COFFEE MACHINE SYSTEM

MPP COURSE PROJECT

APRIL 22, 2017

MAHARISHI UNIVERSITY OF MANAGEMENT

PREPARED BY: GROUP 8:- MIERAF ABOYE & MAEREG SORRI

Table of Contents

Coffee Machine System

Abstract

Among the many services given in different public service businesses is the hot beverage Service. Although the service providing technique for many other respective services have been digitized and modernized, coffee drinks Serving is still in its manual stage. In this modern age, the use of Technology not only eases the activities of life, it is also the biggest tool that service providers and producers use to give their businesses high competitive edge. Customers and consumers are more likely to get services from providers with latest service technologies. This system aims to provide an advanced and modern technology for several types of coffee drinks making, serving and payment processing. This project also aims to automate the coffee making process which also minimizing labor work and increase billing security. This system is analyzed based on the system requirements and different models are used to describe the analysis. UML model is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems. class diagrams and Sequence diagrams are used to put the analysis into Models. The functionalities of the system are generated based on the main activities of getting a coffee drink in a coffee store and the labor based service given in stores. It brings modern technology to create this interactions and actions into a computer system and give out better performance. JavaFX technology is used to design the User interface and Java Object Oriented Programming is applied for backend programming. This technologies gives the system the flexibility to grow more in further enhancements and for the system to reflect the existing object interactions in the real-world to a corresponding software interactions. Testing of the system is done at the implementation phase by subjecting the system to different inputs and analyzing the outputs.

In conclusion, this system brings the Coffee Service business to the next level as it provides easy, affordable and Modern way of service providing. The technologies used to develop this system are the leading technologies for vast development making further enhancement of this system much more realistic.

PART ONE

Introduction

The Project titled as Coffee Machine System is a Coffee drink generating system that is designed for providing coffee drinks for customers. The project "Coffee Machine system" is developed in Java which mainly focus on operations in coffee sales like, providing types of coffee drinks, ingredient processing, providing extras and sugar, accepting payment and returning change.

Coffee Machine System is a desktop application that performs the activities of a Coffee Machine System software. This Software produces various kinds of coffee drinks per user requests. It generates a drink among the available lists of coffees drinks a user is presented with. The system makes different mixes to produces the requested drink. For example, it takes some amount of coffee and some amount of milk to deliver a latte for the user. The payment process accepts payment while ordering, will deduct charge per the drink and extras and give out change or start another order by the end of dispensing per user interest. The user should make a cup size choice. The system also alerts the owner to refill the containers when the amounts of ingredients in the containers reduce to some level. Designing this system will be elaborated using UML class diagram and Sequence Diagrams while Implementation of the project will be done with Java and uses object oriented concepts like polymorphism, Inheritance, Relationships, Factory Methods, Abstract classes, Interfaces and JavaFX for UI.

Objectives

In this time of age where computers and smart systems are making most of our day to day activities easier and less time consuming, It is the logical to think of developing a system that makes our lives even more easier. The main Objective of this project is developing a software that has the capacity of performing coffee serving and making functionalities as used by a customer in a coffee shop or in any public place where it is easier for users to serve themselves without going through the usual hustles most likely to be faced while getting a coffee drink. Lists of Objectives are listed below.

- To provide digital coffee self-serve option
- To reduce the hustle to get or make coffee for customers
- To reduce labor work
- To automate Coffee making
- To increase billing Security

Technology

The Main technologies used for the development of this System are as follows.

Front end Development:

- JavaFX

Backend Development:

- Java

Database:

- MySQL

Developmental Approach

The Water Fall Developmental Approach is used for this system.

The waterfall model is a sequential (non-iterative) design process, used in software development processes, in which progress is flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance. The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. This Approach is used for the development of this project.

PART TWO

Analysis - System Specifications

Functional Requirement

Based on the needs of a customer and administrator, this system has the following functional Requirements.

Choices of Coffee mixes

The system is required to provide choices of Coffee drinks for the Customer to choose the type of drinks he/she wants. This Choices should be viewed when a person uses the software as a Regular user. There is no need of Authentication for a regular user to access the system.

Payment Processing

The System is Required to get payment from the Customer, calculate price of Coffee Choice deduct price from payment and return change. It needs to check if the amount the customer inserted is enough for their choice of drink and extra choices. If enough balance is not found it will display message asking the customer to make other choices or cancel order.

Ingredient Processing

After the choice of Coffee is made by the customer, the System is required to get the recipe of the drink, add ingredients add required extras and sugar and return the drink.

Extra Choices

This feature should be provided for the customer to choose the type of extras they want along with the coffee drink of their choice.

