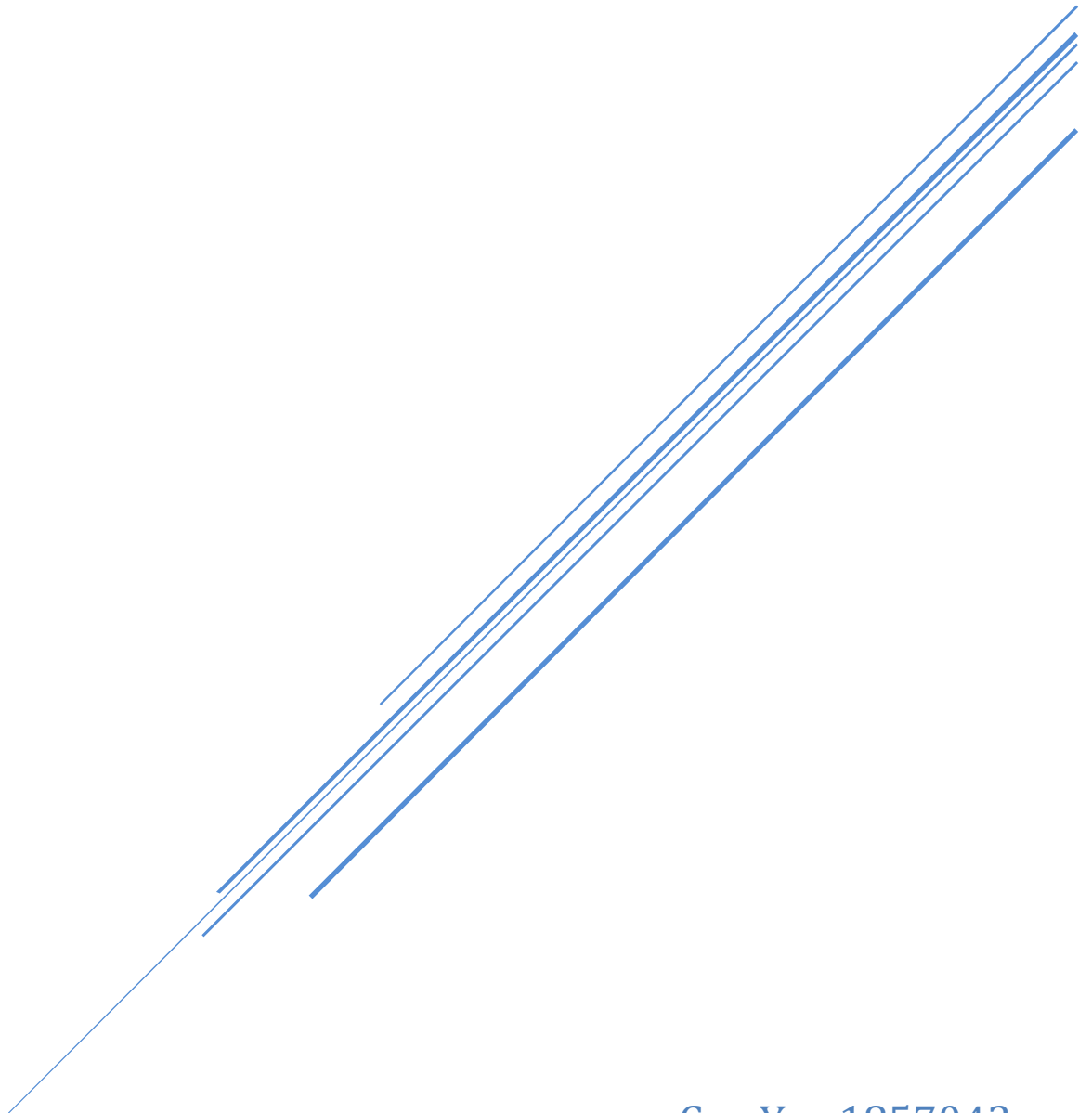


SOFTWARE ENGINEERING

CW1 Requirement



Cen Yu s1857043
Zhou Shengyi s1864509

Table of Contents

Q3.1 Identify Stakeholders	2
Q3.2 Describe System State	3
Q3.3 Use-Cases	5
1. Get quote	6
2. Book Quote	9
3. Register bike return to original provider	11
4. Register a provider account	12
5. Register a Customer account	13
6. Register new bike types	13
7. Make partner agreement	14
8. Record bike return to partner	15
9. Deliver bike to Customer	15
10. Deliver bike to Original provider	16
Q3.4 Use-Cases Diagram	17
Q3.5 Non-Functional Requirements	18
Q3.6 Ambiguities, subtleties, incompleteness	19
Q3.7 Self-assessment	21

