



Software Design Specification

*by Chaz Gabelman, Eric Soto,
Gerardo Reza, and Luke Patterson
(Group 9)*

Software for a Cafe

Requirements:

- Contains food/drink items
- Ability to order for pickup
 - Customization
 - Defer order pickup
- Rewards system
 - \$5 off for every \$50 you spend
- Users
 - Customer
 - Database tracks spending, contains user login info
 - Admin
 - Database verifies credentials, create/edit/remove promotions
- Display promotions
- Choose location
- Accept and process payment securely (use 3rd party service) • Notify if item is in stock/out of stock

Classes:

- Store [1..1] (one store has one menu?)
 - So composition with filled in dot on the store class, coming from menu class
 - String: location
 - String[]: promotions
 - Array[string: ingredientName][int: quantity]: stock
 - removeOutOfStock():boolean
 - addNewFoodItem(menuItem: newItem):boolean
 - addNewDrinkItem(menuItem: newItem):boolean
 - displayPromotions():void
- Menu [1..*] (one menu per store can have many menu items)
 - Composition relationship to menu item
 - menuItems[]: foodItems
 - menuItems[]: drinkItems
 - removeItemsContaining(String: ingredientName):boolean
 - displayMenu(menuItems[]: specificMenu):void
- menuItem [*..*] (many menuItems can have many ingredients)
 - Composition relationship to ingredient class
 - makeMenuItem(string:name, double:price, ingredient[]:ingredients)
 - Ingredients[]:ingredients

- String: name
- Double: price
- getIngredients(): ingredients[]
- addIngredient(Ingredient: newIngredient):boolean
- removeIngredient(String: ingredientName):boolean
- displayIngredients():void
- Ingredient
 - makeIngredient(String: name, int: quantity, int: caloriesPerUnit)
 - String: name
 - Int: quantity (1=light vs. 2=normal vs. 3=extra)
 - Int: caloriesPerUnit
 - Double: pricePerUnit
 - getTotalCalories():Int
 - totalPrice():Double

Should we add a cart class? Contains menu items, generates total price of the cart and then has a checkout function?

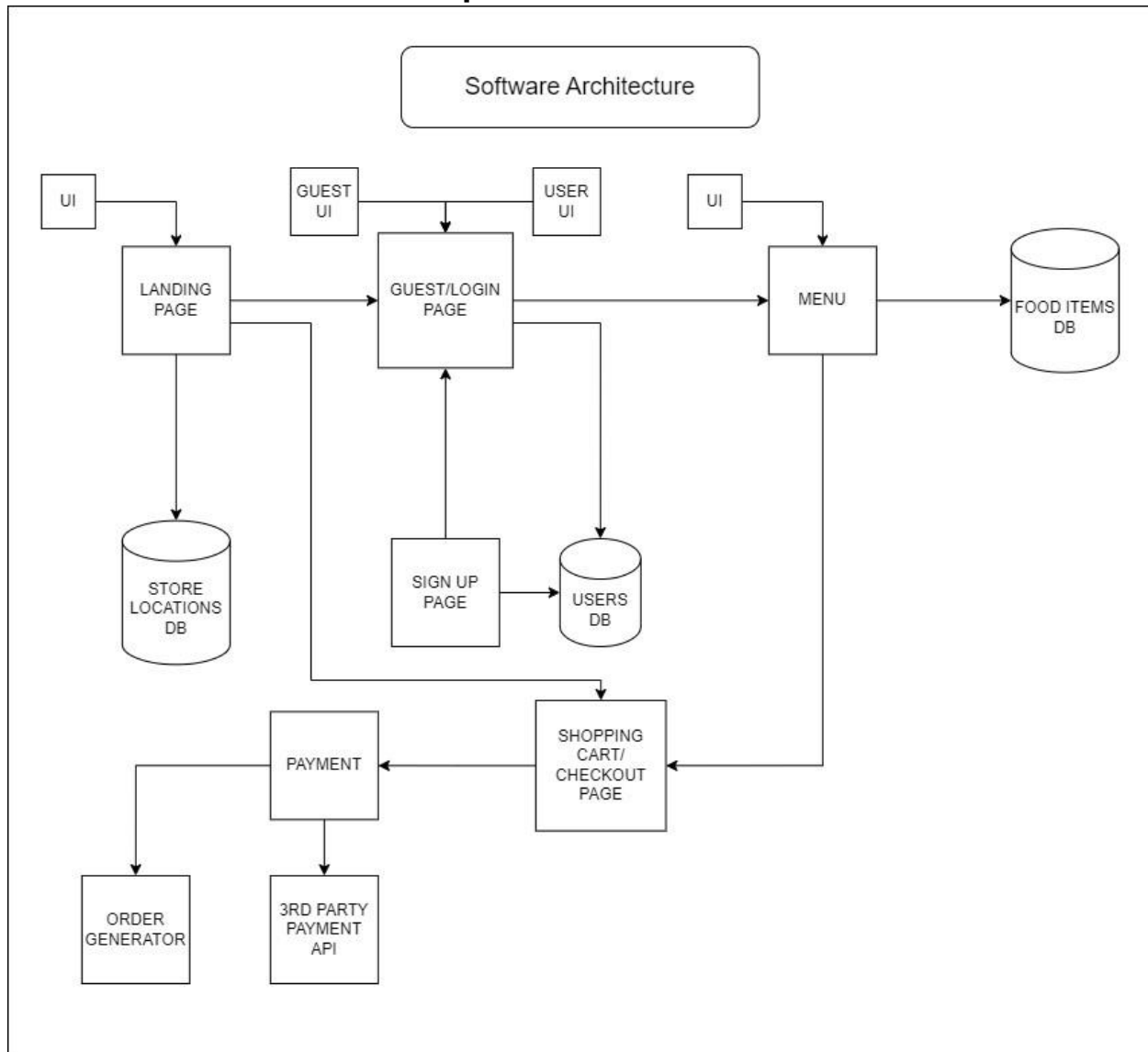
- User (Guest)
 - String: cardNumber
 - makePayment(String: cardNumber):boolean
- Rewards member subclass
 - Int: rewardPoints
 - String : "username"
 - String : "password"
 - Array[string, int] : [favorite item, favorite item price]
 - Array[string, int]: [previous order item, previous order price]
 - login(username, password):boolean