Cancel Order/Return Money

They system must return the Money amount it accepted when the Customer decides to cancel its order. The customer will be provided the canceling order option after choosing an order and before processing to the making of the drink. once everything is checked right and the process of making coffee is started the cancel order option will not be available for the customer.

Choose Size

The system is required to get cup sizes of choice from the customer and make drinks per the size choice.

Refilling Containers

Whenever the containers of ingredients get bellow some level put as mark, the system is required to send notification of the Admin user for a refill. The system will refill with the orders from admin user.

Message Display

They system will have a message displaying feature to be used by different scenarios.

Non-Functional Requirements:

- Performance
- Speed
- Memory size
- User friendly Interface

PART THREE

Design approach

Introduction to Design

Design is the first stage that any development takes after gathering all types of requirements and before starting Implementation. Software and hardware requirements are inputs for the designing process as it will be designed in a way that optimizes both requirements. The functional and non-functional requirements specify how the backend development should be designed in-order for the system to give the needed output. One of the tool we use to develop the design is UML Diagrams. Class diagram is a UML diagram that specified the possible classes and their relationship for the backend development and Sequence diagram shows the message calls between classes and how a functionality will run for different scenarios. Both Sequence and Class diagram for this project are put and described below.

UML Diagrams

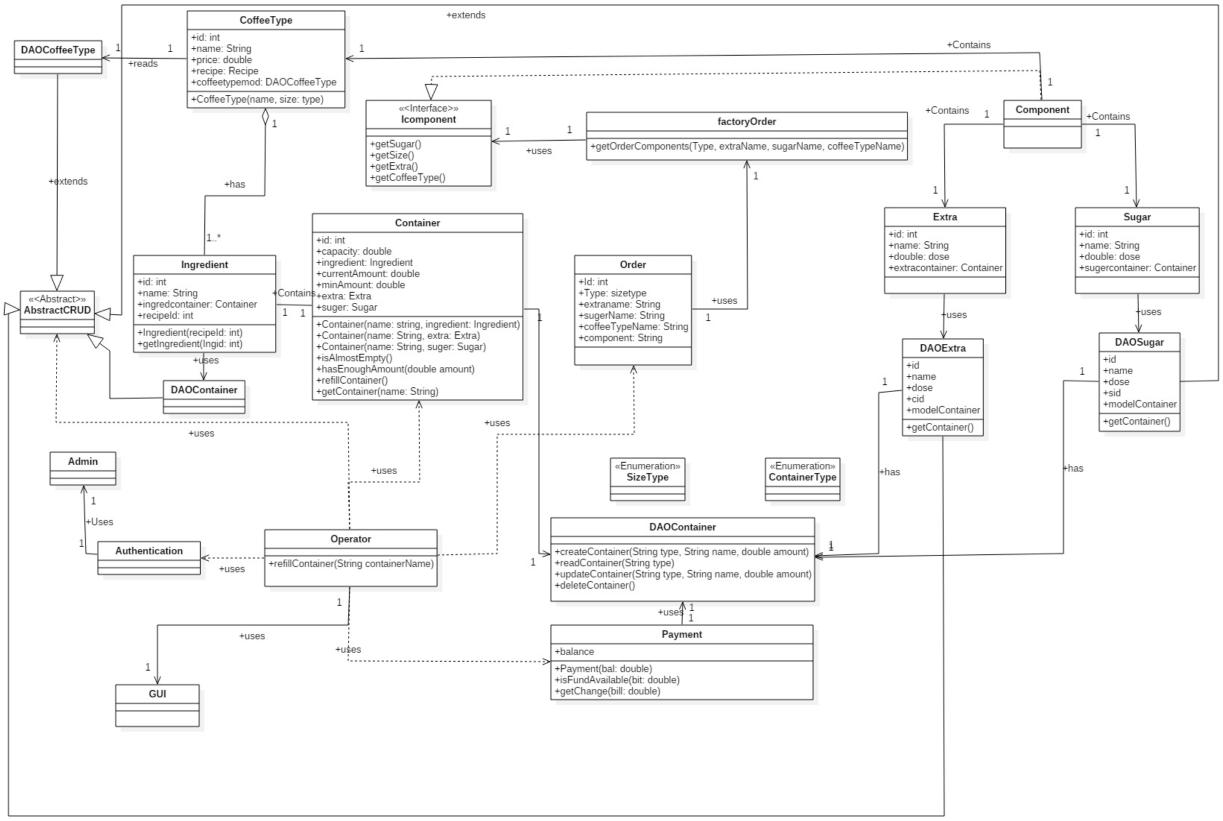
The Unified Modeling Language is a standard visual modeling language intended to be used for

modeling business and similar processes, analysis, design, and implementation of software-based systems. UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

UML can be applied to diverse application domains (e.g., banking, finance, internet, aerospace, healthcare, etc.) It can be used with all major object and component software development methods and for various implementation platforms.