Q3.1 Identify Stakeholders

Stakeholders	How does the System affects them
Bike Provider (Core stakeholders)	Bike provider will be affected <u>directly</u> as it could increase in customer base and competitiveness among competitors. Thus, causing an increase in revenue.
Customers (Core stakeholders)	Customers will be affected <u>directly</u> as it could help customers to get a desirable bike in a very convenient way. If the system is well designed, it may attract more customers to choose cycling as their way of travelling.
Delivery Drivers (Core stakeholders)	Delivery Driver will be affected <u>directly</u> as it provides them with more job opportunities and extra potential incomes .
Local Government (Additional stakeholders)	Local government will be affected as it could provide more job opportunities, boost tourism with more possible tour sites(Cycling routes) and reduce citizens' carbon footprint.
Environmentalism (Additional stakeholders)	Environmentalism will be affected as they are willing to see the publication of the system because it encourages every user to adapt a low-carbon lifestyle.

Q3.2 Describe System State

- What information does the system need to keep on customers, bikes, bookings, providers?
 - Customers' information
 1. Customer ID
 2. Address
 3. PersonallInfo
 - a. Name
 - b. contactNumber
 - c. EmailAddr
 4. Current_Order
 5. PaymentInfo
 - a. Card Number
 - b. CVV
 - c. Expire Date
 - d. Billing_Address
 - Providers' information
 1. Provider ID
 2. PersonallInfo
 - a. Name
 - b. contactNumber
 - c. EmailAddr
 3. Shop Address
 4. Partners
 5. Stock
 6. Deposit_rate
 7. ValuationPolicyCalculator
 8. Current Order
 - Bikes' information
 1. Owner ID
 2. Status
 3. Bike_Type
 - a. Name
 - b. Model
 - c. replacementValue
 - d. Default_DailyRentalPrice
 4. Bike_ID
 5. Daily rental price
 6. Date_of_Purchase
 7. BookedDates
 8. Current_Order
 - BikeRentalOrder's information
 1. Order_Status

2. Order_ID
3. Customer_id
4. Provider_id
5. Customer_Address
6. Provider_Address
7. Ordered_Bikes
8. DateRange
9. Require_Delivery
 - a. True/False
10. totalDeposit

- Any other information the system needs to record?
 - ProviderManagementSystem
 1. Providers
 2. ID_Provider_Map
 - ValuationPolicy
 1. Different ValuationPolicies
 - PricingPolicy
 1. Different PricingPolicies
- What different status of bikes and bookings should the system track?
 - Bikes
 1. INSHOP
 2. DELIVERING
 3. ARRIVED
 4. RENTED
 5. INPARTNERSHOP
 - BikeRentalOrder (Booking)
 1. BOOKED
 2. BIKEDELIVERING
 3. BIKEARRIVED
 4. BIKERETURNEDTOPARTNERSHOP
 5. COMPLETED

Q3.3 Use-Cases

1. Get quote
2. Book quote
3. Record bike return to original provider
4. Register provider account
5. Register new bike types
6. Make partner agreement
7. Record bike return to partner
8. Deliver bike to Customer
9. Deliver bike to original provider
10. Register customer account

1. Get quote

Use case name	Get quote	
Primary actor	Users	
Supporting actor	ProviderManagementSystem	
Stakeholder(s)	Customer: They hope to use system to obtain quotes of their desired bikes Provider: They hope to use system to increase sales.	
Precondition	<ul style="list-style-type: none"> User inputs number and types of bikes, date range, and location of hire. 	
Trigger	Users search for quotes	
Guarantee	Success Guarantee	<ol style="list-style-type: none"> If there exists bike providers who can meet all requirements of the user, the system will return a list of quotes from those providers. If no quote is available for customer's date range, system will recommend any quotes for the same duration within 3 days before the start or after the end of the date range. If no desired providers found, display a message saying so.

