Institute of Systems Science
National University of Singapore

MASTER OF TECHNOLOGY IN SOFTWARE ENGINEERING

Graduate Certificate Examination

Subject: *Architecting Scalable Systems*

Sample Examination Questions

APPENDIX A

'DaDi' CASE STUDY

‘DaDi’ Case Study

In pursuit of the Smart City, one of the initiatives is to use crowdsourcing to minimize the carbon footprint in Singapore, due to increasing traffic on the roads. Also, the public currently has to make use of several apps to determine the fastest way to go from a point to another; eg: would it be faster to take an MRT from office to home as compared to taking a car-sharing service from Grab. Although there are many independent taxi-hailing companies such as Grab that offer car-pooling services from individuals, it does not seem to be popular for several reasons. Some reasons could be lack of governance and lack of incentives for individuals to participate etc. A forward thinking company called ‘DaDi’ wishes to create a viable platform where the public have the information regarding all possible transport options from one point to another and are able to make a decision based on several factors including time taken, cost, etc.

You are appointed as the Chief Solution Architect of the team tasked to develop this platform. You are responsible for the technical delivery of the system which includes the design, development and deployment of the platform and two pilot applications in the first phase. When the first phase completes successfully, the platform should allow the expansion of new services and inclusion of new actors.

1. Initial Requirements of the Platform

The new platform should be able to support the following actors:

1. Public Commuter: Members of public who consumes the services of this platform. They interact with the platform using the DaDiCommute mobile app.
2. Private Driver: Vehicle owners to sign-up as drivers and update their availability and planned travel routes (starting point and destination). They interact with the platform using the DaDiDrive mobile app.
3. Charted Bus Operator: Chartered bus owners to sign-up as transport providers and/or update the availability of seats in their planned routes (starting point and stops). They interact with the platform using the DaDiDrive mobile app.
4. LTA APIs: Land Transport Authority APIs for getting the public transportation data (bus, MRT and taxi related data)
5. Taxi App: Existing taxi-hailing apps such as Grab and GoJek to integrate their services.
6. Taxi App Provider: The operators of existing taxi-hailing services. They interact with the platform using a Web app.
7. Payment Gateway: Third party payment service providers.

Figure 1 overleaf illustrates the high-level functionality of these initial requirements.