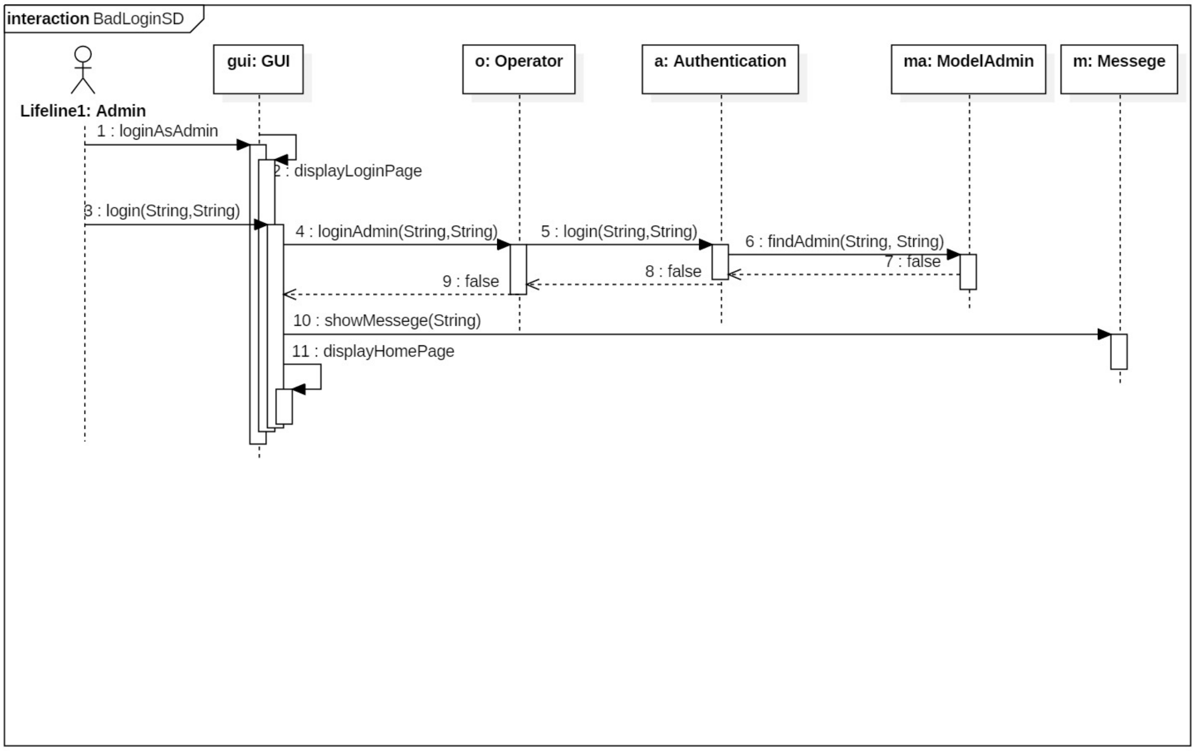
Coffee Machine System functionalities are further described and Abstractly represented by the following UML diagrams. The software objects are shown in the class diagram with their attributes, operations and relationships whereas the following several Sequence diagrams shows the interaction of objects by sending messages among each other for different scenarios expected to happen on the use of this Machine.