Pages and Databases:

- User DB
- Landing Page
- Food Items DB
- Shopping cart / Checkout Page

- Store Locations DB
- Payment Page

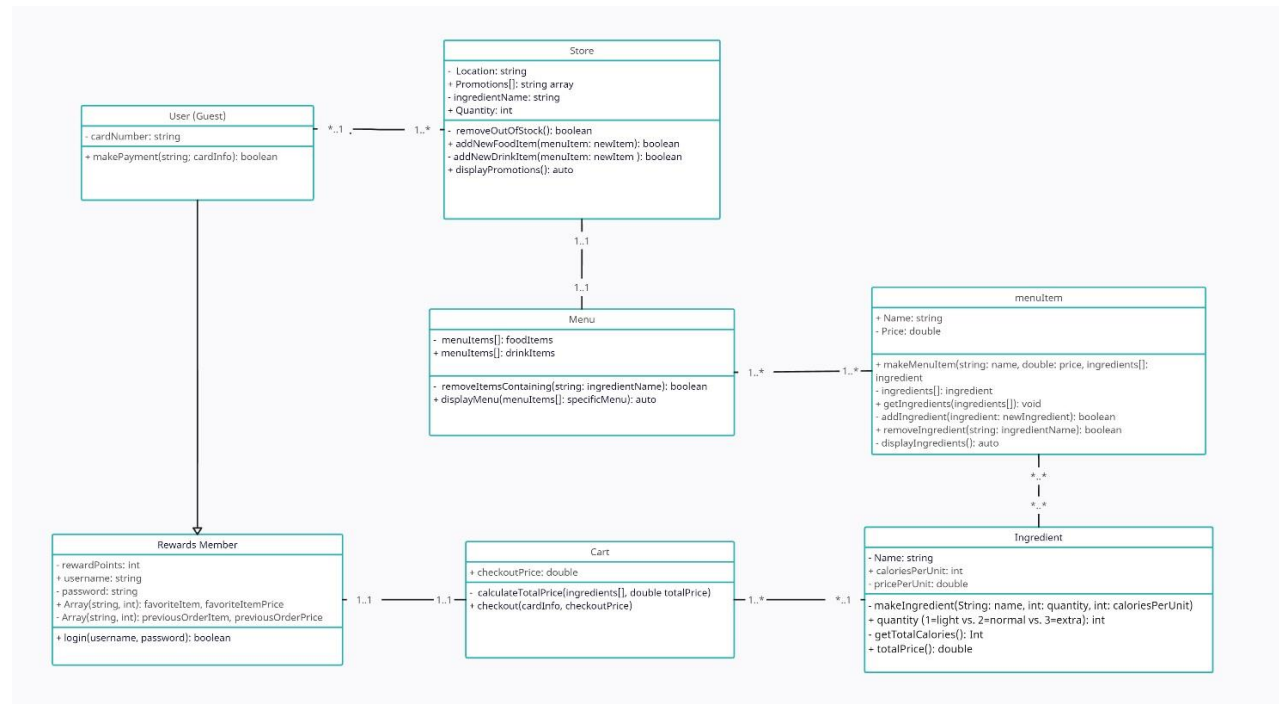
Software Architecture / Component Connections:



- The **landing page** is connected to a **store location DB** and **guest/login page**.
- The **guest/login page** is connected to **sign up page** and **users DB**.
- The **menu** connects to **food items DB** and to **shopping cart/checkout page** since you are adding menu items.
- The **landing page** would be connected to a **shopping cart/checkout page** if the user decides to access it from the home page.
- The **shopping cart/checkout page** would be connected to a **payment page** where the guest/user enters or uses saved payment information.
- The **payment page** connects to a **third party payment API** for processing payments.

- The **payment page** would also be connected to an **order generator** that sends an email receipt and order number.

UML Class Diagram:



Development Plan + Timeline: