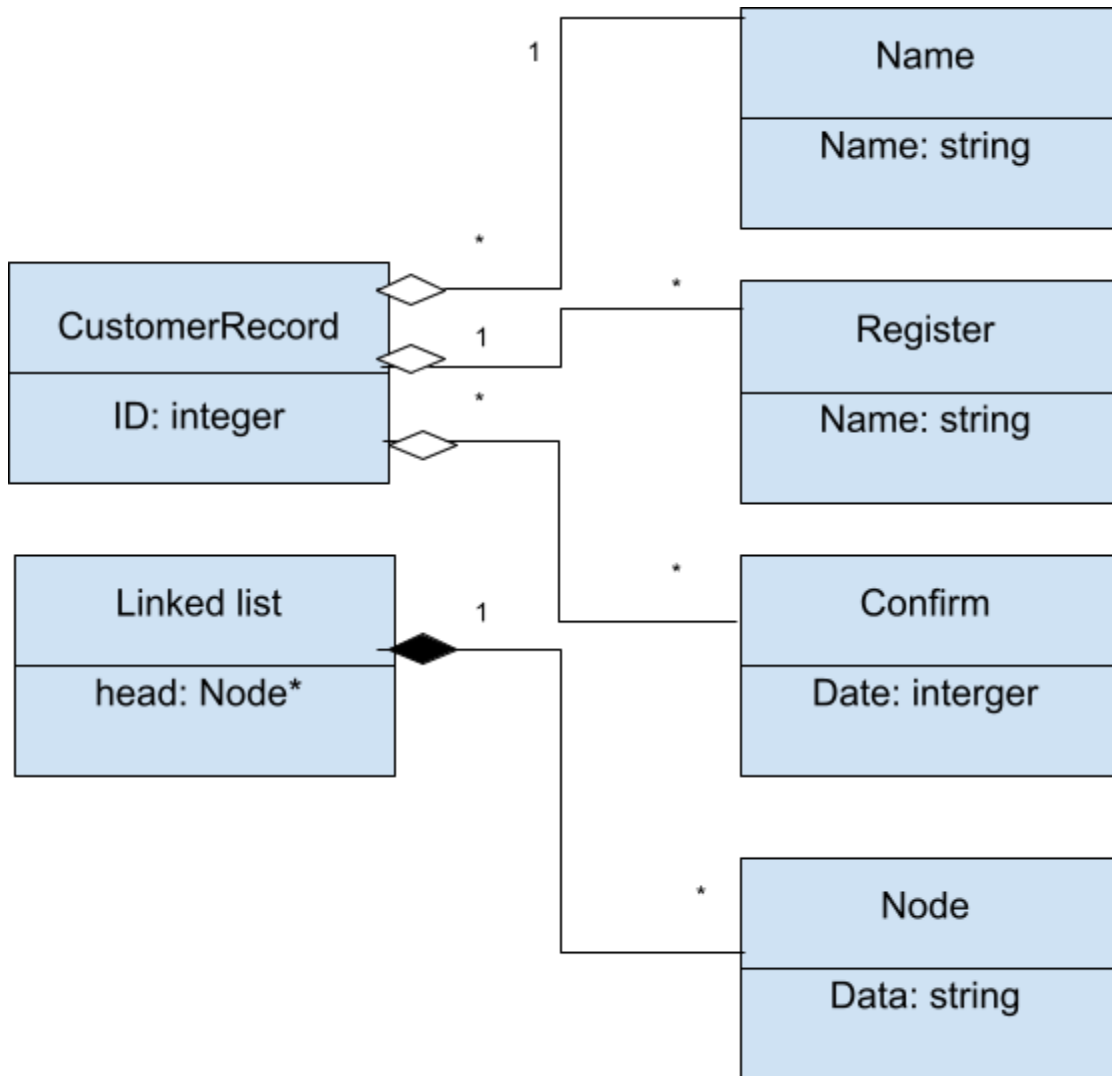
Class Diagrams-Association Class Diagram



Class Diagrams-Multiplicity



Class Diagrams-Composition and Aggregation



Activity Diagrams

Another important behavioral diagram in the UML diagram is the activity diagram, which is used to illustrate dynamic aspects of the system. An activity diagram is a more advanced version of a flow chart that depicts the flow of information from one activity to the next. At various levels of abstraction, activity diagrams explain how activities are coordinated to provide a service.

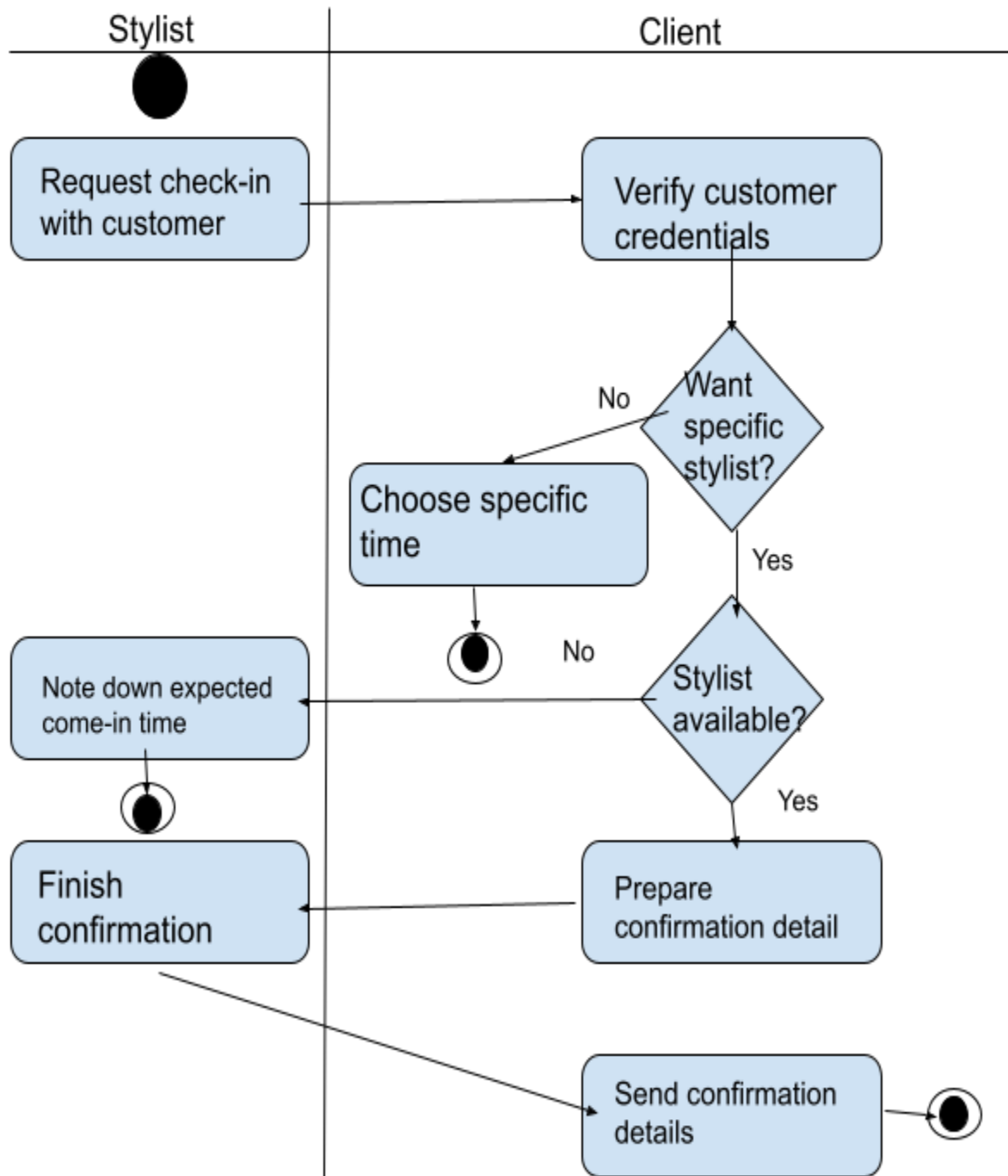
Typically, some operations are required to complete an event, especially when the operation is intended to accomplish a number of different things that necessitate coordination, or how the events in a single use case relate to one another, particularly in use cases where activities overlap and coordination is required. It can also be used to model how a group of use cases work together to represent business workflows.

Now let's talk about action. It's a habit that's broken down into one or more steps. Activities are a network of nodes with edges connecting them. Action nodes, control nodes, and object nodes are all possible. Some action is represented by action nodes. Control nodes represent the control flow of an activity. Object nodes are used to classify the objects used within a given activity. Edges are used to depict a flow of execution or a route. The activities begin at the first node and end at the last node.