	Failure Guarantee	<p>1. System reminds user something is wrong and gives a brief reason why this has occurred.</p> <p>Possible exceptions are:</p> <ul style="list-style-type: none"> • TLE (Time limit exceeded) • No internet connection • Hardware issues (incompatible with the system, etc)
	Minimal Guarantee	System either outputs messages or a list of quotes
Main success Scenario	<ol style="list-style-type: none"> 1. System confirms searching scope 2. Server processes information given by user and find matched quotes from database of Provider based on the searching criteria (location, bikeNo, Bike types, startDate, endDate) 3. Server outputs the list of quotes 4. User browses quotes 	

Extensions	<p>1a: System detects wrong format of the information provided by users</p> <p>1: Users may re-enter the information or may cancel</p> <p>- System detects user exit while searching for quotes</p> <p>1. Skip all steps in MSS.</p> <p>P.S. “- event“ means the event can happen in any steps in MSS</p>
Notes	<p>Considering most users will likely to turn off the system if responding time for search is beyond 10s -15s. It is essential to ensure the system has high efficiency in execution of code.</p> <p>If possible, the system should provide some axillary functions to comfort users’ emotion while waiting for quotes such as progress bar, animated text, and etc.</p>

2. Book Quote

Use case name	Book quote	
Primary actor	Users	
Supporting actor	Bike provider, database of bike provider	
Stakeholder(s)	Customer: They wants to book quote in advance, so they can access the bikes on a specific day.	
Precondition	<ul style="list-style-type: none"> The bike chosen is available on the selected date range 	
Trigger	Customer wants to book quote	
Guarantee	Success Guarantee	User and provider both receive confirmation messages and the booked bike(s) become(s) unavailable to other customers.
	Failure Guarantee	User exit booking interface and there are no changes made to the database of system
	Minimal Guarantee	User either receives confirmation email or nothing.

Main success Scenario	<ol style="list-style-type: none"> 1. User enters personal information required 2. User goes to check out 3. System presents full pricing information 4. User agrees on 'user terms' 5. User enters credit/debit card info 6. System authorizes purchase 7. System updates database 8. System records down info of the financial transaction 9. System sends confirmation email to user 10. System sends order info to provider
Extensions	<p>1a: User is regular customer</p> <ol style="list-style-type: none"> 1. System displays saved personal info 2. User may override or accept default info, and return to step 3 in MSS(Main success scenario). <p>1b: User selects delivery but the address provided is too far away from provider</p> <ol style="list-style-type: none"> 1. User may select "store collection" as mode of collection or may cancel <p>4a: System may fail to authorize card purchase</p> <ol style="list-style-type: none"> 1. User may re-enter card info or may cancel <p>- if Delivery driver is required, continue following steps after step 10 in MSS:</p> <p>11.BikeRentalOrder activates DeliveryService and schedule a delivery for the customer.</p> <p>12. Order status is updated</p>

	<ul style="list-style-type: none"> - Security is the biggest concern in making online payment. - Letting users to accept 'User term' is necessary for managing potential future legal issues

3.Register bike return to original provider

Use case name	Record bike return to original provider
Primary actor	Provider
Supporting actor	Delivery driver/Customer

Stakeholder	Original Provider: he/she wants to register bike returned so that the bike could be available again for the next customers.
Precondition	<ol style="list-style-type: none"> 1. Provider accepts return of bike from customer 2. Provider returns deposit to customer <p>Or</p> <ol style="list-style-type: none"> 1. Provider accepts return of bike from delivery driver
Description	<p>Alternative flow 1: If the bike received from customer is damaged, provider can detain a percentage of deposit based on original provider's 'deposit policy' and damage condition of the bike.</p> <p>Alternative flow 2: If provider receives bikes from delivery driver, after checking the bikes' condition, the provider then could complete the delivering order with delivery driver.</p> <p>For both alternative flows: Provider reports to system that bike is returned and may make the bike available again for other customers. Meanwhile, the system will also record down the order info and save it in the history order from the bike object.</p>

4. Register a provider account

Use case name	Register a provider account
Primary actor	Provider

Description	Provider has to enter her/his name, shop address, shop postcode, phone number and opening hours into the system. Also, the password needs to be set. System will send a SMS to verify user's identification. If there exists wrong input, provider can edit as many times as she/he wants until the system finally accepts the input.
Notes	SMS verification is added to system's functionality as it is an effective mean to perform authentication of users (login, reset password via SMS).

