ANNA UNIVERSITY, CHENNAI
SOFTWARE ENGINEERING PROJECT: LOCKDOWN CATERER

Juanita J - 2018103544
Shruthi M - 2018103592

ABSTRACT

Food has always been one of the most prime indispensability in all walks of life. With the growing changes and advancements in the socio economic scenario, food service and catering has evolved into an expanding industry. The technological advancements have helped these industries in production of food to streamline their activities to be more efficient, improved safety and quality and less tiring for the people.There are a wide range of catering services and systems functioning across the globe, but we have picked up a new initiative:A lockdown catering system. Considering the present day state of affairs of the world, a lockdown catering system will provide a safe and efficient way of fueling people's ration needs.

SLDC MODEL SELECTION

Now, to aid in defining, analyzing and designing,building,testing,deploying and maintaining the system we have picked, we employ a Software Development Lifecycle(SLDC) for our system. SLDC consists of several distinct stages to design, define requirements and develop a high quality software to meet the customer and market requirements.
Among the six SDLC models for building a software, we adopt the Agile model.
This model has been around for a decade with great efficiency and also is the best fit for our Lockdown catering system since it is comparatively more interactive and requires more engagement from the customers.

Moving to the model development:
The Agile development process is composed of components such as:

1. Establish a few initial requirements
2. Analyse the requirements
3. Design
4. Prototype / Implementation
5. Testing
6. Deployment
7. Collect feedback on what’s been presented thus far
8. Establish new requirements for next sprint based on feedback & repeat the micro outcome cycles until you achieve the final desired product

WHY WE PREFER THE AGILE MODEL

To start with, the agile model requires very limited planning. Since the idea of lockdown catering system has emerged suddenly and has not existed previously, we follow the agile model as it allows the product owner to identify the requirements in an iterative way. The idea needs to be implemented in a short span of time and hence agile is the appropriate choice here where action and getting things done is more important than documentation and process.Agile model is usually preferred when new changes have to be implemented within a short period of time or when the ideas have to be evolved over a period of time.

The catering system has evolved significantly and the requirements were different during each phase of the lockdown (stringent lockdown phase1 (April-May), phase2 with ease of few curbs (June-July), phase3 major relaxation (From Aug)). We prefer the agile model since the end users' needs are ever changing in the dynamic business. The most efficient and effective method of conveying information is face-to-face interaction and the model mainly emphasizes on feedback. Similarly, feedback from customers is the most important to us as this helps to improve our deliverables in the next iteration.

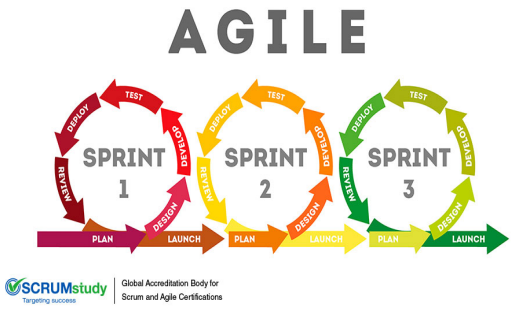
AGILE METHODOLOGIES

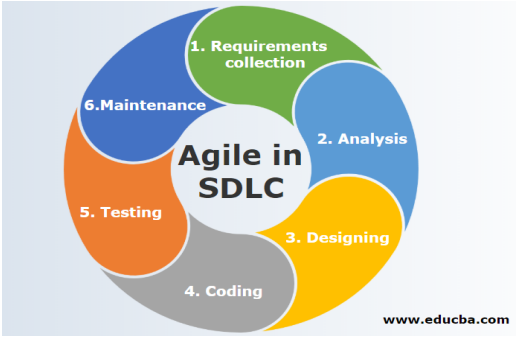
There are seven different widely-used agile methodologies (namely lean software development, Agile scrum methodology, Kanban, XP, Crystal, DSDM and FDD). Though they all share much of the same overarching philosophies and characteristics, from an implementation standpoint,however each has its own unique mix of practices and terminologies.Of the agile methodologies, we have employed the scrum agile methodology for our project.

SCRUM AGILE FRAMEWORK

The scrum is a lightweight process framework for agile development used to manage iterative and incremental projects.
We choose this methodology mainly because of its simplicity, proven productivity and ability to incorporate various overarching practices over the years. While other project management methods emphasize building an entire product in one iteration from start to finish, agile scrum methodology focuses on delivering several iterations of a product to provide stakeholders with

the highest business value in the least amount of time, thus making it more efficient.





ITERATION 1 (April-May)

The initial phase mainly focuses on a limited number of customers (say 20-30) and only within a specific area in the city.

REQUIREMENTS

These components include developing a vision for the high-level details, brainstorming for features of requirements, breaking down features into user stories, and defining detailed requirements. Identifying the target users and purpose is the most important aspect of the requirements phase.

