

# Software Engineering

## *Requirements Engineering*

**Orges Skura - s1813106**

**Sami El-Daher - s1830962**

## Stakeholders

- Customers (cyclists) - the system will make it easier for them to rent the bikes, so more cyclists may come to the area for holidays. It may also encourage more people to cycle when they might drive instead.
- Providers (local bike rental shops) - the system will allow the rental shops to rent bikes to more customers, it will also mean they will have to partner up with other shops and interact with them in ways including transferring deposits, sending bikes between themselves
- Scottish tourism board - tourism will be boosted, and the system may improve the environment in the country which may also result in more tourists visiting.
- Environmental organisations - the system may be beneficial for the environment
- Travel agencies - they will be involved in the increase in tourism, as more people may take holidays to the area due to the system, which would result in more money being spent for tourism.
- Delivery drivers - they will have to transport bikes between providers, which results in more money and jobs for them.
- Bike manufacturers - people will request more bikes so they will get to produce more bikes, earning more money.
- Passport control - they will be affected by an increase in people coming to visit Scotland due to the new bike rental system, so the sector will have to do more work.

## System State

1. available providers
  - 1.1. location of provider
  - 1.2. deposit rate
  - 1.3. partnership agreements
  - 1.4. deposits owed to other partners
  - 1.5. bikes deposited from other providers
2. which bikes a provider owns
  - 2.1. types of all bikes

- 2.2. which bikes are on loan
  - 2.3. which bikes are available
  - 2.4. bike replacement cost
  - 2.5. rental cost for each type
- 3. details about orders
  - 3.1. order number
  - 3.2. customer details
  - 3.3. bikes included in the loan
    - 3.3.1. types of bikes on loan
    - 3.3.2. number of bikes of each type
  - 3.4. which provider the bikes were loaned from
  - 3.5. loan start and end dates
  - 3.6. cost of loan
  - 3.7. deposit amount
  - 3.8. mode of collection
  - 3.9. return information
  - 3.10. return location
- 4. customer details
  - 4.1. first name
  - 4.2. surname
  - 4.3. address
  - 4.4. postcode
  - 4.5. phone number
- 5. bike information
  - 5.1. bike type
  - 5.2. status

Additional:

- 1. list of bike types
  - 1.1. replacement value
- 2. deposit amount proportion

# Use Cases

1. Book quotes
2. Return bike to original provider
3. Get quotes
4. Registering in the system
5. Registering bikes
6. Set deposit policies
7. Registering partnership agreements

## Use Cases Templates

### Book quotes

**Use case name:** Book Quotes

**Primary actor:** Customer

**Supporting actors:** Bike Providers

**Summary:** The customer books one of the quotes listed before. He can collect the bike from the provider's store or get it delivered to his address

**Precondition:** There is at least one quote listed from the system

**Trigger:** Listing of the quotes, in which case the customer selects one of them

**Success Guarantee:** The booking will be successful and the bike will be delivered to his address or he can now collect it from the local provider.

**Failure Guarantee:** The system will not register a booking, so no bikes are taken out of stock and the customer will not be charged

**Minimal Guarantee:** All bikes marked as available in the system will be available in real life

**Main Success Scenario:**

1. Customer browses the list of quotes and selects one of them
2. Customer provides his personal information(first name, surname, address, postcode, phone number)
3. Customer provides mode of collection
4. System redirects to payment
5. Customer pays
6. System generates order number (unique for each order),order summary, deposit, total price, delivery and return information.
7. The booking is successful and now the customer can collect the bike via the method he provided in the above step

**Stakeholders & Interests:**

1. Customer-He has an interest in the booking process being successful so he can collect the bike
2. Local Bike Provider-His bike is booked so he can keep for the one who booked it and collect the money
3. Delivery Driver-If the booking is successful and customer has chosen deliver bike as an option then he will have a job and get paid

**Return bike to original provider**

**Use Case Name:** Return bike to original provider

**Primary Actor:** Customer

**Supporting Actors:** Original bike provider, partnered bike provider, delivery drivers

**Precondition:** The customer must be renting at least one bike to be able to return it

**Trigger:** The use case will occur when the customer tries to return a number of bikes to a store, and the provider registers the return of the bikes into the system

**Guarantee:** The use case will always result in a success. The original provider will end up with their bikes, and the customer will have their deposit returned.

**Main Success Scenario:**

1. The provider returns the deposit, paid in the booking process, to the customer.
2. The provider returns the bike(s) to their stock, allowing it to be booked again.

**Extensions:**

*Return To a Partner Provider:*

- occurs at the trigger, if the customer brings the bike(s) to a provider that has joined a partnership agreement with the original provider.

1. The partner provider returns the deposit to the customer.
2. The partner notifies the original provider that the bike(s) are now returned.
3. The partner organises for a delivery driver to send the bike(s) back to the original provider.
4. The delivery driver returns the bike(s) to the original provider

- returns to step 2 of the Main Success Scenario.