5. Register a Customer account

Use case name	Register a Customer account
Primary actor	Customer
Description	Customer has to enter her/his name, address, postcode and phone number into the system. Also, the password needs to be set. System will send a SMS to verify user's identification. If there exists wrong input, customer can edit as many times as she/he wants until the system finally accepts the input.
Notes	SMS verification is added to the system's functionality as it is an effective means to perform authentication of users (login, reset password via SMS).

6. Register new bike types

Use case name	Register new bike types
---------------	-------------------------

Primary actor	Provider
Supporting actor	Bike
Description	Provider needs to input name of bike type, model description, full replacement value and daily rental price for the bike. Meanwhile, provider needs to set their own deposit policies, the deposit amount is by default, a fixed percentage of the replacement value of the bike ($\leq 100\%$). Provider may also customize the deposit rate by taking other factors into account (e.g. age of bike).

7. Make partner agreement

Use case name	Make partner agreement
Primary actor	Provider (Party A)
Supporting actor	Provider (Party B)
Description	<p>*Party A* accepts the partner agreement on system and input details of *Party B* (shop id, name of owner, address). System sends a confirmation message attached with the agreement to *Party B* on the other side. The partnership will not be activated until *Party B* accepts the partner agreement.</p> <p>The partner agreement will be invalid if *Party B* does not accept it in two weeks. *Party A* can also cancel the agreement at any time during the two weeks.</p>

8. Record bike return to partner

Use case name	Record bike return to partner
Primary actor	Provider
Description	<p>If the bike received from customer is damaged, partner can detain a percentage of deposit based on original provider's 'deposit policy' and the damage condition of the bike.</p> <p>Otherwise, partner returns the full deposit to the customer.</p> <p>System records down amount of deposit, updates bike status, and sends a message to notify the original partner to return the deposit to the partner.</p>

9. Deliver bike to Customer

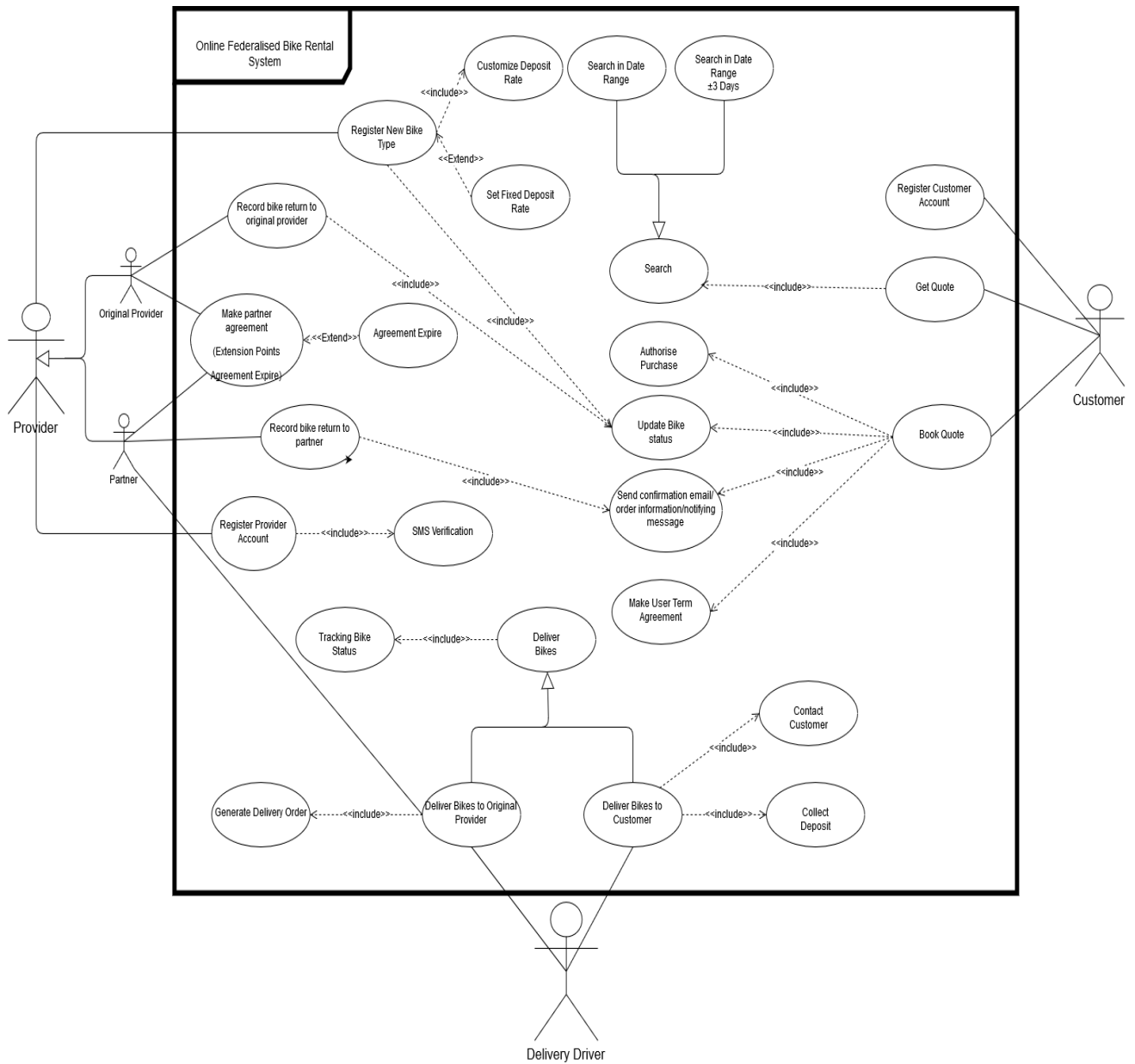
Use case name	Deliver bike to Customer
Primary actor	Delivery driver
Precondition	<ol style="list-style-type: none">1. Customer has booked quote2. Customer chooses delivery

