Capturing Requirements for a Federalised Bike Rental System

Coursework 1

Informatics 2C - Introduction to Software Engineering

Justin Howe
s1840358

Rohan Nittur
s1803949

Contents

1. Stakeholders Identification

The federalised bike rental system comprises of both primary and secondary stakeholders as emphasised in the following structure.

1.1 Core Stakeholders:

1. User
   - (a) They are the ones using the bike rental website through any mobile device or computer.
   - (b) In addition, they must put in information regarding their required rental needs for the bike and personal details.
2. The Customer (another primary stakeholder- but may also be the user)
   - (a) The only responsibility is that they must pay for the service in the system.
3. Bike Shop Staff Members
   - (a) They have to provide information for their shop and register the different bike types into the system.
   - (b) They can also set and modify the deposit amount needed along with the delivery surcharge rate into the system.
   - (c) They are also liable for establishing partnerships with other bike shop providers, if necessary for the user to allow to drop off the bike at different locations.
4. Delivery Driver
   - (a) They are liable for delivering the bike from one place to another.
     - (i) For example, the driver is responsible for delivering the bike from the shop to the user's residence.

(b) In addition, if the user requested to obtain the bike from one shop but the bike ended up in another shop, then the driver must drive his truck starting from the location of the other shop into the location of the original shop from where the user initially requested.

1.2 Additional Stakeholders

1. Scottish Tourism Board

   (a) That particular organisation initially influenced the creation of the system of the federalised bike rental system.

2. Bicycle Manufacturers

   (a) Each bike shop associated with the federalised bike rental system will need a large supply of bicycles from the manufacturers themselves. There should be a sufficient number of bikes available for the users to ride on.

3. District Councils

   (a) Each council is responsible for receiving tax from each bike shop's revenue. The tax money earned will allow them to buy more bikes from the bicycle manufacturers and then this will keep the federal bike system running.

4. Taxpayers

   (a) The money to launch the scheme and pay the government-hired workers (e.g. software developers) will come out the tax paid by the general public, and so they will be interested in the outcome of the enterprise.

2. System State

1. The system should record each customer's personal information details:
    - (a) First name and surname
    - (b) Address and postcode
    - (c) Phone number
    - (d) Method of collecting the bike: in-store or delivery
2. The system should store the status of each of the bike:
    - (a) System should record the availability of each bike for given dates
        - (i) Whether they will be in the shop
        - (ii) Whether they are available to rent on the given day
    - (b) The daily price for each bike, which will be set by the individual bike providers (see below) should be stored in the system database
    - (c) The type of the bike, and its full replacement value
3. The system should retain the following information about the provider:
    - (a) Name, shop address, shop postcode, phone number and opening hours.
    - (b) The system should also store the deposit rate for each particular provider
    - (c) The system should also store the full replacement value for each bike in that provider's
    - (d) stock for determining the deposit amount.
    - (e) Whether the provider has any partnership agreements with other providers
        - (i) The name of the other provider
        - (ii) The location of the other provider
4. The system should keep information of each booking:

(a) Order number

(i) This is required for the bike provider to refer to the details in each order such as retrieving when and what the user requires for the bike rental

(b) Delivery and Return Information

(i) Location of the bike shop where the user is expected to pick up his bike from

(ii) Location of the bike shop where the user returned his bike

A. If the user dropped off his bike at another bike shop which has a partnership agreement with the bike shop where the user picked up the bike from, then notify both bike shops

(iii) The location of the user's residence address if the bike is to be delivered to his house

(iv) The pick up date and return dates

3. Use Cases

3.1 Use Case I - Book Quote

Primary Actor: Customer

Supporting Actor(s): User- if the customer is renting the bike for another human user other than themselves

Summary: The customer is required to provide his/her personal details and their mode of collecting the bikes (in-store or delivery) and pays the required amount needed for the quote.

Precondition: The customer must have already selected his desired quote in the website system for the given dates.

Trigger: Customer selects desired quote from a list of quotes based on particular fixed/flexible dates given.

Guarantee: There are both success and failure possible guarantees. All customer's personal information details have been filled out completely and payment successfully went through. However, payment may not have gone through or there were issues with selecting the desired quote on the website including issues with processing the transaction or the user does not have enough money to continue with the transaction. This would then be a failure guarantee.