Class Diagram

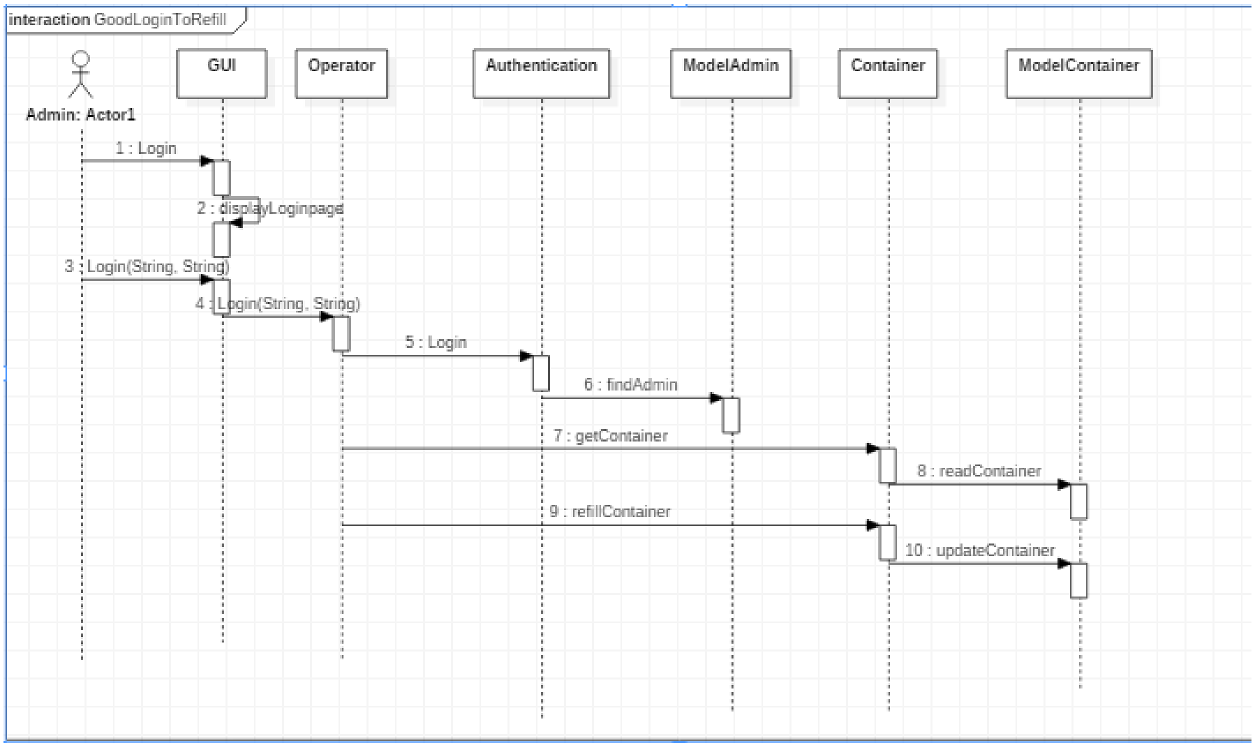


Sequence Diagrams

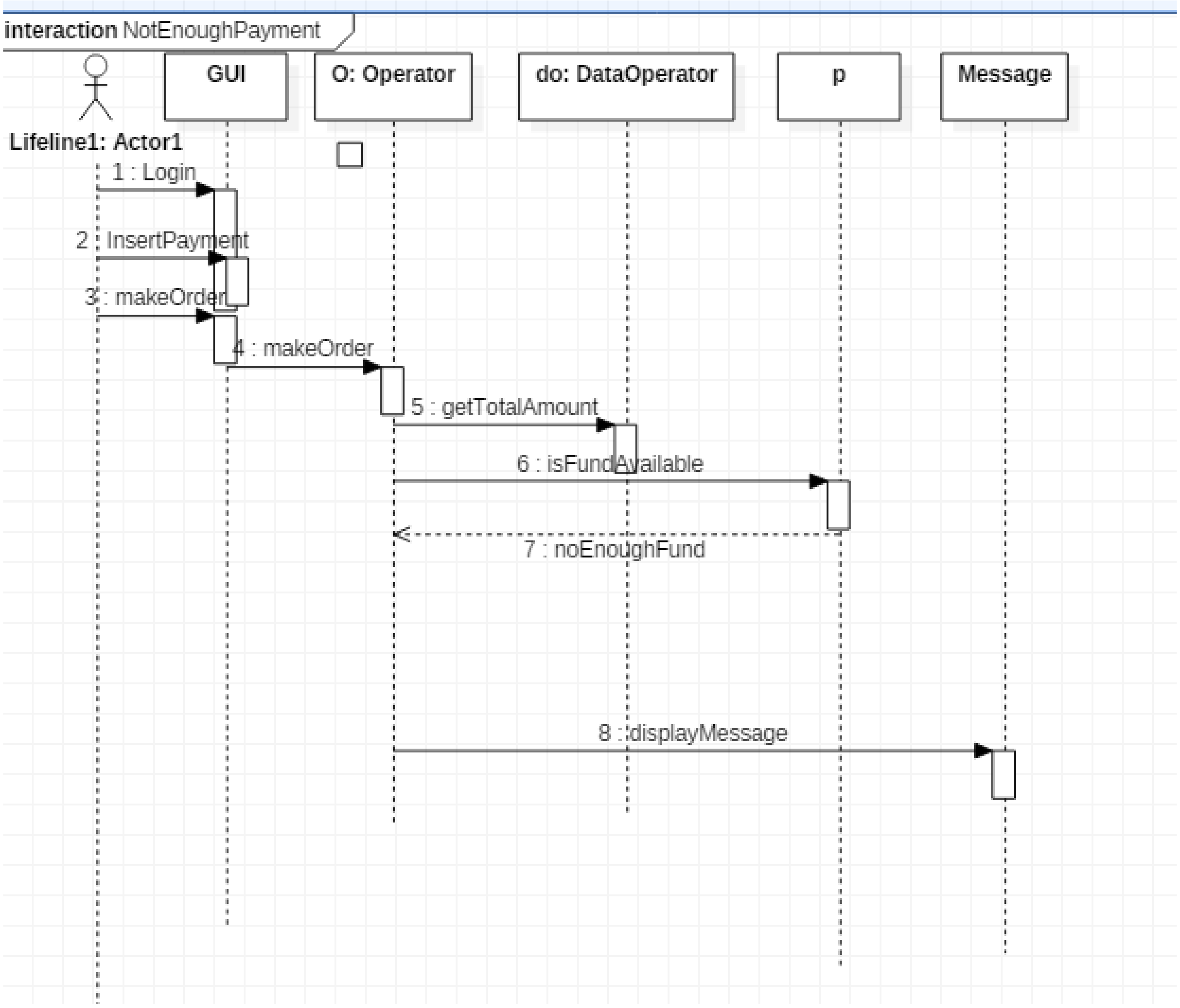
Bad Login scenario.