Description	<p>Delivery driver collects information from booking order (where to collect bike? where is bike delivered to? when does customer need the bikes).</p> <p>Delivery driver then contact with the provider via system to in order to collect bike(s) from providers.</p> <p>Delivery driver also needs to contact customer via system to double check order information and appoint a time slot to conduct delivering service.</p>
-------------	--

10. Deliver bike to Original provider

Use case name	Deliver bike to Original provider
Primary actor	Partner
Supporting actor	Delivery Driver
Description	<p>Partner will send out a notifying message to delivery driver(s) for them to collect bike on time.</p> <p>After the collection of bike, partner has to enter the basic information regarding the original provider and then generates a delivering order which shares among delivery driver and providers.</p>

Q3.4 Use-Cases Diagram



Q3.5 Non-Functional Requirements

Categories	Possible Non-functional requirements
Performance	<ol style="list-style-type: none">1. Short response time. When a user is searching for quotes meeting his/her needs, the system should show the result within 30 seconds with proper internet connection. If the response time has passed 30 seconds, the performance of the system is considered to be unqualified.
Privacy	<ol style="list-style-type: none">1. Show only the relevant details of one stakeholder to another. Most user information should be kept confidential to others such as credit card information, contact number and order history. The system will only make the information visible in a specific group of stakeholders if there is a business dealing among them. For example, delivery driver must know customer's phone number in order to contact them to collect the bike.
Accessibility	<ol style="list-style-type: none">1. Provide the system with different languages with proper translation such as English, Chinese, French, etc. As one of the focus groups for this system is tourists, having a language choice for different nationalities could be more accessible for potential customers.2. Using simple, short instruction and colorful UIs to help customer register and be familiar with the product. With lots of wording and plain background and UIs, customers will have a bad first impression of the system. It may seem dull with instruction for each step but with proper expression, holding customers though the tutorials could more likely attract customers to choose the system than competitors.
Security	<ol style="list-style-type: none">1. Using Encryption for all the information to ensure stakeholders information' safety. As stakeholders share their personal information into the system, if the system is hacked, the leakage of stakeholders' details such as customers' credit card information could result in loss of image or even lawsuits. Thus encryptions are important whether within the codes or by other security companies.2. Set up multiple firewalls to protect our systems from hackers. As hackers could possibly attack our server, we have to implement firewalls to protect customers, providers and delivery drivers' personal information.