Target users - In the initial phase of lockdown, the focus is on bachelors and old age homes.
Purpose - Provide homemade food with delivery service at nominal rates to cater to people's needs who find it difficult during the lockdown. The catering system takes order for 20-30 people in the initial phase where in the end users will be provided a menu for breakfast, lunch and dinner and the order needs to be booked in advance.
Features - The system needs to provide the following services: services booking in advance, ordering, goods procurement, sanitation, menu listing for each day, logistics for delivery service

ANALYSIS

The analysis includes both business analysis as well as software analysis.
Analysis on how the software must be maintained is done in this phase.
The system must be a double ended one that is used by both the end users (customers) as well as the caterers and admin. The number of customers we will be working for is essential for the scalability of the system. Separate modules for different segments of the system have to be

maintained. The system first needs to find a suitable database structure to store and retrieve order details.
The modules include:
(i) Goods procurement - Details on how and where and from whom the goods are procured is important. Also we need to keep track of inventory and the stock levels need to be updated.

(ii) Menu management - As this is the initial phase of lockdown and with the peak in covid cases, the system must come up with a menu that emphasizes on improving immunity.
Filters based on:

1. Veg / Non-veg
2. Age group
3. Location
4. Average cost

(iii) Food ordering - Here the system must be able to communicate with the caterers as and when the order is placed. Analysis on the process of ordering (including how advanced the order must be placed) is made.

(iv) Delivery service - The initial focus is manually handling the delivery service. List of workers who are willing to deliver should be maintained and those who are available at a time will be chosen for this. The system needs to keep track of covid hotspots everyday which is updated in the dashboard and this is accessible by those who handle delivery to ensure maximum safety for both the delivery person as well as the clients.

(v) Payment - System must get details of the cost and a simple calculation is performed. The customers will be able to pay via credit/debit cards or COD and bill is generated for the same.

DESIGN
Design a web and mobile-based application.
A dashboard is to be created with different features. A suitable architecture for frontend and backend is decided at this stage.
Scalability is an important issue and hence a suitable database management system is chosen where schema for menu, customer, ordering, payment is designed.

(i) Registration
First time users have to register with the system. They will be given a unique reference ID which can be used in the future. Initially, there are only 1-2 caterers registered with the system. The caterers also need to register.

DATABASE TABLE - CUSTOMER

Attributes: UniqueID(Primary Key), Name, Address, Email, Phone, Address

DATABASE TABLE - CATERER

Attributes: CatererID(Primary Key), Name, Address, Phone, Address

(ii) Goods Procurement and Inventory Management

Details of the inventory need to be stored. A management report has to be produced for the same along with the details of goods supplier.

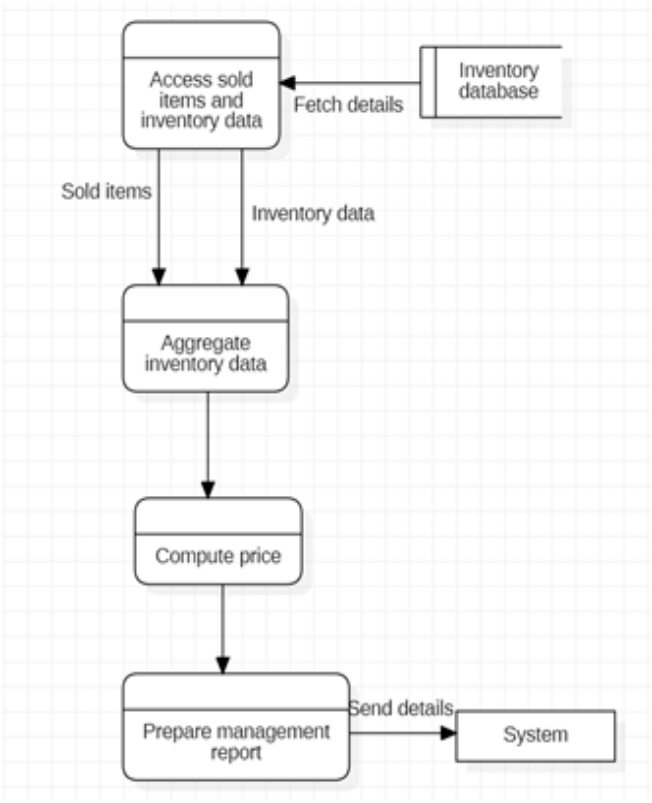
DATABASE TABLE - INVENTORY

Attributes: Ingredient, Stock level, Purchase date

DATABASE TABLE - GOODS SUPPLIER

Attributes: Items, cost, location

DFD DIAGRAM FOR GOODS PROCUREMENT



(iii) Menu Page

1. Different time slots are listed (breakfast, lunch, dinner)

2. Filter options based on veg/non-veg, age group are listed along with the photos and detailed recipe
3. The initial phase contains mostly text-based option with quantity and cost for each
4. Checkboxes for each options across different filters

DATABASE TABLE - MENU

Attributes: Item, Price, Description, Availability, Time (time of availability of the dish - Breakfast, Lunch & Dinner)