Main Success Scenario:

1. Customer provides all of the personal information needed and their choice of collecting the bikes (in-store or delivery) on the website
2. Customer pays for payment for the quote including the deposit
3. Confirmation issued to customer

Extensions:

1a. Customer doesn't fill out all personal information required

   1. The system should then prompt the customer to fill out the missing details

2a. Customer has invalid payment details (such as invalid card number)

   1. The system should request the customer to pay again

2b. Customer encounters an Internet issue or System error with booking the quote while providing his/her personal information.

   1. Once the Internet issue or System error is resolved, then the website would take him/her to the previous page where he would have to provide his personal information again.

3a. Bike becomes unavailable for rent while a user is in the process of booking it

   1. The system should notify the customer that the bike is now unavailable and return them to the home screen

Stakeholders & Interests:

1. Customer - They want to use the system to pay for the bike.
2. Software developers - They want to ensure the payment has gone through the website system successfully and that the customer's personal information has been stored in the website's database.

Notes:

- The system could send the customer an SMS confirmation of the order number, provided that the customer has filled out his phone number
- A non-functional requirement would be that the customer would have to book the quote within 5-10 minutes, otherwise booking the quote would result in a timeout.

3.2 Use Case II - Pick Up Bike

Primary Actor: User

Supporting Actor(s): Staff Member

Summary: Firstly, a staff member from the bike shop, where the user wants the bike to be collected, will register the deposit into the system. The status of the bike in the bike shop inventory would also be updated.

Precondition: Ensuring the user has given the deposit to the bike shop provider and the user wants to collect his bike in-store.

Trigger: User steps into the desired bike provider shop.

Guarantee: The deposit should be registered in the system and the system database updated to reflect the collection of the bike from the store.

Main Success Scenario:

1. Customer pays deposit to bike provider
2. Deposit amount is confirmed and registered on the system
3. Status of the bike(s) on the system is updated to show its collection
4. Customer is issued bike(s)

Extensions:

3a. Database system for updating the bike status is down or not working

   1. Bike would still be issued to customer
   2. Staff member would temporarily store the bike status on a piece of paper and a staff member would update the status when the database system is running

4a. Upon final checks it is revealed that the selected bike contains one or more defects

   1. Staff member would issue a bike similar to the original bike
   2. System would be updated to reflect the absence (renting) of the new bike

Stakeholders & Interests:

1. Customer - The user ultimately wants to ride the bike.
2. Bike Shop Provider Staff Members- They must confirm on the system the deposit amount required and issue out the bike from their respective shops.
3. Bike manufacturer- They issue out quality checks for bike.

Notes:

- When the user picks up the bike from the shop, it may be implied that at least one staff member from the bike shop provider would issue out a quality check for the bike.

3.3 Use Case III - Get Quotes

Primary Actor: Customer

Supporting Actor(s): User- if the customer is renting the bike for another human user other than themselves.

Summary: The system should retrieve quotes from bike shops which have the user selected bike types available for the user selected dates. In addition, the user should have already put down details for what type of bike and the location hire for the bike.

Extensions: An extension to this main success scenario is when no shops have the required bikes available for the given date. In this case, the system suggests any quotes for the same duration within 3 days before the start or after the end of the date range.

3.4 Use Case IV - Bike Delivery

Primary Actor: Bike Provider

Supporting Actor(s): Delivery Driver(s)

Summary: The customer may opt to receive the bike(s) by delivery. In this case, first the bike(s) are transferred by the provider to the delivery driver, and the status of the bikes on the system changed by the provider to show they are in transit. When they arrive at the customer, the deposit is collected and recorded on the system. The bikes are handed over and witnessed by the driver, who updates their status on the system to show that they have been successfully delivered.

3.5 Use Case V - Recording Partnerships Between Bike Providers

Primary Actor: Bike Provider

Supporting Actor(s): None

Summary: The first step would be one bike provider recording in the system the partnership between the other bike provider. Once the other bike provider has done the same, the system records the partnership and is able to inform customers that rentals from either of two bike providers may also be returned to the other.