Data Retention	1. Recording down customers' information for ease of potential future bookings. With accounts binding to customers, if they want to book the bikes again, the system shall record the information of customers and key in for them when they are booking, providing convenience.
-----------------------	---

Q3.6 Ambiguities, subtleties, incompleteness

Ambiguity, subtleties, incompleteness in System	Potential options for resolution of ambiguity
In aspect of creating a provider account, there is not enough effective measure taken by the system to ensure account security.	<ol style="list-style-type: none"> 1. SMS Verification function module is added to the use case of creating provider account 2. Requiring provider to login with password
The information of the bike that provider inputs to the system while registering new bike type is not verified.	By sending team members to check the bikes condition off-line to ensure the quality.

<p>Many use cases involve updating bike status. However, in each different case, the actor may execute different operation and have different status options for them to update to. For example, customer does not need to click anything and the system will automatically set bike status to “Booked” once they have made payment online. On the other hand, the provider may need to modify these statuses manually.</p>	<p>Create a new use case named “Update bike status” to specify each different scenario.</p> <p>Alternative flow 1: when system authorizes payment,</p> <p>Alternative flow 2: when original provider receives bike, ...</p> <p>Alternative flow 3: when delivery driver collects bike from original provider, ...</p> <p>Alternative flow 4: ...</p>
<p>There is no exact time stated in use case for partnership to expire. It is essential for partners to re-agree the terms or cancel it on right time, because there may be a modification to the original partnership agreement. Also, terminating partner contract during the operation of the shop could be very costly.</p>	<p>Add a fixed time to state how long the agreement could last. On the expiring date, the partners could choose either continue the partnership or cancel it.</p>
<p>It is not stated in use cases whether cancelation of bookings is allowed.</p>	<p>Set relevant booking cancelling policies and state it in user terms which users must accept before making payment.</p>
<p>System does not include a method to resolve conflicts between stakeholders.</p>	<p>Providing a platform for stakeholders to report the conflicts or problems to the team managing the system.</p>

Q3.7 Self-assessment

Criteria	Marks	Marks and Justification
Q3.1 Identify stakeholders	15%	
- Identify core stakeholders of the system	5	Three core stakeholders identified: Customers, bike providers, delivery drivers. They are the main primary actors in the use cases.
- Identify additional stakeholders	3	We only identify two additional stakeholders who is indirectly influenced by the system.
- Describe how the system affects each stakeholder	3	In each description, we clearly state what is the influence system have on each stakeholder, and how will they benefit from the system.
Q3.2 Describe system state	10%	
- Include state essential to the operation of the system	4	We have stated customers, providers, bookings, bikes' states for system to be able to function.
- Include additional state mentioned in the description	4	We have stated delivery drivers' state and bikes, bookings' status to support the proper function of the system.
Q3.3 Describe use cases	40%	
- Identify use cases	9	We identify 10 use cases
- Describe use cases using the appropriate templates	20	Two use cases are described with full template. The rest use cases are explained with detailed description.
Q3.4 Use case diagram	15%	

- Correctly use UML use case notation	5	
- Include key actors and use cases	5	Key actors are Delivery drivers, Customers and providers. All the possible use cases involving key actors have been listed clearly.
- Identify connections between actors and use cases	4	Every main actor in the diagram has association with the relevant use cases. We also categorize the provider into original provider and partner to build the connection between them and relevant uses case in a better manner.
Q3.5 Describe non-functional requirements	10%	
- Identify non-functional requirements within the context of the system	5	We have stated 7 non-functional requirements within 5 categories, yet the linkage to the context may not be enough.
- Provide means for assessing non-functional requirements	2	Some of the non-functional requirements do not come with a proper assessment method.
Q3.6 Ambiguities and subtleties	5%	
- Identify some ambiguities in system description	3	6 reasonable ambiguities are identified
- Discuss potential options for resolution of ambiguities	2	We have come up with feasible solutions for all identified ambiguities and incompleteness.
Q3.7 Self-Assessment	5%	
- Attempt a reflective self-assessment linked to the assessment criteria	4	The self-assessment may be subjective and miss out lots

		of potential problems to talk about
--	--	-------------------------------------