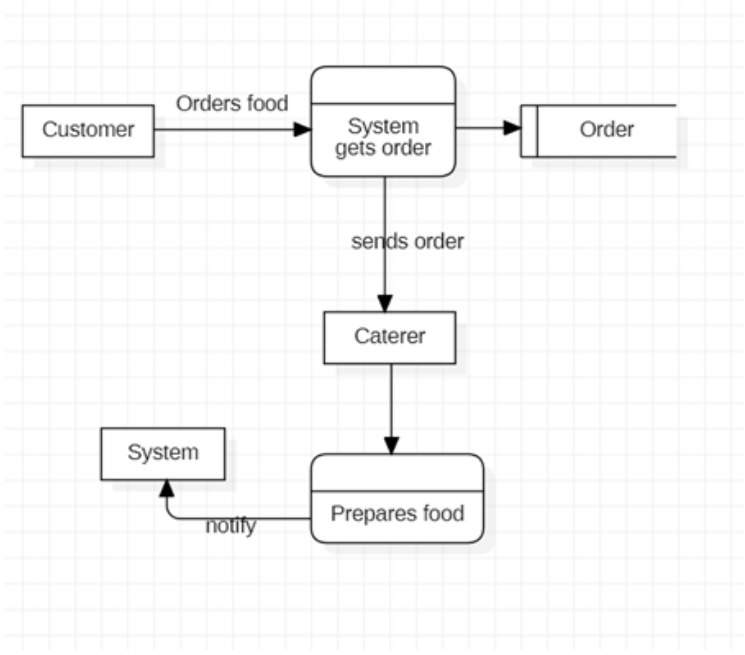
(iv) Online Ordering Page - Via call as well as online ordering option.

1. First time users must register. Others can login with the already existing one.
2. In the first iteration, there are both call as well as order online options as customers initially prefer to get a real time live opinion.
3. Have a cart feature that displays order in the side
4. Final order details page to be included

DATABASE TABLE - ORDER

Attributes: Order ID, Customer ID (foreign key), Item, Quantity

DFD DIAGRAM FOR ONLINE ORDERING



(v) Payment Page - Initial phase allows only COD and credit card transaction

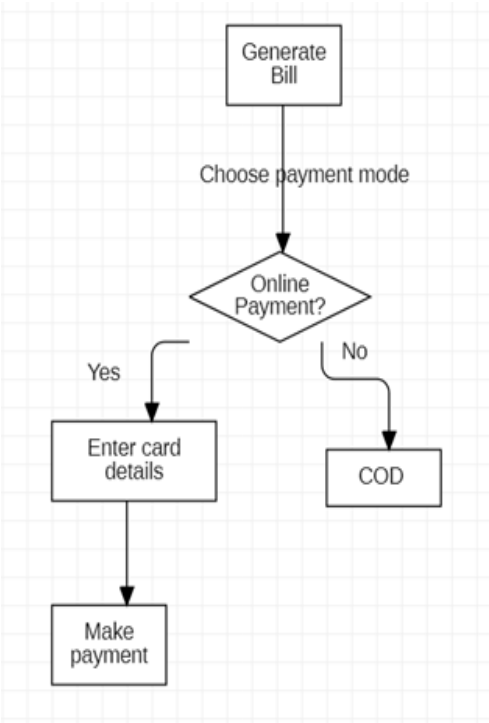
1. A bill is generated with features to either Cancel or Checkout with payment

2. Integrate with a payment gateway for credit/debit card transaction
3. In case of COD option, the bill is later updated to database

DATABASE TABLE - PAYMENT

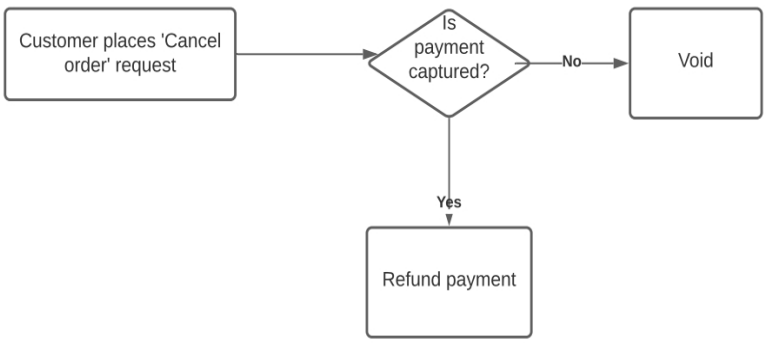
Attributes: CustomerID (foreign Key) , OrderID (Foreign Key), Card details, Bill, Receipt