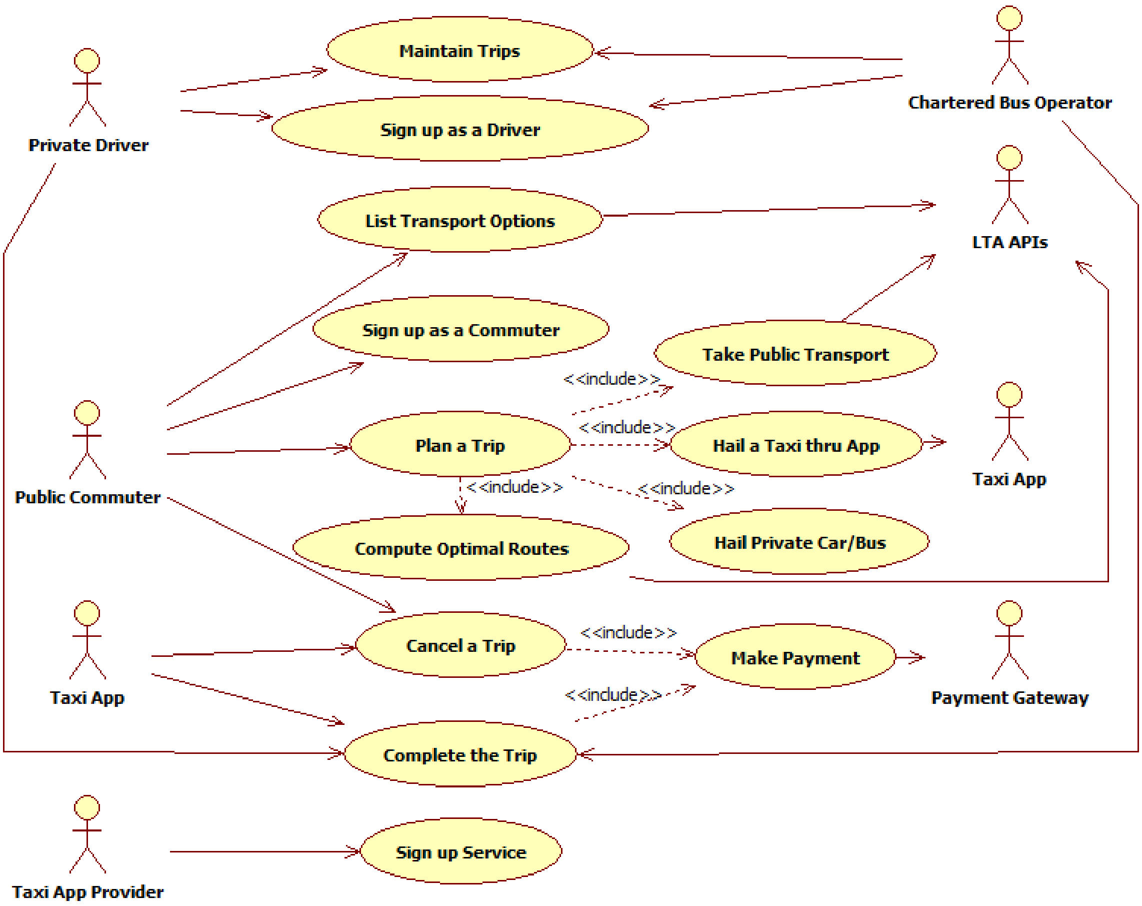

Figure 1: System Context for DaDi

Here is a brief description of the use cases for DaDi:

Use Case	**Description**
List Transport Options	This use case is invoked when a Public Commuter wishes to know his/her transport options to commute from one place to the other. The platform will invoke the available LTA APIs to collate the public transport options available. In addition, the platform will also collate the available matching trips offered by Private Drivers and Chartered Bus Operators. It will also list the existing taxi hailing services like Grab, who have signed up as a provider in the platform.
Plan a Trip	This use case is invoked when a Public Commuter wishes to plan a trip from one place to another. It invokes the “Compute Optimal Routes” use case to compute a few optimal routes from one place to another in ascending order of preference based on the available public transport, Private Drivers and Chartered Bus Operators. Based on the commuter’s choice, this use case will then invoke use cases “Take Public Transport”, “Hail a Taxi thru App” or “Hail Private Car/Bus” correspondingly to show more details of the public transport, launch the preferred Taxi App and send a message to the Private Driver or the Chartered Bus Operator. Note: The payment for the trip will be done when the trip is completed.

Cancel a Trip	If the Public Commuter cancels a trip, the “Make Payment” use case will be triggered to deduct a nominal cancellation charge from the commuter’s account, based on the rules for cancellation charges. The payment may be done using the Payment Gateway depending on the commuter’s preference of payment method.
Complete the Trip	When the trip is complete, the Private Driver/Chartered Bus Operator/Taxi App will then indicate so, upon which the “Make Payment” use case will be triggered. The payment may be done using the Payment Gateway depending on the commuter’s preference of payment method.
Sign up as a Commuter	This use case is invoked when a Public Commuter wishes to come on board the platform as a Consumer of the transport services.
Sign up as a Driver	This use case is invoked when a Private Driver/Chartered Bus Operator wishes to come on board the platform as a Provider of the transport services.
Sign up Service	This use case is invoked when a Taxi App Provider wishes to come on board the platform as a Provider of the transport services.
Maintain Trips	This use case allows a Private Driver/Chartered Bus Operator to add, delete or modify a trip by giving details such as the starting point, ending point and number of seats available in the transport etc.

2. Additional Requirements of the Platform

In addition to the initial requirements, this platform is to be extended to support HuoYun and BanJia systems. HuoYun is a system that allows Private Drivers to bid for delivery requests for goods submitted by Goods Owners. BanJia is a system that allows Home Movers to bid for requests for moving home submitted by Home Owners. Both systems coordinate the goods delivery and home moving processes and handle the payments from consumers to providers.