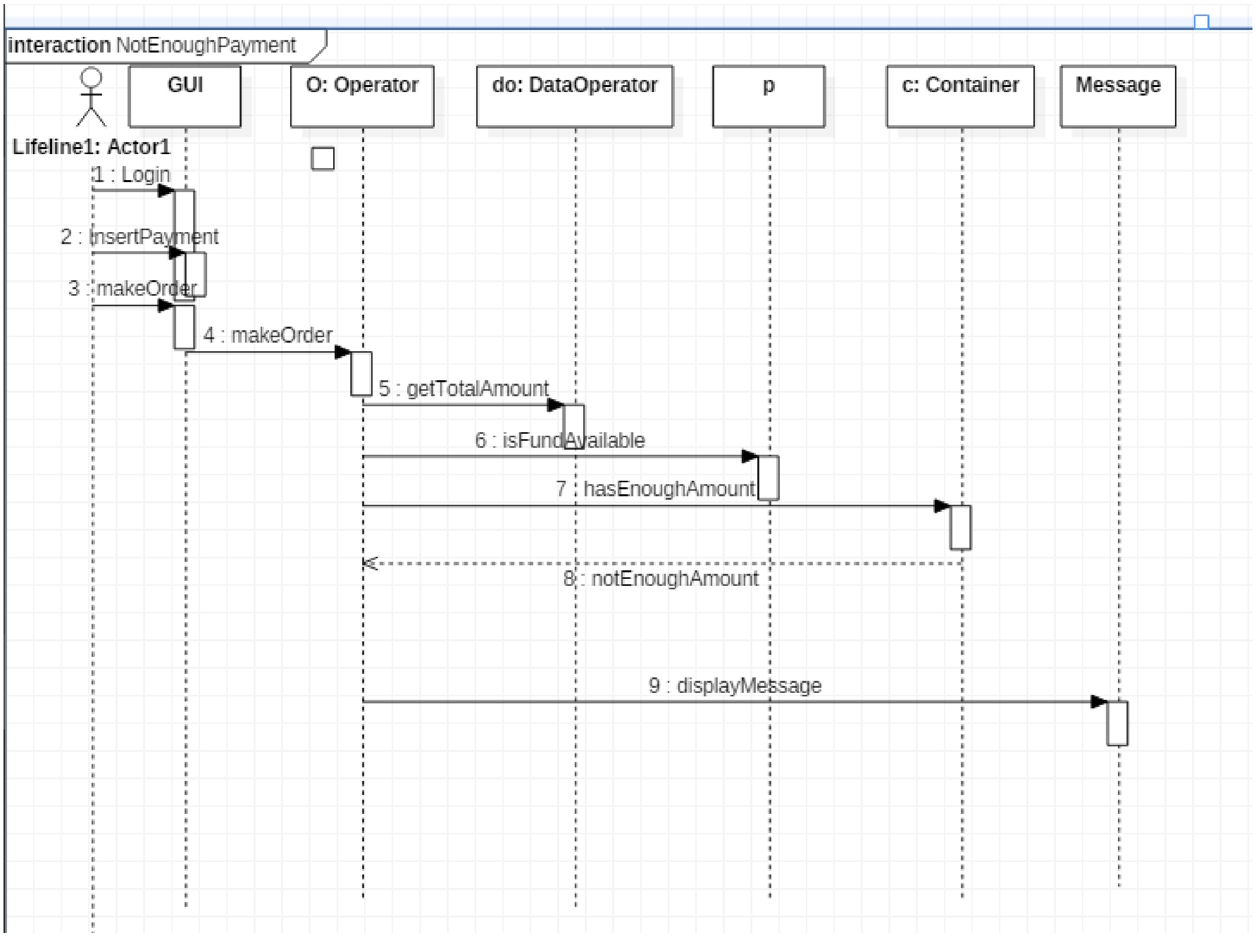
Good Admin Login Scenario



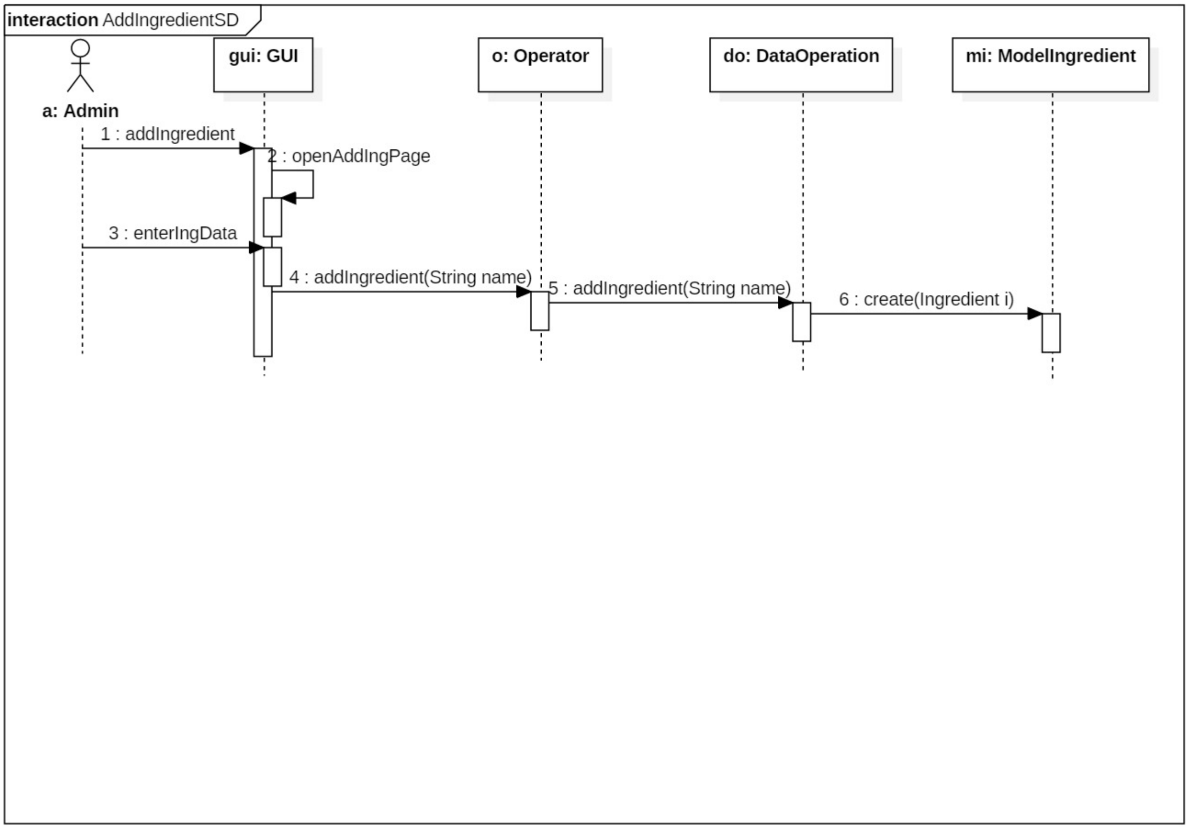
Insufficient Payment Scenario



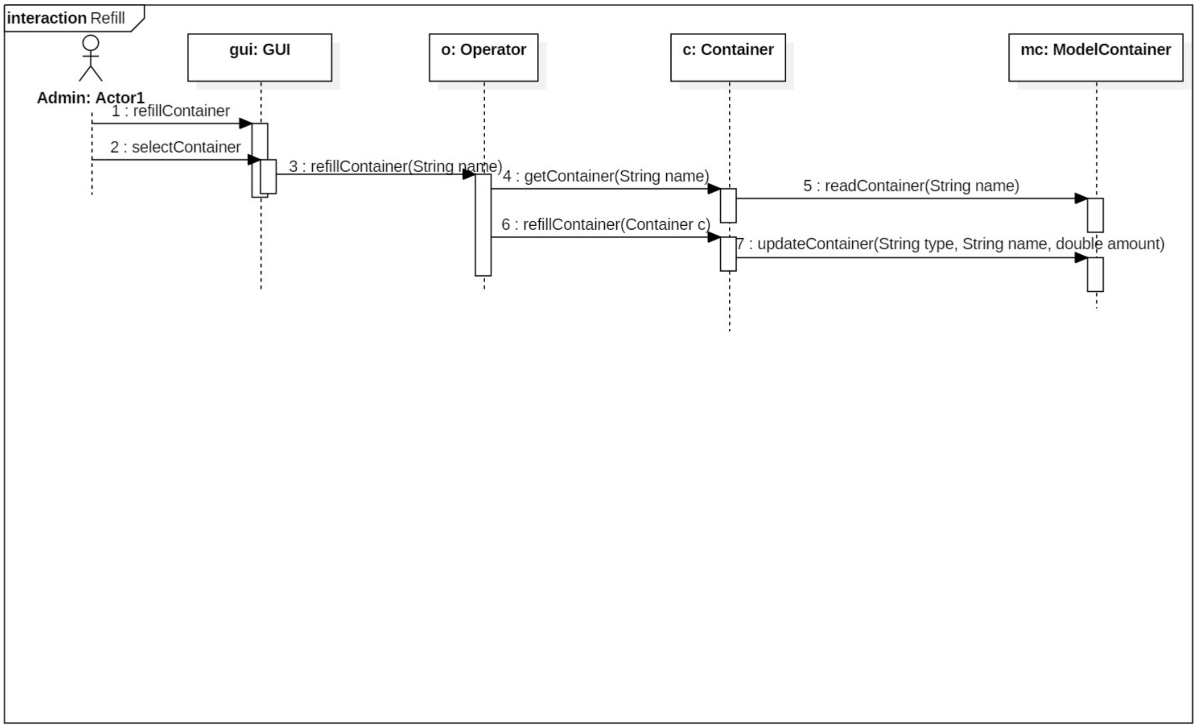
Insufficient Amount in Container Scenario



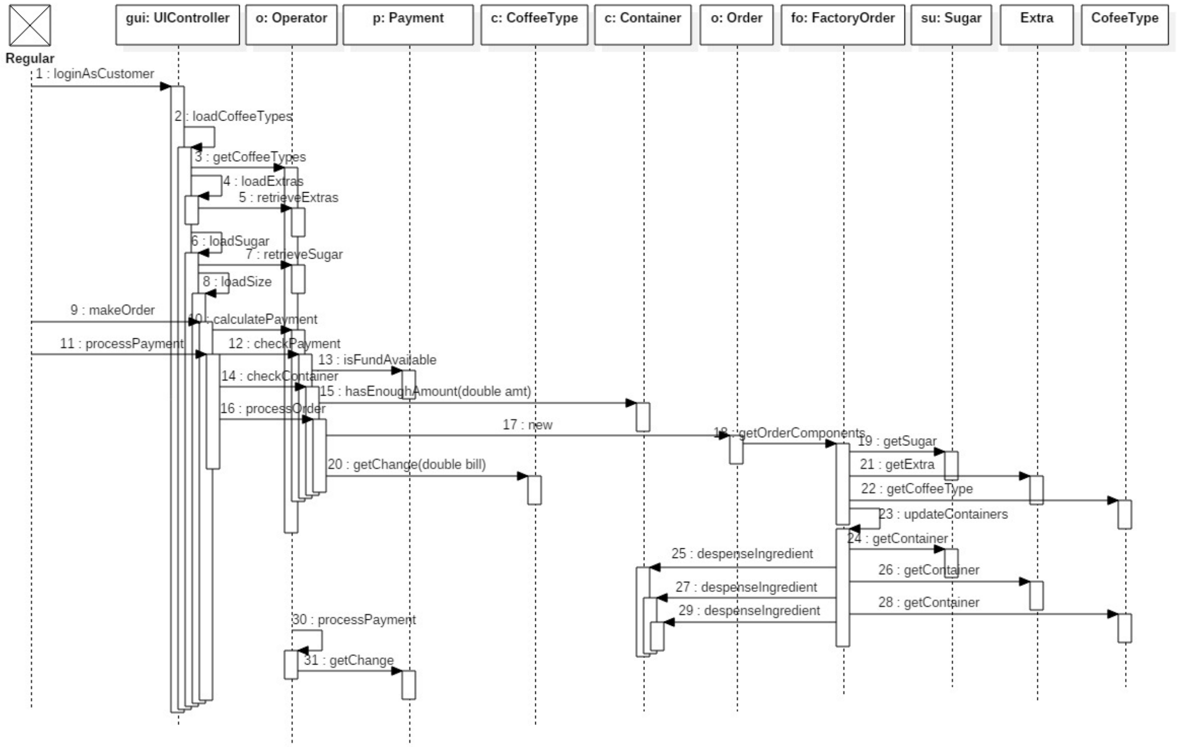
Adding Ingredient Scenario



Refilling Scenario


interaction Refill
gui: GUI
o: Operator
c: Container
mc: ModelContainer
Admin: Actor1
1 : refillContainer
2 : selectContainer
3 : refillContainer(String name)
4 : getContainer(String name)
5 : readContainer(String name)
6 : refillContainer(Container c)
7 : updateContainer(String type, String name, double amount)

Make an Order Scenario



PART FOUR

Implementation

Overview of technology used

Front end Technology

JavaFX:

JavaFX is a set of graphics and media packages that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms. This project uses JavaFX technology to design the user interface. Since the JavaFX library is written as a Java API, JavaFX application code can reference APIs from any Java library. For example, JavaFX applications can use Java API libraries to access native system capabilities and connect to server-based middleware applications.