DFD FOR PAYMENT

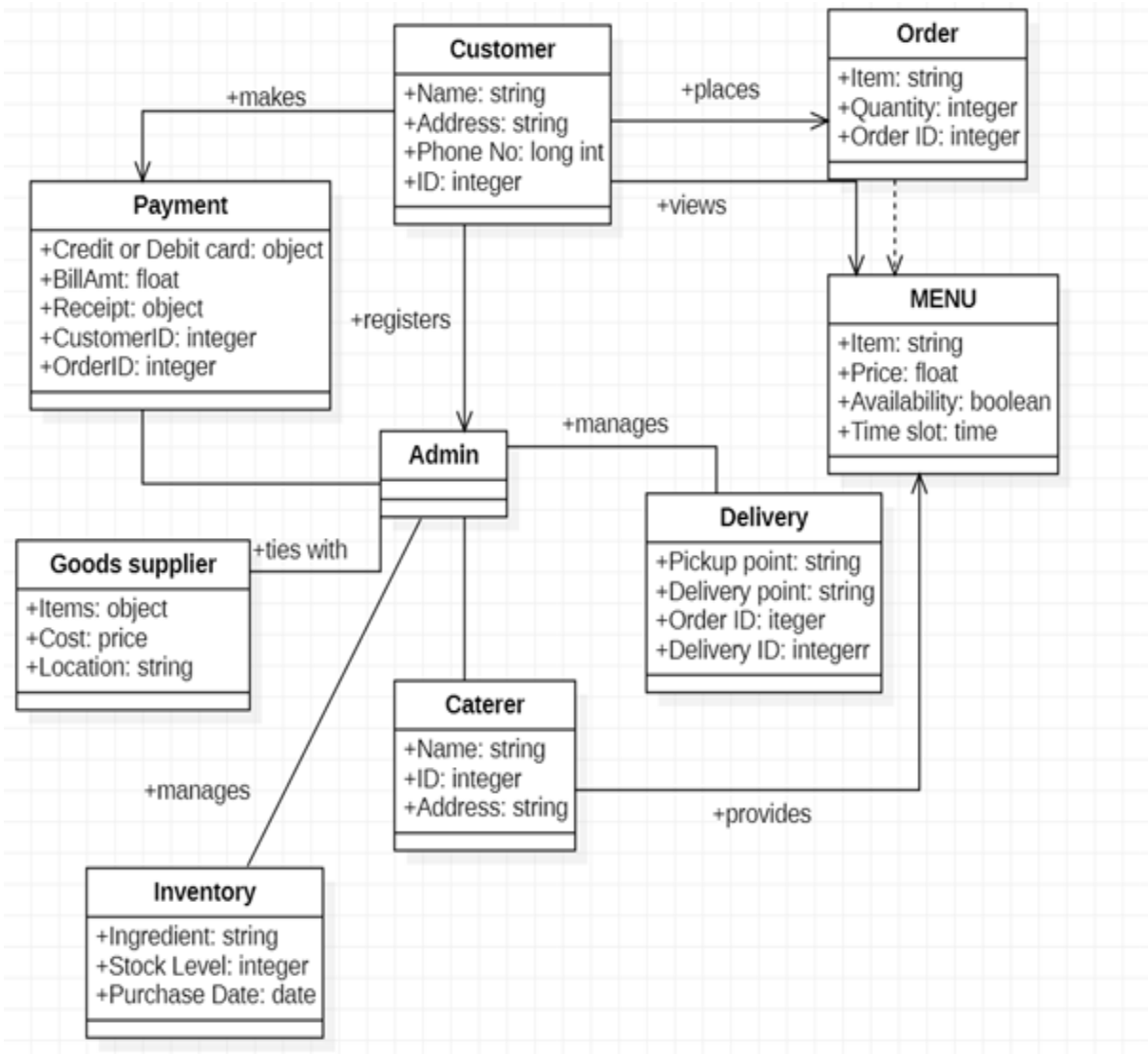


CANCEL ORDER AFTER PAYMENT

In this case, refund is done but the person is charged 20% of the order amount.



RELATIONSHIP DIAGRAM BETWEEN DIFFERENT MODULES



PROTOTYPE / IMPLEMENTATION

- ➔ The front end is developed using ReactJS and MySQL is the database server that is preferred.
- ➔ The front-end is connected with the back-end through API calls.
- ➔ REST (REpresentational State Transfer) architecture is used for request and response URLs.
- ➔ Backend is developed using Java frameworks and micro services.
- ➔ The web based application can also be used on mobile androids and is launched to the PlayStore.

➔ The system must use reliable third party servers to disburse the fund and ensure there is no sharing of sensitive information.

Priority order for engineering build and deployment, in my opinion after impact analysis, would be:

1. Developing the menu page with initially necessary filters
2. Managing the inventory (procuring the goods)
3. Developing the online ordering page with properly working APIs that act as an interface between the frontend and backend
4. Integrating payment gateway for online ordering

TESTING

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed. Retesting, regression testing is done until the point at which the software is as per the customer's expectation.

UNIT TESTING - During this first round of testing, the program is submitted to assessments that focus on specific modules of the software to determine whether each one is fully functional.

Goods Procurement and inventory

- Empty the stock details and test how the inventory works
- Update the stock details with different values to check

Payment

We need to ensure the payment processor does not fail if multiple users try at the same time. Verify the flow:

- Place the order
- Check if funds are received in merchant account
- Verify if refund takes place properly in case of cancelled transaction

Menu Management

Change the time slots and test if it gets grouped in the correct slot. (i.e if the timing of a dish is changed from 8AM to 12PM, it must be moved from Breakfast to Lunch)

INTEGRATION TESTING - Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. We test if the entire web page is collaborating with

each other. Need to ensure there is proper navigability between 2 interfaces and changes in a module are reflected properly in the other phase.