The platform should be able to support the following actors:

1. Goods Owner: The owners of goods to be delivered. They interact with the platform using the HuoYunGoods mobile app.
2. Home Dweller: The dwellers of the content of home to be moved. They interact with the platform using the BanJiaHome mobile app.
3. Home Mover: The providers of home moving service. They interact with the platform using the BanJiaMove Web or mobile app.
4. Private Driver: Vehicle owners to sign-up as drivers and bid for goods delivery requests. They interact with the platform using the DaDiDrive mobile app.
5. Payment Gateway: Third party payment service providers.

Figures 2 and 3 in the next few pages illustrate the high-level functionality of these additional requirements.

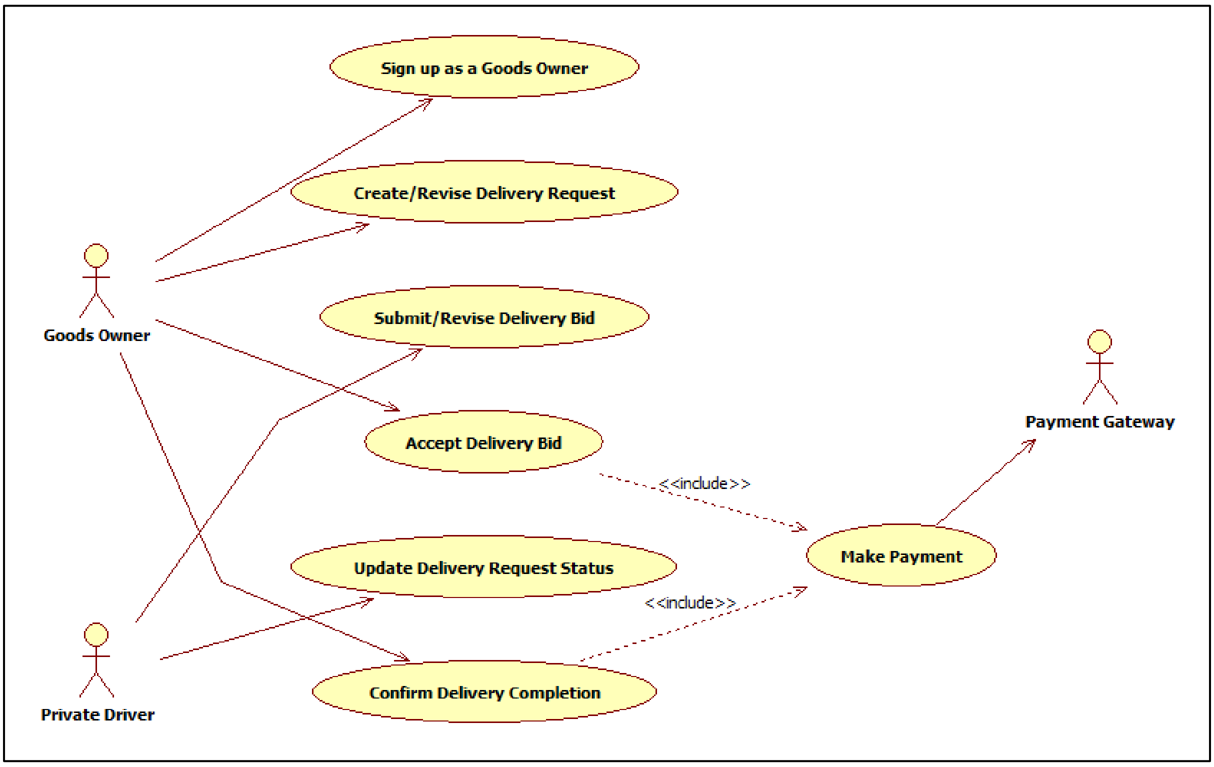

Figure 2: System Context for HuoYun

Here is a brief description of the use cases for HuoYun:

Use Case	**Description**
Sign up as a Goods Owner	The owner of goods signs up as a Goods Owner. He/she can then consume the services provided by HuoYun.
Create/Revise Delivery Request	When a Goods Owner has some goods to deliver, he/she creates a delivery request along with details on the type and volume of goods, the destination, the speed of delivery, etc. As long as he/she has not accepted any delivery bid, he/she may revise the details of the delivery request.
Submit/Revise Delivery Bid	A Private Driver browses the available delivery requests. When he/she is interested in a delivery request, he/she submits a delivery bid along with details on the cost and the dates of pickup and delivery completion. As long as his/her delivery bid is not accepted, he/she may revise (including retract) the details of the delivery bid.
Accept Delivery Bid	A Goods Owner browses the submitted delivery bids for his/her delivery requests. He/she may accept a delivery bid. Upon the acceptance of a delivery bid, a deposit payment would be effected through the Make Payment use case. The Private Driver who submitted the bid would also be informed via push notification. He/she would then carry out the delivery request accordingly.
Update Delivery Request Status	As a Private Driver carries out a delivery request, he/she updates the status of the request timely. The statuses are scheduled, picked up, in transit, delayed, delivered and completed.
Confirm Delivery Completion	Once a Goods Owner independently verifies the delivery of his/her goods for a delivery request, he/she marks the delivery request as complete. The remaining amount of payment for the delivery request would be effected through the Make Payment use case. The Private

	Driver who delivered the goods would also be informed via push notification.

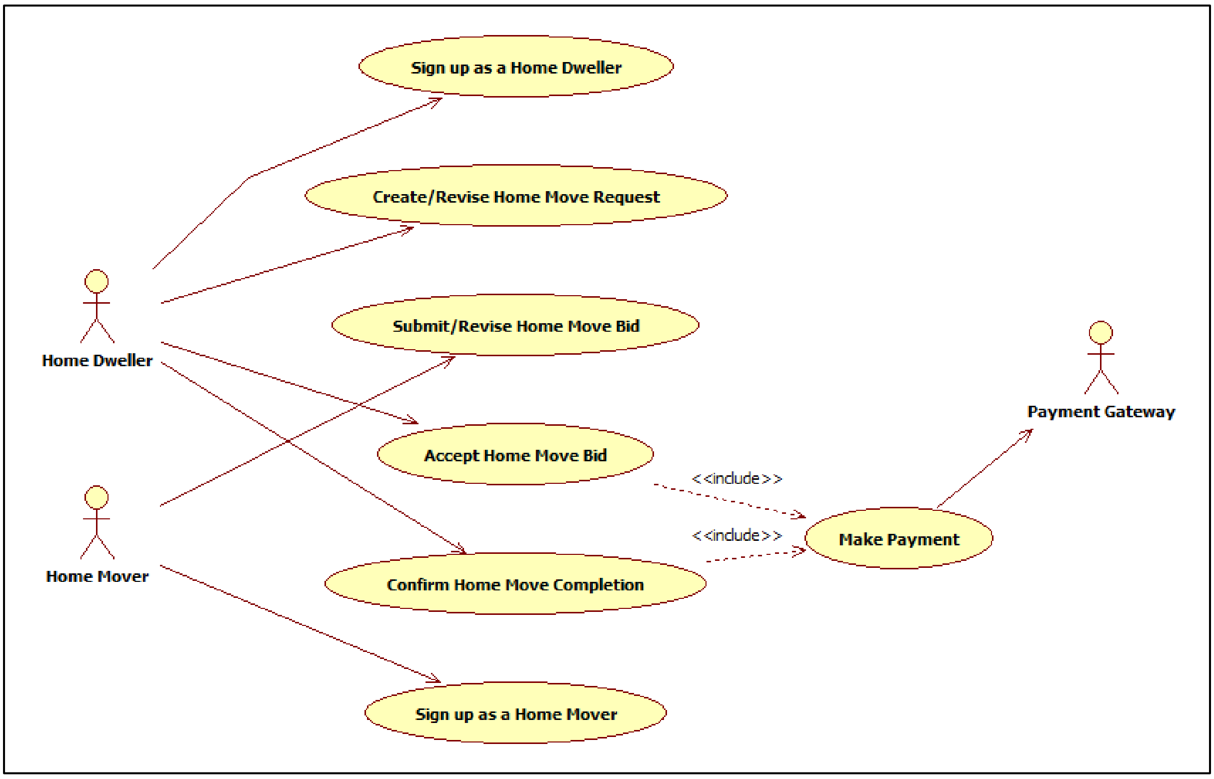

Figure 3: System Context for BanJia

Here is a brief description of the use cases for BanJia:

