

Milestone 2 - Stakeholders and scenario-based elements

Stakeholders:

Developers

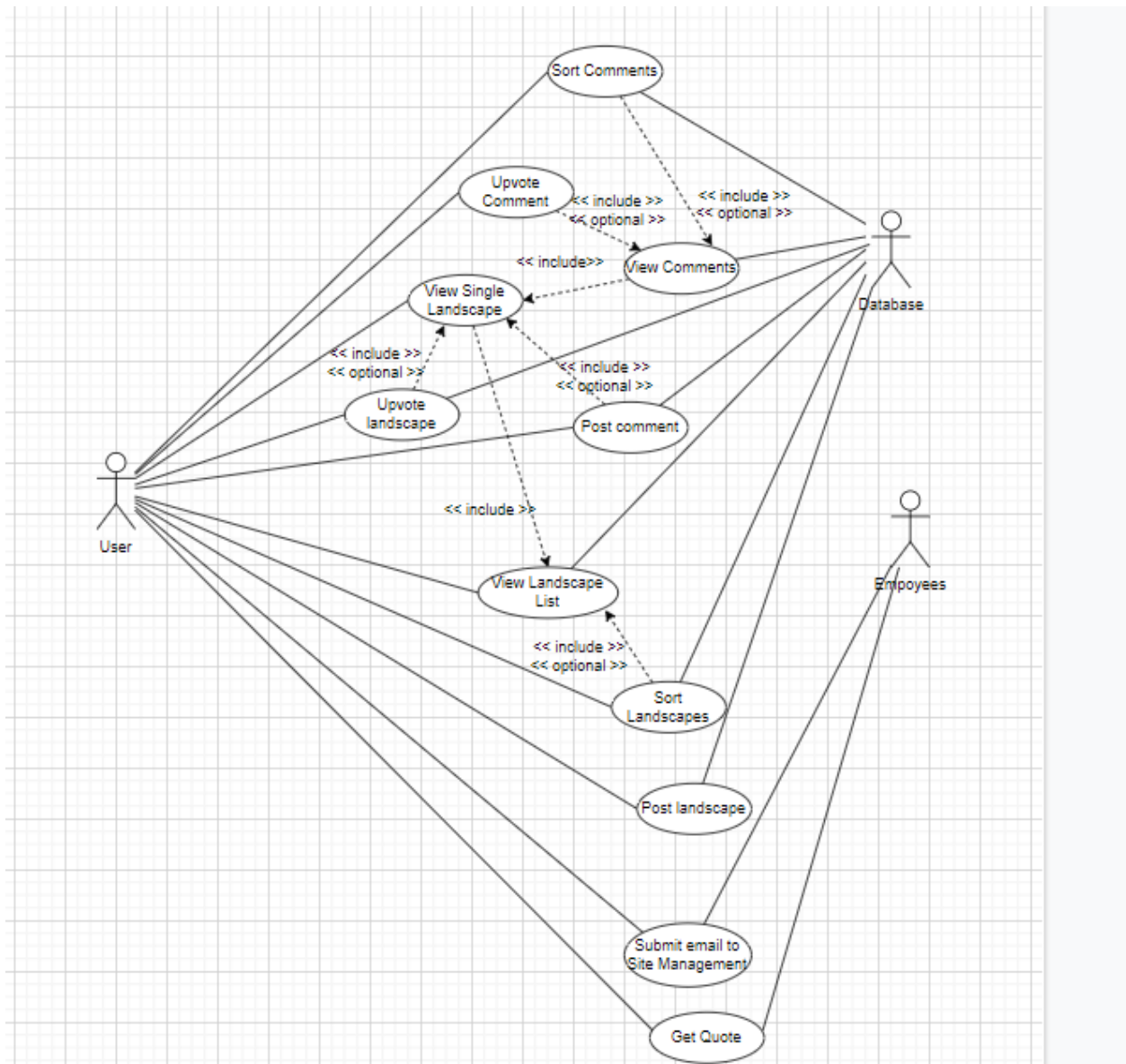
The programmers for the website

Landscapers

Businesses or homeowners who want a quota on their property or general advice.

Note: There will be no investors due to being a class project

Diagram:



UML Diagram Tool:

For our UML diagram tool we used Diagrams.net which allows for users to easily create diagrams and flowcharts. There are many reasons why we decided on this tool, the main reason we decided is that this tool is integrated with Google Drive, which allows for collaboration on the diagram to occur synchronously for all group members. Diagrams.net also has an easy to understand UI and allows for the diagrams to be stylized to our liking.