SYSTEM TESTING - System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards.

DEPLOYMENT

The deployment phase is the final phase of the software development life cycle (SDLC) and puts the product into production. After the project team tests the product and the product passes each testing phase, the product is ready to go live. This means that the product is ready to be used in a real environment by all end users of the product.

Based on the features developed in the previous week, deployment takes place. Deployment team would spend (Tuesday-Thursday) first in staging and then to production. The code is deployed to a QA or testing environment that is accessible to specific users and as close as possible to a real-world environment. That way, users can continuously test the software and send it back for improvement.

For the first 2 weeks, the system accepts orders only for 5 customers.
Small patches will be fixed on the fly in production stage.
Small fixes involving errors in updating databases will be taken care here. Continuous testing and integration will be done.

In the upcoming weeks, the number of customers served will be increased gradually and new features will be included.

FEEDBACK

Collect feedback on ease of using the app, payment method (transaction process), quality assurance - quality of food, is the quantity sufficient, procurement of goods is received.

1. Feedback on the ease of using the app
2. Changes in payment method - Include mobile wallet
3. Scale the product to a higher level
4. Order tracking system
5. Add more filters on choosing the menu
6. Improvements in delivery system and tracking
7. Include ratings feature

Details collected in feedback will be used in the next iteration

ITERATION 2 (June- July)

In the second phase of iteration, we increase the scalability of the product. We gradually move to serving more customers and the focus is not restricted to a specific area, but the whole city is covered in this phase.

REQUIREMENTS & PLANNING

The requirements involve working on the feedback collected from the previous iteration. At this stage, since there are few relaxations on the lockdown, new features are covered.

Target Users - This is now increased to 50-60 people and anyone within the city will be able to avail the service.
Features - More importance is given to client satisfaction. Order tracking system is one of the major feature that end users look out for. Additional improvements to the existing model of payment is needed.

In the first 2 weeks of June, order tracking feature is set as the top priority. The next two weeks will focus mainly on including other payment options - like connecting to mobile wallet via UPI. Likewise, improvisations to the already existing features(adding filters, ratings) can be done as and when required.

ANALYSIS

(i) Order Tracking - This requires tracking at several levels. Users need to be notified at regular intervals.

- Intimate the users once the order is placed
- At the second stage, the users are notified when their order is packed and ready for delivery
- The users must be able to track the vehicle once ready for delivery.

(ii) Route Tracking - This step is a subset of Order Tracking

- Vehicles must be fitted with sensors that can track the vehicle and send it to the application for user to track

(iii) Payment - Integrating mobile wallet options like GPay, BhimUPI to the already existing methodologies.

DESIGN

Adding extraneous features to the already existing ones.

(i) Ratings

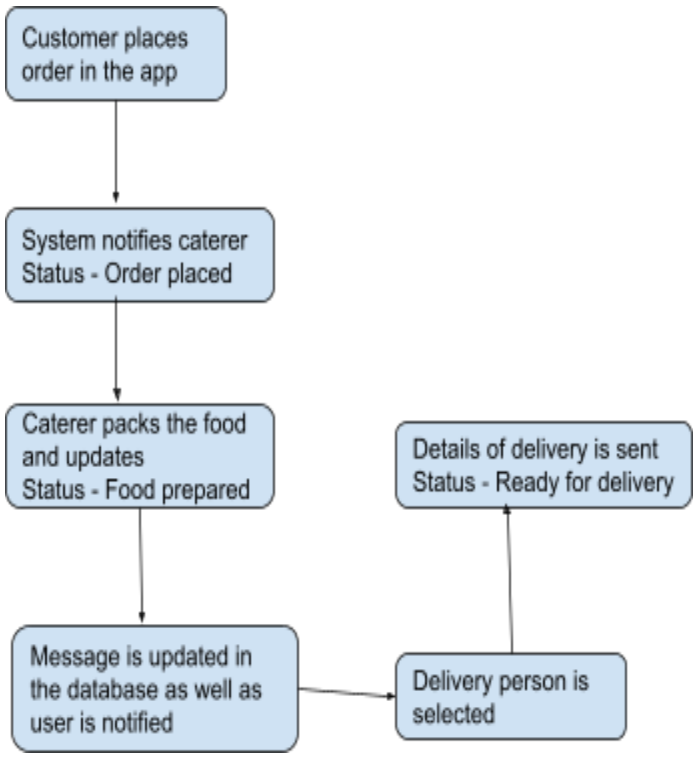
1. Initially, the option to Add Rating would be a long bar of sorts, to capture intimate users to add rating, which is a new option.
2. Option to Add Rating will be placed beside Rating Icon, in a much smaller size to the rating icon.

(ii) Search filters

1. Add Quick search filters and also an option to store previous filters
2. Suggest menus based on the most ordered food.

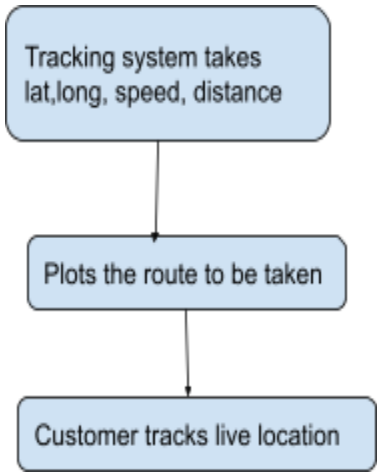
(iii) Order Tracking

A dashboard is developed for the users, system as well as the caterers to track the order right from placing till the delivery.
Flow works like this:



DATABASE TABLE - ORDER TRACKING
Attributes: CustomerID (foreign Key) , OrderID (Foreign Key), Status

Route Tracking
Based on the distance between source and destination, **ETA** (Estimated Time of Arrival) is calculated.

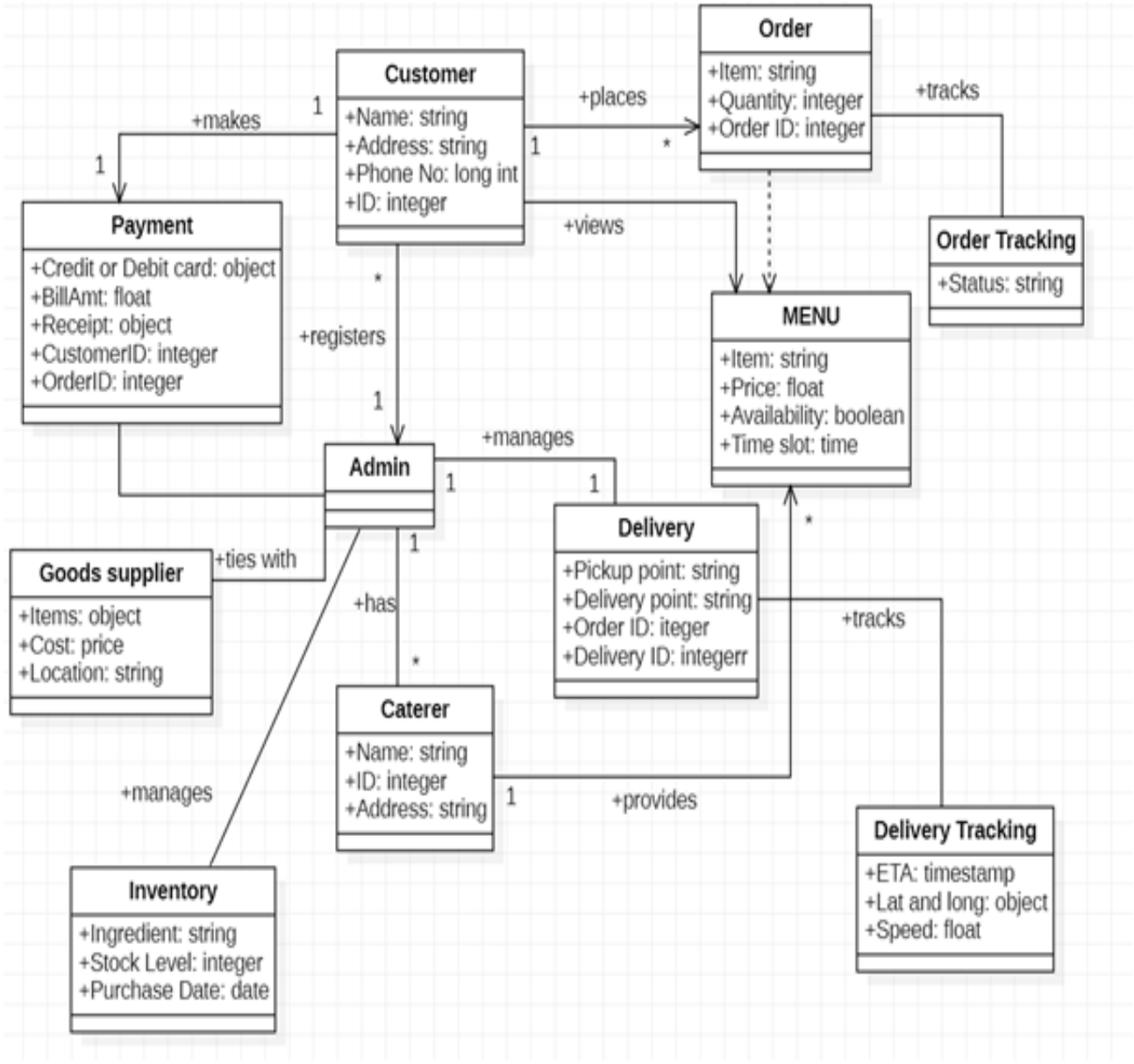


DATABASE TABLE - ROUTE TRACKING
Attributes: CustomerID (foreign Key) , OrderID (Foreign Key), ETA, Speed, Location

Payment
1. Integrate mobile wallet in Payments page
2. Split bill facility - Generate link that can be shared with others

ENTITY RELATIONSHIP DIAGRAM FOR SECOND ITERATION

Additional classes such as order tracking and route tracking are done here.