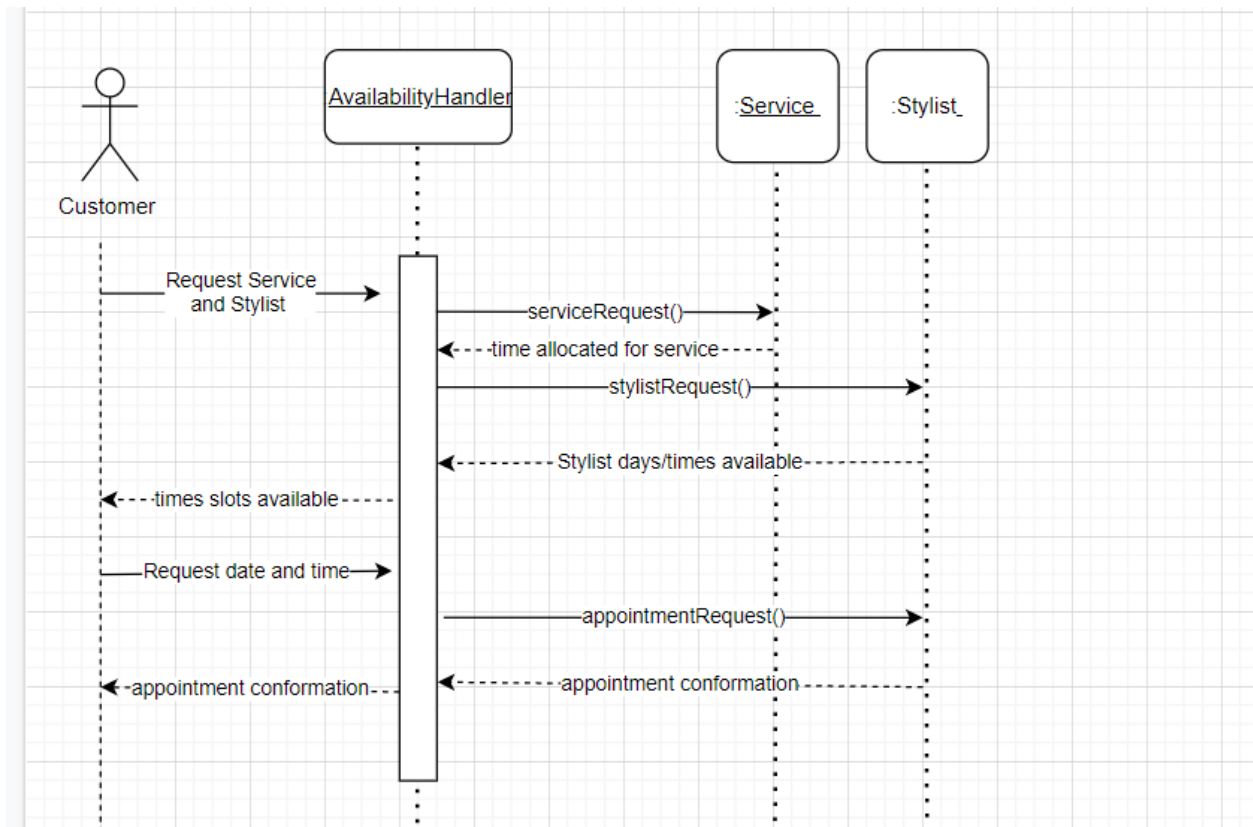
Now talking about abstraction in an activity diagram, a dependence between model components that expresses the same idea at various levels of abstraction or different perspectives is known as an abstraction relationship. Several diagrams, including use-case, class, and component diagrams, can be used to bring abstraction relationships to a model.

The activity diagrams serve the same underlying purposes as the other four diagrams. It depicts the system's dynamic behavior. The other four diagrams illustrate message flow from one object to the next, while the activity diagram shows message flow from one activity to another. The activity diagram will be used to model the system's activity flow. Multiple technologies may be used in a single application. These systems are often captured in an activity diagram, representing the flow from one system to the next.

Activity Diagram



Sequence Diagrams



A sequence diagram is a dynamic view of the system that shows the flow of messages when creating an appointment.

Plan of work

This State Machine Diagram shows how appointments will be checked and requested. Appointments will be put in through a request and then processed by the employee to be approved or declined. Choosing the specific stylist to see their calendar, available then adding an appointment on specific time/ day. First, the customer will be asked to choose a stylist; then, after one has been chosen, they will be taken to their calendar. The calendar will show all the available times the stylist is available and will then request the appointment that best fits best. The appointments are based on the average time to complete the appointment, as some take longer depending on the person. Once the stylist confirms or denies the request, the customer will receive a receipt.