3.6 Use Case VI - Record Bike Return to Original Provider

Primary Actor: The Original Bike Provider

Supporting Actor(s): None

Summary: The bike provider to whom the bike was returned uses a delivery driver to ship the bike(s) back to the original provider. The status of the bikes is tracked throughout the process until receipt of the bike(s) by the original provider. This bike provider then updates the system to confirm receipt of the bike(s). The deposit amount which was paid by the customer is then refunded by the other bike provider, and in turn is refunded to them by the original provider.

3.7 Use Case VII - Setting Bike Prices

Primary Actor: Bike Provider

Supporting Actor(s): None

Summary: The bike providers may each individually set their own daily prices for their bikes. First the bike provider accesses the system and sets the prices in the database corresponding to the types of bikes in their particular shop. The online system should then display the new prices for a type of bike from a particular bike provider when a customer searches.

3.8 Use Case VIII - Performing an In-Store Transaction

Primary Actor: Bike Shop Staff Member

Supporting Actor(s): Customer

Summary: A customer who does not book the bikes through the system but instead arrives in-person at a bike shop should still be able to rent a bike. First the customer informs the provider of the required date range and type of bikes. Upon the provider querying the system with these parameters, it should return quotes specific to their own shop. If the customer accepts one of the quotes, the deposit is collected and the provider updates the system with the status of the bikes (i.e. they are no longer available to others for the duration of the rental).

3.9 Use Case IX - Registering a New Bike On the System

Primary Actor: Bike Provider

Supporting Actor(s): None

Summary: The bike provider selects the type of bike that they wish to add from the list of bike types shared between providers. The provider then sets the daily rental price for the bike, and then the back should be available to rent for customers through the system.

Extensions: If the type of bike the provider wished to add does not already exist on the system, they can add it on by giving the full replacement value for bikes of that type. The new bike type will then be available on the shared list.

4. Use Case Diagram

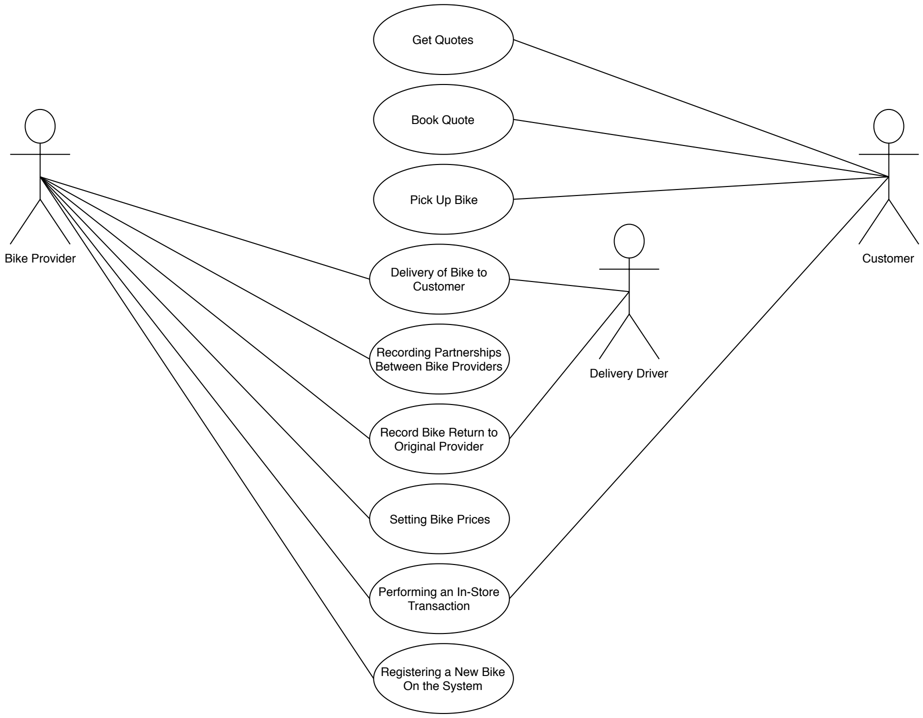
Get Quotes
Book Quote
Pick Up Bike
Delivery of Bike to Customer
Recording Partnerships Between Bike Providers
Record Bike Return to Original Provider
Setting Bike Prices
Performing an In-Store Transaction
Registering a New Bike On the System
Bike Provider
Delivery Driver
Customer

Figure 4.1: UML Use Case Diagram showing actors' relationships with the 9 identified use cases.

