Listify Software Design Specification

Contributors

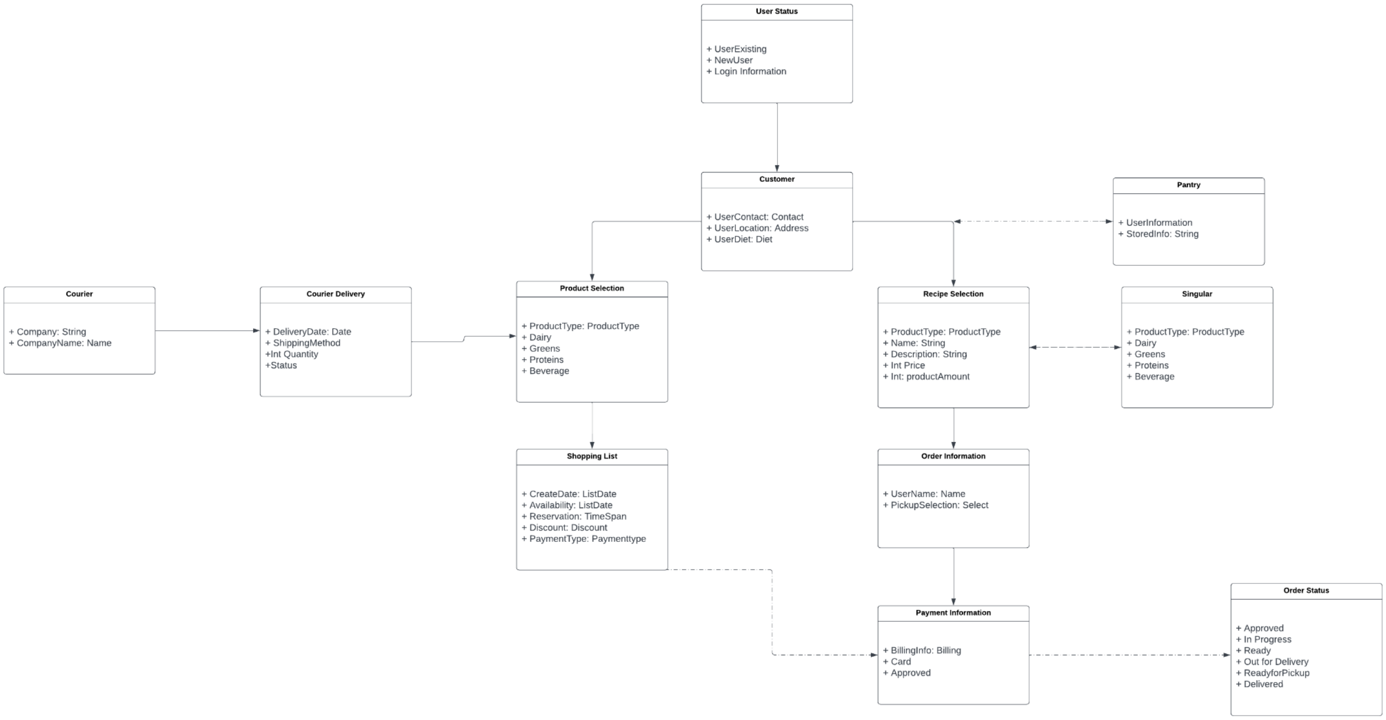
- Jaydon Eppinette
- Joshua Constine
- Agustin Munoz
- Marc Rodriguez

System Description

Listify is the best way to organize the mundane task of grocery shopping. Listify empowers our users by simplifying the process of planning their weekly meals. Using our extensive database of recipes, we quickly compile a list of the required ingredients. Users can also manually add items to the list. Additionally, Listify enables users to share their shopping list via a quick text message.

With Listify, grocery shopping becomes efficient and hassle-free.