Use Case	**Description**
Sign up as a Goods Owner	The dweller of a home signs up as a Home Dweller. He/she can then consume the services provided by BanJia.
Create/Revise Home Move Request	When a Home Dweller wishes to move his/her home, he/she creates a home move request along with details on the types and volumes of furniture, the estimated number of boxes, the destination, the date of relocation, etc. As long as he/she has not accepted any home move bid, he/she may revise the details of the home move request.
Submit/Revise Home Move Bid	A Home Mover browses the available home move requests. When he/she is interested in a home move request, he/she submits a home move bid along with details on the cost and the timings of pickup and delivery completion. As long as his/her home move bid is not accepted, he/she may revise (including retract) the details of the home move bid.
Accept Home Move Bid	A Home Dweller browses the submitted home move bids for his/her home move requests. He/she may accept a home move bid. Upon the acceptance of a home move bid, a deposit payment would be effected through the Make Payment use case. The Home Mover who submitted the bid would also be informed via Web or push notification. He/she would then carry out the home move request accordingly.
Confirm Move Completion	Once a Home Dweller independently verifies the completion of his/her home move request, he/she marks the home move request as complete. The remaining amount of payment for the home move

	request would be effected through the Make Payment use case. The Home Mover who carried out the home move request would also be informed via push notification.
Sign up as a Home Mover	The provider of a home move service signs up as a Home Mover. He/she can then bid for home move requests of BanJia.

3. Extensibility Requirement for the Platform

The platform should contain common services and libraries that support and speed up the development of future services provided by the platform. In the short term, the design evaluation of the extensibility of the platform will be done based on how the platform can support the development of the commute functionality as well as new types of delivery.

The platform should be accessible from any computers including mobile devices and PCs.

4. Performance and Scalability Requirement of the Platform

The platform should be designed for the following performance and scalability requirements:

- The platform should be able to handle increasing amount of data while maintaining the response time of the application. The design should consider the possibility of sharding and partitioning of data for different trips, routes, etc. where necessary.
- For interactive request-response type of operation (e.g. listing the transport options for Public Commuters), the expected response time should be less than 1 second for 90% of the requests.
- The platform should be able to handle multitenancy with proper data and application isolation between tenants. Example of tenants would be different Taxi App Providers and Home Movers. As the multitenancy requirement is not known in detail at the moment, the architect should use his/her judgement to choose the optimal trade-off regarding multitenancy.
- The platform currently manages 10TB OLTP data sets. Trip data transactions are monitored and the current system handles 1M trip transactions per minute. Similar count for the payment data store as well. But there is a sudden surge in the requirements as we plan to expand business in United States of America and Australia.

5. Security Requirement of the Platform

The platform should be designed with the following security requirements:

- The system should follow the general security best practices in infrastructure and application security e.g. reverse proxy in DMZ, etc.
- Personal information should be stored in encrypted form and the platform should be able to find all information stored about an individual. According to GDPR, the actors should allow users to request access to user's data and the platform should be able to facilitate the compliance.
- Sensitive tenant information should be encrypted with tenant-specific encryption key to avoid accidental data leakage due to software bugs. The encryption keys should be stored securely and the system should able to handle change of encryption key through an automated process.

- Subsystems that handles payment and cardholder information should comply to PCI-DSS and therefore payment and cardholder information handling should be limited to as minimal subsystems as possible to minimize cost and minimize risk exposure.

6. Details on Some Significant Use Case Flows (Main Flows)

6.1 List Transport Options Use Case

- The Public Commuter selects the "Transport Options" option in the DaDiCommute app (a mobile app used by Public Commuters).
- Upon being prompted by the app, the Public Commuter provides the starting and ending locations of the intended trip.
- The platform retrieves a few routes for public transport (i.e. bus, MRT and taxi) from the LTA APIs.
- The platform retrieves a few trips offered by Private Drivers and Chartered Bus Operators.
- The app displays the above transport options.