PROTOTYPE / IMPLEMENTATION

In my opinion, priority order for implementing the design built would be:

(i) Order Tracking

Support for multiple clients - Customer as well as caterer. This is done via API calls

Creating a dashboard to keep track of the status

(ii) Route Tracking
Use Google Maps to integrate with the Tracking system.
Use route planning algorithm and implement the optimised route to travel between 2 points

(iii) Payment
Implementing an in-house developed payment gateway and integrating with GPay, BhimUPI

Then, we can add extra features such as **Ratings** and **Filters.**

TESTING

Testing is done on how accurately the tracking is done.
(i) For **order tracking**:

- Place multiple orders for same caterer and check how status is updated
- Generate sample orders and check on database updation
- In case of errors, rely on manual updation

(ii) For **route tracking**:

- The app is first tested for distances less than 2ke first test on distances that are less than 2kms and then slowly improve.
- Test the GPS module
- Geography based GPS signal testing
- Artificial signal feeding to the software system to check anomalies

(iii) For **payment** with UPI apps, we need to test in a different way:

- Generate sample token to ensure decryption takes place securely
- Set timeout for UPI transactions and test how it responds once timer expires

DEPLOYMENT

Sprint Release: The first 2 weeks of June focuses on developing the order and route tracking system. Small bug fixes will be done on the fly. Status updation is the important feature to look upon.
In the next 2 weeks of June, integrating mobile wallet payment is released for the customers to use it. Split sharing the bill feature will not be released at this stage as initially we mainly focus on payment processing.

Simultaneously, new features such as Ratings and Filters are added. These features will be updated as and when required based on the comments from users.
In the first week of July, we release the new feature of Split Sharing for users.

FEEDBACK

Collect feedback on ease of using the app, payment method (transaction process), quality assurance - quality of food, is the quantity sufficient, procurement of goods is received.

1. Users wish to have review system
2. Expand the catering system to take orders for office contracts
3. Have a separate quick search option
4. Offers and coupons

ITERATION 3 (From August)

In the third phase of iteration, we expand the type of services offered. With the ease of restrictions on lockdown and business resuming to operation, this iteration focuses on expanding the services to office contracts.

REQUIREMENTS & PLANNING

With many relaxations and ease in lockdown, we provide buffet service offers for office catering.

Target Users - Buffet service model where the company can select their own type. This includes providing the service to around 40-50 office employees.

Features - In this 3rd iteration, we expand the service. Hence, a different menu placing feature is added exclusively for corporates. Along with that, additional features like user review system, quick search option and offers & credits options are added.

ANALYSIS

(i) Review
Giving only ratings will not be useful as areas of improvement will not be known. With a written review feature, the users can also state their problems as well as give testimonials that are helpful for both the system as well as the catering service.

(ii) Buffet Service Model
- A separate model exclusive for corporates

- Clients can place order along with choosing their own menu for breakfast, lunch and dinner

(iii) Automatic Customer Support
For queries, customer support is offered.

(iv) Suggestions - Quick Search Option
This comes handy when users are confused what to choose. Based on the past order history and the order details (such as menus that were selected more frequently, menus that were given higher ratings) will be displayed for the users as suggestions.

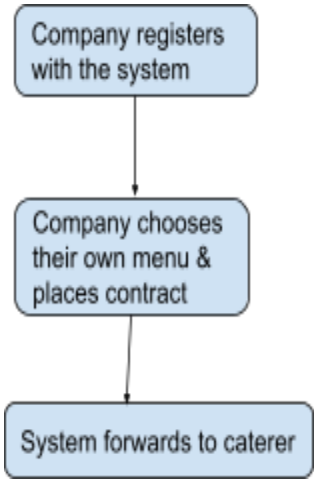
(v) Offers & Credits
Regular customers will be given coupons which can be applied. Also special offers will be given from time to time.

DESIGN

(i) Review
Reviews Count Icon, and the option to view them on clicking this icon, would be present close to the Rating Icon, to indicate similarity **(Principle of Proximity)**
The review page is placed in the home page as users can easily access.