**Stakeholders & Interests:**

1. Cyclists - will have to return the bikes to the registered providers.
2. Providers - will be receiving bikes, and returning deposits to customers.
3. Delivery Drivers - will be transporting bikes between providers.

**Notes:**

1. It is unclear whether the original provider transfers the deposit to the partner provider who has received one of their bikes

## **Get quotes**

**Use Case Name:** Get Quotes

**Primary Actor:** Customer

**Summary:** The customer search for a list of quotes predefining his rental needs. The system then generates a list of available quotes from which the customer needs to choose one.

### **Main Success Scenario:**

1. Customer provides information on their rental needs(number and types of bikes, date range, location of hire)
2. The system generates a list of quotes
3. Each quote shows a bike provider who can offer the bikes requested on the desired dates, as well as the total price and deposit for renting the bikes.
4. If no quote is available for the customer's date range, the system suggests any quotes for the same duration within 3 days before the start or after the end of the date range.

### **Extensions:**

1. What if no quote is generated in the end but there are available quotes for 4 days after date specifies?

## **Registering in the system**

**Use Case Name:** Registering Bike Provider in the system

**Primary Actor:** Bike Provider

**Summary:** The system has a database of bike providers. The bike providers provide their name, shop address, shop postcode, phone number and opening hours to the system and this allows them to register bikes.

**Trigger:** The bike provider attempts to register a bike in the system

**Main Success Scenario:**

1. System prompts the bike provider to enter their details
2. Bike Providers enter their name, shop address, shop postcode, phone number, and opening hours.
3. The system registers them into the database

**Extensions:**

1. If bike providers have the same name, do the other details overlap or not?

**Registering bikes**

**Main Actor:** Bike Provider

**Summary:** The bike providers register the bikes into the system by providing bike details. They can also set their daily price of renting

**Trigger:** The bike provider attempts to register a bike in the system

**Main Success Scenario:**

1. The system prompts bike providers to enter bike details
2. The bike providers enter each bike type name and the full replacement value for bikes of that type.
3. Bike Provider can also enter bikes of a certain bikes type as part of their stock
4. Bike Provider sets the daily renting price
5. System registers the bike into its database

**Extensions:**

1. Providers can set custom pricing policies, which may be more advanced, taking into account more complex factors, e.g. the number of bikes or number of days



## **Set deposit policies**

**Use Case Name:** Bike Providers set the deposit policies

**Main Actor:** Bike Provider

**Summary:** The bike providers set their deposit as a fixed percentage. That is dependent on replacement value of the bike, the deposit rate and other stuff.

**Extensions:**

1. Providers may want to set custom deposit policies, which the customer will be charged against

## **Registering partnership agreements**

**Use Case Name:** Registering Partnership Agreements

**Primary Actor:** Bike provider

**Supporting Actors:** A desired partner provider

**Summary:** Bike providers can partner with other providers, allowing loaned bikes to be returned to any bike providers partnered with them.

**Guarantee:** The system will record a partnership between the providers.

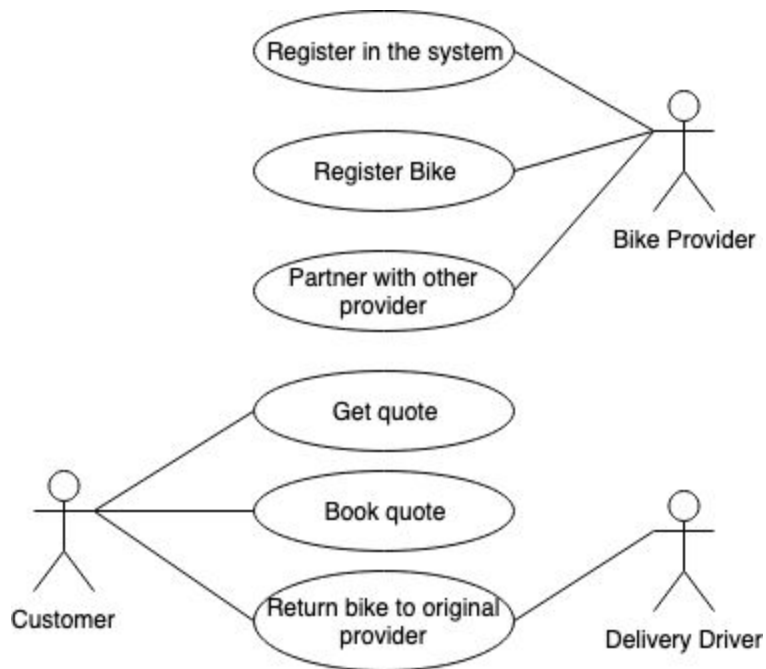
**Stakeholders & Interests:**

1. Providers - will be able to partner with other providers.
2. Customers - will be able to return bikes to providers in multiple locations.