6.2 Plan a Trip Use Cases

- From the list of menu options shown in the DaDiCommute app, the Public Commuter selects the "Plan Trip" option in the DaDiCommute app.
- The platform invokes the "Compute Optimal Routes" use case to perform the following:
    - The platform retrieves additional information on public transport from LTA APIs.
    - The platform computes the traveling distances, traveling times and costs of a few optimal routes based on the list of transport options.
- The DaDiCommute displays the details of the optimal routes. The Public Commuter may sort them by traveling distance, traveling time or cost.
- The Public Commuter picks one of the optimal routes and selects the "Confirm Trip" option.
- Depending on the choice of the optimal route, the platform performs the following:
    - Bus or MRT: The platform invokes the "Take Public Transport" use case to retrieve the detailed information of the route from LTA APIs and display it on the DaDiCommute app.
    - Taxi: The platform invokes the "Hail a Taxi thru App" use case to launch the Taxi App in the mobile device of the Public Commuter. The Taxi App is specified as a user preference in the DaDiCommute app.
    - Private Driver or Chartered Bus Operator: The platform invokes the "Hail Private Car/Bus" use case to send a push notification to the DaDiDrive app (a mobile app used by Private Drivers and Chartered Bus Operators) to notify the Private Driver or the Chartered Bus Operator of the trip assignment.

6.3 Complete the Trip Use Case

- When a trip is completed, depending on the type of trip, the platform performs the following:
    - Bus or MRT: Not applicable.
    - Taxi: The Taxi App notifies the platform of the completion of a taxi trip. The platform invokes the "Make Payment" use case to effect the payment and send the receipt to the Public Commuter via email. The "Make Payment" use case invokes the Payment Gateway for some payment methods.
    - Private Driver or Chartered Bus Operator: The Private Driver or the Chartered Bus Operator confirms the completion of the trip using the DaDiDrive app which invokes the "Make Payment" use case to effect the payment and send the receipt to the Public Commuter via email.

6.4 Cancel a Trip Use Case

- If the Public Commuter needs to cancel a trip before it begins, depending on the type of trip, the platform performs the following:

  - Bus or MRT: Not applicable.
  - Taxi:
    - The Public Commuter cancels via the Taxi App.
    - The Taxi App notifies the platform of the cancellation of the trip.
  - Private Driver or Chartered Bus Operator:
    - The Public Commuter selects the “Cancel Trip” option in the DaDiCommute app.
    - The platform sends a push notification to the DaDiDrive app to notify the Private Driver or the Chartered Bus Operator of the cancellation of the trip.
- The platform marks the trip as cancelled.
- The platform invokes the “Make Payment” use case to effect a nominal cancellation fee and send the receipt to the Public Commuter via email. The “Make Payment” use case invokes the Payment Gateway for some payment methods.

7. Initial Architecture of the Platform

As the Chief Solution Architect, you have provided the following architectural guidelines for the platform and the applications.

