

---

# Architecture Modelling

Software Architecture

February 28, 2022

Richard Thomas

---

## 1 Before Practical

Prior to attending your practical class,

- read *at least* the [architectural views notes](#)<sup>1</sup> and this worksheet,
- review a modelling notation such as [UML](#)<sup>2</sup> or [C4](#)<sup>3</sup>, and
- either install a modelling tool on your computer, or
- have set up an account to use an online modelling tool.

## 2 This Week

Our goal is to get acquainted with a modelling notation and tool to be able to create diagrams to provide a visual representation of parts of a software architecture. You will need to be able to produce diagrams representing parts of your architectural models throughout this course.

## 3 Example System

During this practical you will create a set of diagrams to describe the architecture of an on-line food delivery system (e.g. a simple [Uber Eats](#)<sup>4</sup>).

Customers can search for restaurants that deliver to their address using a mobile or web application. They can view restaurants on an interactive map to see their location. They can select a restaurant to view its menu and select menu items to add to their order. They can complete and pay for an order. Customers can monitor the progress of their order through either the web or mobile apps. Once their order has been collected for delivery, customers can track the location of the delivery agent. Customers are notified by an alert in the app and a text message when the delivery agent is within 2 minutes of arriving.

Payment is handled through a payment gateway (e.g. PayPal, Stripe, ...). When a customer completes an order, the system contacts the restaurant to submit the order. The restaurant confirms acceptance of the order and gives an estimate of when it will be ready to be collected. The system schedules a delivery agent to arrive at the restaurant by the estimated collection time. Restaurants can update the system with a new estimated collection time. This may cause the system to reschedule when the deliver agent will arrive or assign the delivery to a new agent. The restaurant notifies the system when the order is ready to be collected. The system notifies the delivery agent that the order is ready.

Delivery agents are people who deliver orders. They have a mobile app that they use to register their location and availability to deliver orders. When a delivery agent is allocated to collect an order from a restaurant, the app displays a map with a route to the restaurant from the agent's current location and estimated travel time. When a delivery agent collects the order, they scan a QR code on the order to

---

<sup>1</sup><https://csse6400.uqcloud.net/schedule/#week2>

<sup>2</sup><https://www.uml-diagrams.org/>

<sup>3</sup><https://github.com/structurizr/dsl/blob/master/docs/language-reference.md>

<sup>4</sup><https://www.ubereats.com/au>

confirm they have collected the correct order. Their mobile app then displays a map with a route to the delivery address. The app sends the delivery agent's location to the system when they have moved 100 metres or they have been in one location for 2 minutes.

Restaurants interact with the system through either a web or mobile application. They use their app to notify the system when they are able to accept orders and when they are not able to accept orders. The app notifies the restaurant when an order has been placed. The restaurant confirms acceptance of the order and enters an estimate of when the order will be ready. The restaurant uses the app to indicate an order is ready and it prints out a delivery receipt with QR code that is attached to the order.

The system keeps track of customer browsing and order history. It uses this data to make recommendations and special offers to customers. Customers can view their past orders and reorder the same items or they may save orders as frequent orders.

Restaurants can view their accounts to see what orders have been placed and what income has been generated through the system. The system pays restaurants by bank transfer on a weekly basis.

If a restaurant rejects an order, the system notifies the customer and refunds the purchase. For simplicity, we will ignore details about how customers, delivery agents, and restaurants register with the system.

## 4 System Context

Create a context diagram for the on-line food delivery system. It is recommended that you use the C4 system context diagram for this, even if you intend to use UML for the other diagrams. You may use a high-level business use case diagram with system boundaries (*formally subjects in UML*<sup>5</sup>) [1], and possibly packages, to provide a similar contextual diagram.

TODO: Remainder of diagrams and notation and tools notes.

TODO: Get Brae to integrate plantuml diagrams better

---

<sup>5</sup><https://www.uml-diagrams.org/use-case-subject.html>

Dot Executable: /opt/local/bin/dot  
File does not exist  
Cannot find Graphviz. You should try

```
@startuml
testdot
@enduml
```

or

```
java -jar plantuml.jar -testdot
```

## 5 Further Work

Continue working on your model of the architecture for the on-line food delivery system. Share your ideas, diagrams, and model source on Slack.

## References

[1] K. Fakhroutdinov, “Uml-diagrams.” <https://www.uml-diagrams.org/>, 2020.