The look and feel of JavaFX applications can be customized. Cascading Style Sheets (CSS) separate appearance and style from implementation so that developers can concentrate on coding. Graphic designers can easily customize the appearance and style of the application through the CSS. If you have a web design background, or if you would like to separate the user interface (UI) and the back-end logic, then you can develop the presentation aspects of the UI in the FXML scripting language and use Java code for the application logic. If you prefer to design UIs without writing code, then use JavaFX Scene Builder. As you design the UI, Scene Builder creates FXML markup that can be ported to an Integrated Development Environment (IDE) so that developers can add the business logic.

Backend Technology

Java:

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. Object Oriented concepts like Inheritance and Polymorphism are applied to the development of this project in java.

Database

MySQL:

MySQL is an open source relational database management system. Information in a MySQL database is stored in the form of related tables. MySQL databases are typically used for web application development (often accessed using PHP).

A MySQL database can be accessed (queried) directly using; C, C++, Eiffel, Java, Perl, PHP and Python computer languages. There is also third party support for connection in a Windows environment.

MySQL databases are queried using a subset of the standard Structured Query Language (SQL) commands.

MySQL is a principal component of the LAMP open-source web application software stack.

Concepts and Techniques

3-tier Architecture: this System is Implemented involving a 3-tier architecture. By splitting the solution into 3 logical layers - presentation, business logic and data access, this makes the development efforts better and increase maintainability.

Business layer Classes:

The business logic layer contains the domains that refers to the business entities that are involved. For example, we have included domain objects namely `Order`, Recipe and `Operator`.

Order:

The order domain class represents a type of coffee drink that the user selects. An order is created when a type of drink, extras and sugar is selected with a specific size. So an order has an extraName, sugarName, CoffeeTypeName and size as an attribute.

Payment

Payment is an object that is created when a customer or user insert money as a payment for their order. It is saved as balance and it has behaviors like isFundAvailable and getChange methods which checks the status of payments and which returns change for customers after getting a difference between the bill and balance.

Recipe

A recipe is a fixed signature of the mixture of ingredients to give out a coffee drink type. For example, A macchiato has some portion of milk and some portion of Coffee mixed in a certain way. This recipe is represented in this software by having list of Ingredients and id as an attribute. It contains getters and setters for recipe id and a getter for list of Ingredients

Operator:

Operator Class is one of the most functioning class in this system as it is the class that collects information from other classes and perform different actions. It's functions are listed bellow

- Get login information from user and from database compare the two
- It checks if enough money is found in balance
- It checks if Container has enough ingredient
- It makes order if all conditions are set right
- It gets change from payment class
- It dispenses coffee
- It makes message displaying possible.

Some of the Data Access layer and Model Layer Classes

DAO Admin

This Class is responsible for finding and Admin from the database with the given username and password.

DAO Coffee Type

This Class is responsible for creating, reading, updating and deleting (CRUD operations) coffee type from CoffeeType table in the database.

DAO Container

This class is responsible for creating, reading, updating and deleting (CRUD operations) Containers from the container table in the database.

DAO Ingredient

This class is responsible for the CRUD operations for Ingredient table in the database.

Component

A component is a collection of coffee type, sugar, extra and size. It is a type created when an order is generated from the operator class.

Model Authentication

Its main function is to check the validity of the username and password by using the admin class to fetch data from the database.

Model Coffee Type

A coffee type consists name, price, recipe, recipe ID and size, this class represents coffee type objects with these attributes, it can be created with by giving name and size. It has getters and setters for its attributes.

Model Container

There are three types of containers, one for ingredient, another for Extra and another for sugar, these containers have ID, capacity, minimum amount and current amount as common attributes. This class is useful to define and create all type of containers as it consists three construction. It also has methods to check if container is almost empty and if it has enough amount.

Model Extra

Extra is an additional drink on the coffee drink type ordered. This class represents an extra type with name and ID and dose as an attribute.

Model Ingredient

Ingredient has an ID and name and is the material used to make a drink. It has a constructor with name as a parameter.

Model sugar

This class defines a sugar object with id, name and dose. It is used to create a sugar object with name and dose parameters. It contains getters and setters for it's attributes.

Presentation Layer

This layer contains the Graphical User interface that the user interacts with. It is designed with JavaFX and contains the GUI class.