7.2 Architecture Layers

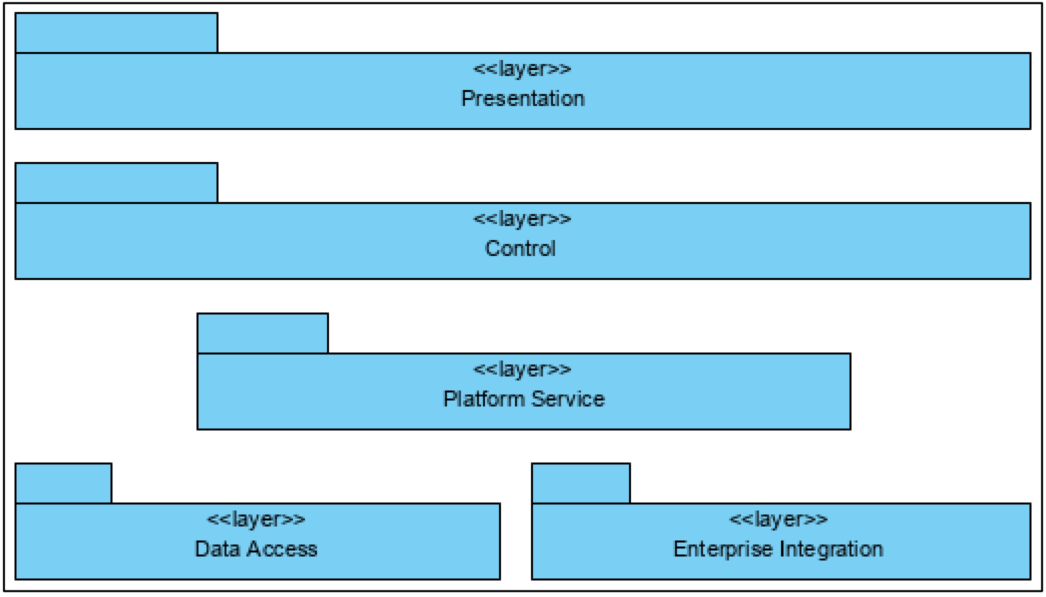

Figure 4: Architecture Layers

7.3 Architecture Overview

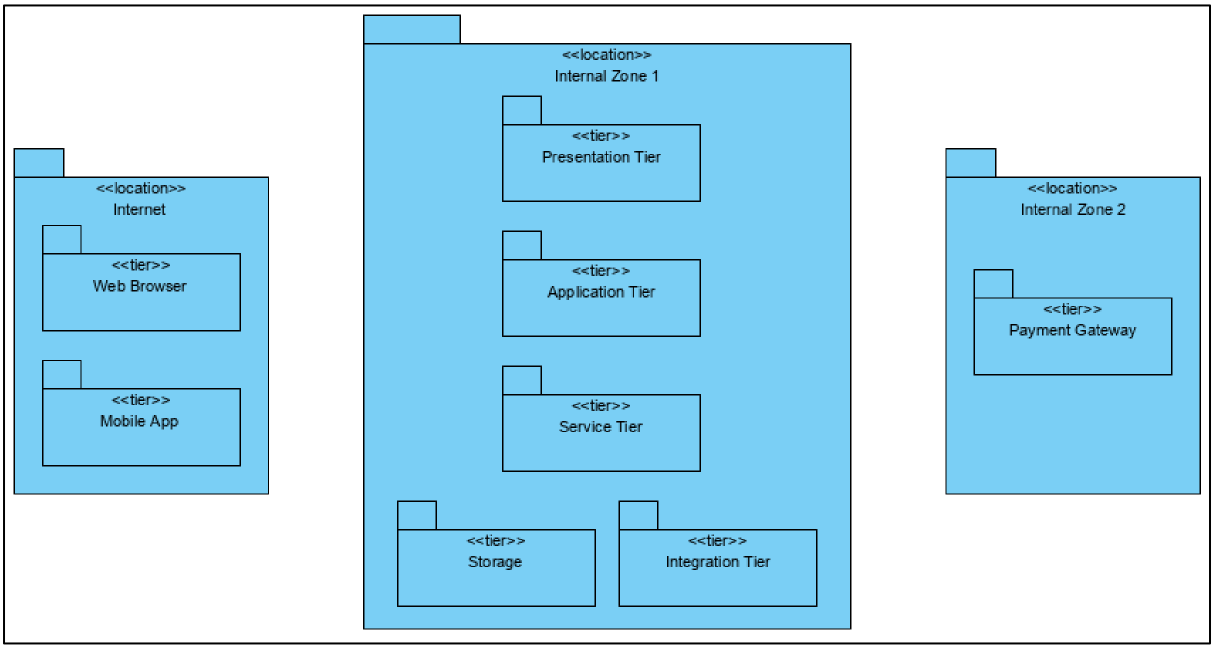

Figure 5: Architecture Overview

7.4 Deployment Options

The platform is to be designed as a Cloud native solution. The reusable services should be implemented as microservices.

8. Projected Growth of the Platform

The Business Development team had conducted a comprehensive market survey on the projected growth of the platform. The following two charts summarize the key growth data that the platform should be architected and designed to handle in the first three years of operation. The data is projected over half-yearly intervals. E.g. “Y1-2H” denotes the second half of the first year.