UML Diagram



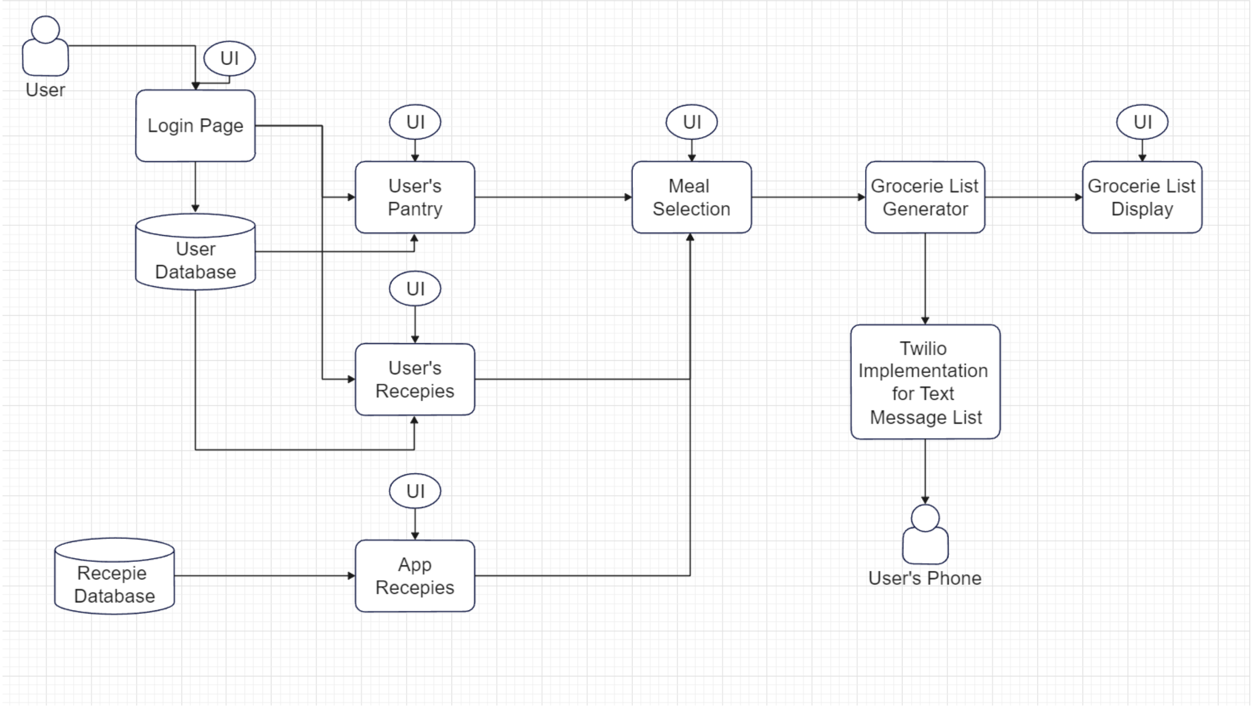
General Description of Classes

Classes	Descriptions	Attributes	Operations
User Status	*This class pertains to the general status of the user. Holds the information in relation to the registered account.*	*UserExisting and NewUser pertain to the status of a user in regards to their account. This is tied to Login Information.*	*The operations describe whether the user has existing information or needs to register new data.*
Customer	*Holds general customer information in relation to the previous class.*	*User: Contact, Location, and Diet are all key to user information and the specific needs of the customer. The diet portion is relative.*	*Operations include the input of information related to personal clientele data.*
Pantry	*Stores information in relation to the user's current items in storage.*	*UserInformation and StoredInfo pertain to current data related to previous client input.*	*The operation is performed in relation to currently held data from previous input.*
Recipe Selection	*Content within this class is in relation to options users are allotted to recipe information.*	*ProductType, Name, Description, and related integers are attributed to certain recipes and their cost.*	*The operations within this class hold data in relation to recipes available to the user and their cost.*
Singular	*Class holds information to single items that the user may add a la carte.*	*Attributed to single items and their categorization.*	*Operators are to hold single product information rather than bundled recipes.*
Order Information	*This class holds information in relation to general pick up.*	*UserName pertains to the name of the client and PickupSelection pertains to where the order is to be carried out from.*	*Limited to crucial information in terms of data relating to order status available to users.*
Payment Information	*Stores the payment information of customers and general clientele.*	*BillingInfo holds general billing information, Card holds type of*	*Data in relation to billing and status of payment.*

		payment in relation to cards, and Approved signifies whether the transaction has successfully gone through.	
Order Status	*Updated class in relation to all order information. Should be updated in relation to order status.*	*The attributes in this class hold a general step by step process in terms of the order's timing and status available to the client.*	*Data operators in relation to status of the order at hand.*
Product Selection	*A class containing general a la carte items sorted by type of product.*	*Similar to the attributes found in the Singular class given categorization and a la carte items.*	*Operators are in relation to types of singular products and their categorizations*
Shopping List	*Contains user's selected products.*	*CreateDate: Holds date to current standards* *Availability: Dictates the item's ability to be purchased.* *Reservation: Serves as a time of purchase.* *Discount: Applicable coupon or discount via promotion* *PaymentType: Form of payment*	*Operations are attributed to the availability and status of the user's intended products to purchase.*
Courrier	*Dependent on the courier delivering.*	*Company and CompanyName relate to specific couriers.*	*Operators in relation to certain or specific courrier.*
Courier Delivery	*Holds courier delivery information*	*DeliveryDate: Date of delivery of items* *ShippingMethod pertains to method of shipment* *Integer of Quantity dictates amount of certain product* *Status is for*	*Operators are in tune with specific courier delivery information.*

		estimation of delivery.	