**Notes:**

1. How are partnerships removed from the system, who would trigger it.

## Use case diagram



## Non-functional requirements

### 1. Security

- 1.1. Payments should only be made when authorised by the customer, when they are booking a bike.
- 1.2. Only real local bike shops should be able to register in the system. This could be tested by checking registered providers against an official list of businesses in Scotland.

### 2. Privacy

- 2.1. Customer details should not be accessible to unauthorised users, i.e. only available to providers who are loaning bikes to the customer, or delivery drivers that are transporting bikes. This should be tested by checking to see if one customer can view another customers details
- 2.2. Provider details, such as overall stock and bike prices, should not be accessible

to other users on the system.

### 3. Performance

- 3.1. The system should be responsive, especially as loans and bike returns may be time critical. All processes, e.g. searches or transactions, should occur within a reasonable amount of time, such as 2 seconds, and should give the user an indication that they are taking place
- 3.2. The system should maintain a valid state, that is consistent both internally and with the real world. This means for example that bikes on loan should not be accessible to other customers, or that a provider shall receive a deposit that a user has paid.

### 4. Availability

- 4.1. The system should allow as many providers to register as possible.
- 4.2. The system should be available for any potential customers.

## Ambiguities, subtleties, and incompleteness

- 1. Does the user pay on the system or on an external system like PayPal?

**Possible Solution:** The system uses an external system like PayPal to process the payment.

- 2. How does the user pay the deposit?

**Possible Solution :** The system redirects the user to pay the deposit into an external system and if the user does not do it within a certain time, the order is canceled.

- 3. What happens if the customer has not paid the deposit but the bike is coming with a delivery driver?

**Possible Solution:** The customer has to pay the deposit to the delivery driver and then delivery driver sends the money to bike provider.

- 4. After the customer returns the bike, how does the system return the deposit?

**Possible Solution:** The system uses the same payment method the user used to return the money to the user's provided credit card or bank account

- 5. How does bike provider receive money when he is not providing payment method?

**Possible Solution:** When registering bike providers in the system, there are provided also some fields for payment method?

- 6. How does partner sent the deposit amount to the other partner?

**Possible Solution:** System provides an option where one partner sends money directly to the other partner. This would be visible only for bike providers.

# Self-assessment

## Q 3.1. Identify Stakeholders - 13/15%

- **Identify core stakeholders of the system** - 5/5% - as we believe we have identified all of the stakeholders that are directly affected by the system.
- **Identify additional stakeholders** - 4/5% - there may be some other stakeholders that we have not been able to identify correctly.
- **Describe how the system affects each stakeholder** - 4/5% - we have described the systems effects on the stakeholders in enough detail for any stakeholders that we have identified, but if there are stakeholders outside of our description then they will not have a description.

## Q 3.2. Describe system state - 9/10%

- **Include state essential to the operation of the system** - 5/5% - we believe that with the state given in the description the system could be totally functional.
- **Include state essential to the operation of the system** - 4/5% - there may be some additional state required, as we have not specifically separated the system state into essential and additional categories.

## Q 3.3. Describe use cases - 36/40%

- **Identify use cases** - 9/10% - we believe we have captured all use cases for the system, but we may be giving them at a level of detail that is inappropriate for this stage.
- **Describe use cases using the appropriate templates** - 27/30% - for the two use cases that we have gone into detail (booking quotes and returning bikes to their original providers) we believe that the description given in the template would be implementable without any ambiguity. We also describe the other use cases using the template, but to a lesser degree of detail, as this will help greatly for the other steps in the engineering process. There may also be some assumptions either in the original description or in our use cases that we have unintentionally made.

## Q 3.4. Use case diagram - 12/15%

- **Correctly use UML use case notation** - 5/5%- We believe that the notation used for UML

use cases was correct and on point.

- **Include key actors and use cases** - 4/5%-We believe that the key actors are included and the most important use cases are included but maybe some of them are higher level than they should be.

- **Identify connections between actors and use cases** - 3/5%-Most of the connections between actors and use cases are identified, but we think that maybe one or 2 connections between actors are missing.

### **Q 3.5. Describe non-functional requirements - 9/10%**

- **Identify non-functional requirements within the context of the system** - 6/7% - we believe that we have found most functional requirements, but there may be some requirements, especially in other categories, that we have not been able to capture.
- **Provide means for assessing non-functional requirements** - 3/3% - some requirements have been described in a level of detail that would allow it to be verified with the final system, as well as giving some explicit tests for others.

### **Q 3.6. Ambiguities and subtleties - 4/5%**

- **Identify some ambiguities in system description** -2/3%-We believe that some of the most ambiguities are included here, but they may be some important ones that we may have missed.
- **Discuss potential options for resolution of ambiguities** -2/2%-We think the solutions for our identified ambiguities are on point.

### **Q 3.7. Self-assessment - 5/5%**

- **Attempt a reflective self-assessment linked to the assessment criteria** - 5/5% - we have tried to judge in what areas we have succeeded in reaching the assessment criteria, and in those that we have failed to reach it we have given explanations on how we might.