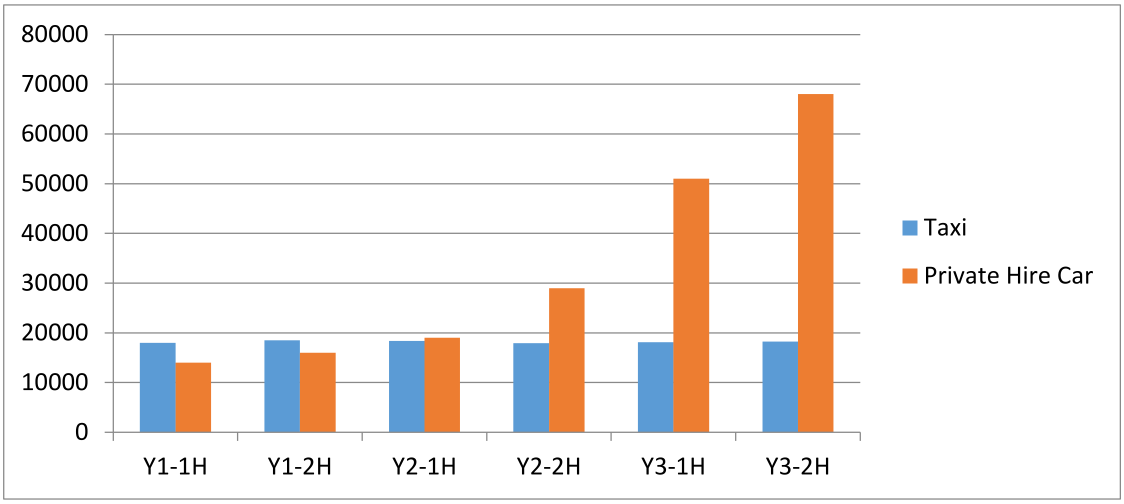

Figure 6: Number of Taxis and Private Hire Cars

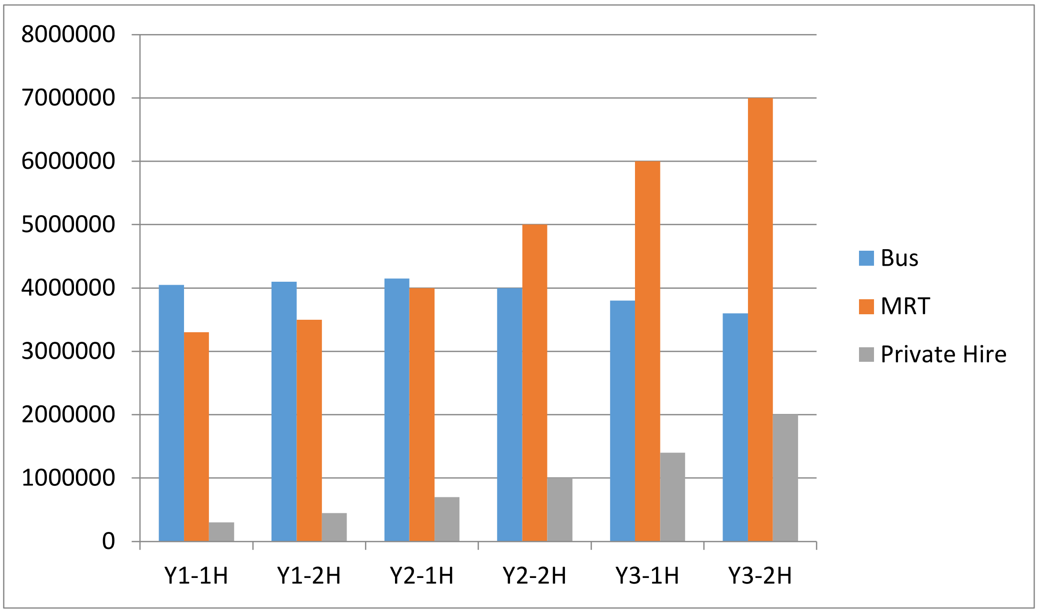

Figure 7: Average Daily Ridership

Note that the numbers of MRT lines and bus routes are not expected to change significantly. It is more important to cater to the number of public commuters and the number of trips made.