(ii) Buffet Service Model

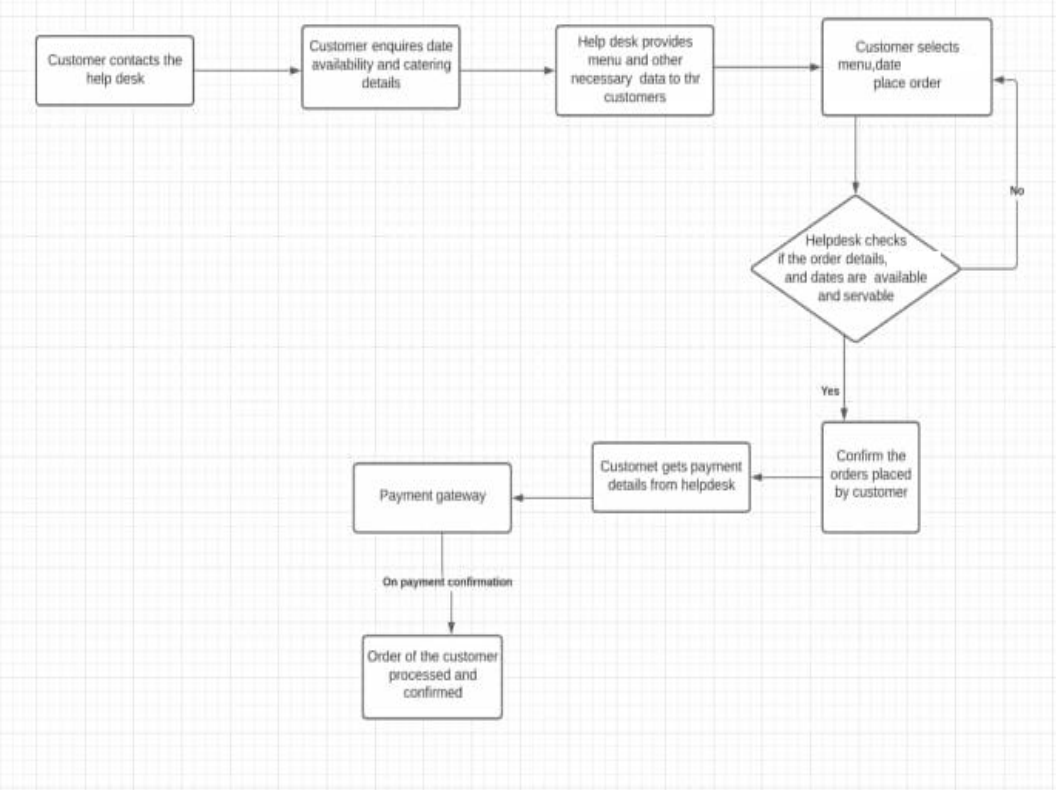
DATABASE TABLE - BUFFET SERVICE
Attributes: CompanyID (foreign Key) , OrderID (Foreign Key), Number, Food choice



(iii) Automatic Customer Support

In case, the user is not able to make orders, he can direct the help desk to do that as well as the payment.

Help desk is used when the users find it difficult to make order. In this case, the user can tell his queries and the helpdesk resolves.



DATABASE TABLE - AUTOMATIC HELP DESK

Attributes: CustomerID (foreign Key) , OrderID(foreign Key), Query

(iv) Suggestions - Quick Search Option

In the place order page, the suggestions filter is positioned so that users can view the top rated, frequently ordered dishes.

(v) Offers & Credits

Coupon codes are placed in the home page to lure the users.

PROTOTYPE / IMPLEMENTATION

The 3rd iteration focuses on expanding the scalability. Priority order for this iteration would be:

I. Buffet Service Model
II. Automatic Customer Support
III. Suggestions
IV. Offers & Credits

TESTING

Unit Testing

(i) Automatic Customer Support

- Create different test issues like payment failure, order tracking failure
- Handling technical issues faced by customer

(ii) Offers and Credits

- Generate different coupon codes and offers
- Test for invalid coupon codes and ensure it does not reflect any discount
- Test with expired offers

At the next level, **bottom-up integration testing** is done followed by **system testing.**

Finally, **acceptance testing** is done. The aim of this type of testing is to evaluate whether the system complies with the end-user requirements and if it is ready for deployment.
The scope of acceptance testing ranges from simply finding spelling mistakes and cosmetic errors, to uncovering bugs that could cause a major error in the application.
By performing acceptance tests, the testing team can find out how the product will perform when it is installed on the user's system. There are also various legal and contractual reasons why acceptance testing has to be carried out.

DEPLOYMENT

The most important feature that will be worked on is the Buffet Service model and this will be released to go live after the first 2 weeks of August. This is then followed by deploying the Help Desk for customers to use.Additional features such as Suggestions, providing Offers & Credits will be done simultaneously and released.
Continuous deployment (CD) adds an additional step to the automation process where code is sent to production after it passes a series of tests. Only if testing fails will code be sent back to developers, and the automation process will remove it from the next deployment. Once developers fix any issues, the process is repeated until it's finally sent to production. With both

CD and CI, integration of the latest software upgrades happen faster, are seamless to your users, and help development teams become more agile.

CONCLUSION

The lockdown caterer system is developed using the **Agile Model**. We have discussed the development of the system right from the basic stage and then slowly moving to the next level with each iteration.
The primary advantage of the agile model is that we need not wait until the entire system is developed. We can release the features for customers to use as and when required, which proves to be efficient in the present case scenario. We use the **scrum framework** which ensures effective planning and each sprint planning lasts for 2-4 weeks that ensures developers are also not overloaded with work and also ensures complex projects are broken down effectively and solved in incremental steps. We thus hope the above proposed agile model for the lockdown caterer system ensures faster, collaborative development of the product.