UML Use Cases:**Submit Email To Site Management**

Actors:	User Employees
Goal:	To email employees
Preconditions:	None
Scenarios:	<ul style="list-style-type: none">• User has feedback to share with management• User fills out feedback form• User hits "submit" and form is converted into an email and sent to an employee
Exceptions:	None

Get Quote

Actors:	User Employees
Goal:	To get a quote to pay for landscaping
Preconditions:	None
Scenarios:	<ul style="list-style-type: none">• Dimensions of yard are known• Dimensions are input into online form• Image of yard is uploaded• User's email submitted• User hits "submit"• User is shown a rough estimation of the cost to landscape that yard.• Employees receive an email of the image, estimation, and the email of the person who submitted the form.
Exceptions:	None

Post Comment

Actors:	User Database
Goal:	To post a comment
Preconditions:	<ul style="list-style-type: none"> Property that the user is posting a comment on exists
Scenarios:	<ul style="list-style-type: none"> User finds a property that interests them User writes a comment about the property in on website form User submits the form, comment is now publicly available
Exceptions:	<ul style="list-style-type: none"> Property is not in the database

Sort Comments

Actors:	User
Goal:	Sort comments by an attribute of the comments
Preconditions:	<ul style="list-style-type: none"> User must have selected a property This property must exist in the database
Scenarios:	<ul style="list-style-type: none"> User sorts by date posted User sorts by upvotes
Exceptions:	<ul style="list-style-type: none"> The targeted property does not have any comments

Upvote Comment

Actors:	User Database
Goal:	Upvote a comment that a user likes
Preconditions:	<ul style="list-style-type: none"> The comment must exist The user must click the button to upvote the comment
Scenarios:	<ul style="list-style-type: none"> User clicks upvote button to like the comment The comment contains an upvote button that can be highlighted over.
Exceptions:	<ul style="list-style-type: none"> The user already upvoted the comment The comment is not in the database There are no comments in the database

View Comments

Actors:	User Database
---------	------------------

Goal:	Shows a list of all the comments posted on a single landscape
Preconditions:	<ul style="list-style-type: none"> • Comment must be included within a landscape • Comment can show the amount of upvotes • Comments can be in different orders if the user specifies.
Scenarios:	<ul style="list-style-type: none"> • User views landscape and scrolls down the page to see the comments
Exceptions:	<ul style="list-style-type: none"> • No comments exist. • The user is attempting to view comments on a landscape that doesn't exist.

Upvote Landscapes

Actors:	User Database
Goal:	User upvotes a landscape they like in the list
Preconditions:	<ul style="list-style-type: none"> • The user must be viewing a landscape before upvoting it. • The user can't upvote a landscape twice. • The landscape being liked must be in the landscape list
Scenarios:	<ul style="list-style-type: none"> • The user is viewing the landscape and clicks on the upvote button for the landscape.
Exceptions:	<ul style="list-style-type: none"> • There are no landscapes in the page

Sort Landscapes

Actors:	User Database
Goal:	The user is able to order the landscapes by upvote amount or in alphabetical order.
Preconditions:	<ul style="list-style-type: none"> • There must be more than one landscape in the database
Scenarios:	<ul style="list-style-type: none"> • The user wants to see the most upvoted landscape, so they click on the top of the page to change the landscape ordering when viewing the catalog. • The user wants to organize the available landscapes based on alphabetical ordering.
Exceptions:	<ul style="list-style-type: none"> • There are no landscapes.

View Landscape List

Actors:	User
---------	------

	Database
Goal:	To view list of landscapes
Preconditions:	<ul style="list-style-type: none"> • The landscape list has to exist in order for user to view it • All landscapes are listed
Scenarios:	<ul style="list-style-type: none"> • User wants to see the different types of landscapes • Error message will be displayed if the list is empty and no landscapes can be viewed • If the list has only one landscape, then only one landscape will be viewed • If the list has multiple landscapes, then list of landscapes will be displayed
Exceptions:	<ul style="list-style-type: none"> • Not all landscapes are listed • No landscapes are listed • User can't find landscape in the list because it is not listed

View Single Landscape

Actors:	User Database
Goal:	To view a single landscape
Preconditions:	<ul style="list-style-type: none"> • The landscape has to exist in order to be viewed • Landscape exists in database
Scenarios:	<ul style="list-style-type: none"> • User finds a landscape that they like • Provides details about landscape
Exceptions:	<ul style="list-style-type: none"> • Landscape is not in the database

Post Landscape

Actors:	User Database
Goal:	To post an image and description of a landscape
Preconditions:	None
Scenarios:	<ul style="list-style-type: none"> • User wants feedback on the landscaping they just got done • User wants to show an image of their yard before landscaping, might want suggestions of what to do with it. • User fills out a landscape form, including a description, title, and image. • User hits "submit", data sent to database.
Exceptions:	None