Software Architecture Diagram



Jaydon Eppinette, Joshua Constine, Agustin Munoz, Marc Rodriguez

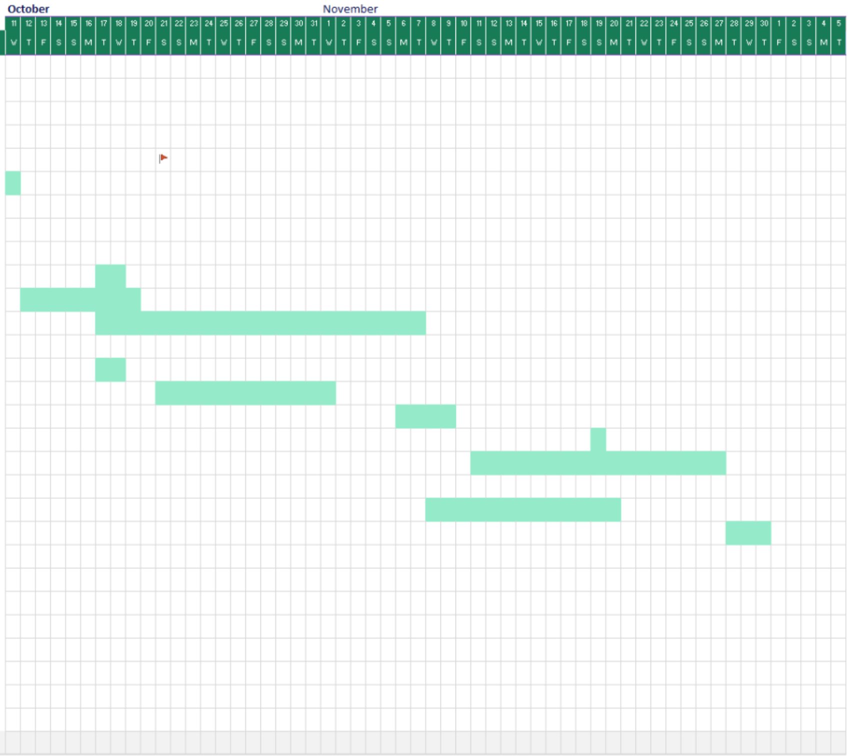
Project start date: 9/28/2023

Milestone marker: 1 🚩

Scrolling increment: 13

Milestone description	Assigned to	Progress	Start	Days
Analysis				
Requirement Meetings	Jaydon; Joshua; Agustin; Marc	25%	10/1/2023	3
Communication with Stakeholders	Jaydon; Joshua; Agustin; Marc		10/6/2023	1
Feasibility Meeting	Jaydon; Joshua; Agustin; Marc	50%	9/28/2023	10
Tools and Tech Selection	Jaydon; Joshua; Agustin; Marc		10/21/2023	1
Analysis Finished		10%	10/7/2023	6
Design				
Create Design Specifications	Jaydon; Joshua; Agustin; Marc	60%	10/8/2023	1
Software Design	Agustin; Marc	50%	10/8/2023	4
Interface Design	Jaydon; Joshua;	33%	10/16/2023	4
Class Design	Jaydon; Joshua; Agustin; Marc		10/11/2023	10
Design Finished			10/16/2023	24
Development				19
Coding Implementation	Jaydon; Joshua; Agustin; Marc		10/16/2023	4
Code Review	Jaydon; Joshua; Agustin; Marc		10/20/2023	14
Performance Optimization	Agustin; Marc		11/5/2023	6
Security Checks	Jaydon; Joshua		11/18/2023	3
Refactoring	Jaydon; Joshua; Agustin; Marc		11/10/2023	19
Testing				
Test Case Creation	Jaydon; Joshua; Agustin; Marc		11/7/2023	15
System Test Execution	Jaydon; Joshua; Agustin; Marc		11/27/2023	5
Fix Issues/bugs	Jaydon; Joshua; Agustin; Marc		12/20/2023	5
Review Code Performance	Jaydon; Joshua; Agustin; Marc		1/3/2024	1
Implementation			1/5/2024	15
On-Site Installation	Jaydon		1/6/2024	1
Support Plan for the System	Agustin		1/7/2024	14
Completion			1/21/2024	9
System Maintenance	Joshua		1/30/2024	9
Evaluation	Marc		1/30/2024	9

To add more data, Insert new rows ABOVE this one


October
November
W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T