5. Non-Functional Requirements

The non-functional requirements can be described as the constraints on how the federalised bike rental system should function.

5.1 Security

1. The bike rental system should not allow unregistered users to access any part of the system other than the GUI provided for the customers. Registered users will be bike shop staff members and any other designated members of the project such as the software developers.

2. Payments made through the system should:

   (a) Not permanently record any card information relating to the customer

   (b) Accept the customer's payment through a secure, trusted and ratified third party gateway

   (c) Include detailed information and links to the payment method and third party gateway used

   (d) Generate and return to the customer a receipt of the payment once the correct sum has been received

5.2 Privacy

1. The system should not share any of the users' information with any other third party

2. The system should not track a user's activities on the website (by means of cookies or otherwise) without duly informing them in accordance with the GDPR Data Protection Act.

5.3 Data Retention

1. The system should reliably store the user provided data in the database.
   (a) Data stored in the database should follow the ACID database paradigm, such that any and all database transactions are: atomic, consistent, isolated and durable:
      (i) Either all of the transaction succeeds or none of it does.
      (ii) All data will be valid according to all defined rules.
      (iii) No transaction will be affected by any other transaction.
      (iv) Once a transaction is committed, it will remain in the system, regardless of whether the database subsequently crashes.
   (b) Customers should not be able to directly access and manipulate the database used to store bikes; they should only use the website as an interface to querying the database.
   (c) The collected data and uses of it should not violate any measures introduced by the GDPR Data Protection Act.

5.4 Availability

1. The system should be available to be used by customers with minimum downtime - It should aim to be fully available every day from 06:00 hrs to 21:00 hrs with no planned maintenance at this time. Planned maintenance may be allowed to occur out-with these hours but at least 24 hours notice should be given before, and the system should aim to have as much of itself operating normally even through this downtime.

5.5 Accessibility

1. The system should have a latency of no longer than 3 seconds when accessing the database and displaying the query results to the user.

2. The website should be able to handle 1 million users without a noticeable decrease in performance.

3. The website should be able to support usage by visually impaired users, by means of both support for speech-to-text and high-contrast mode.

6. Ambiguities, Subtleties, Incompleteness

The description of the system has no mention of the procedure to be carried out if renting is not completed online but rather the customer walks into a physical store to rent a bike. The assumption has been made that the system should allow the bike shop staff to easily carry out the renting procedure in-store, collecting all the necessary information and payments from the customer, and immediately updating the availability of the bikes on the system database. Another obvious solution is that this is not made possible and thereby the customer may not rent the bike physically. In this case the bike stores would simply act as physical outlets from where to collect and return the bikes which were bought online. However the first solution is clearly better as it allows people who did know know about or use the system to still rent a bike.

It is also not fully clear on how errors in the scheme's operation are handled. For example, the customer may request the bike to be delivered to an address, but the bike arrives in a different place in error. In this case with the current model the system would record the bikes as having left the shop and in transit but then the customer would not receive them: should it be too late and the bikes have to be returned, the system should deal with the fact that the bikes are not being rented and are in fact back in the shop. It would also have to deal with the payment being refunded. To deal with occurrences like this, the system could include a mechanism whereby bike shop staff are able to manually adjust the database in addition to how the system does so in normal operation. This way, any unpredictable mistakes which may not have been considered at the planning stage can be rectifies by the staff as an when they occur

Although each of the use cases have individual extensions which have described solutions, resolving a combination of one or more of these extensions by combining the individual solutions for each of the extensions may not work, as they may interfere with one another. An example of this would be in Use Case 2: Pick Up Bike, where if both the bike has a defect, and the system is down, despite the fact that both of these problems can be resolved individually as described in the extensions, combining the solutions would not solve this problem, as the solution to the defect relies on the system not being down, which is the exact scenario in the second extension. In this particular scenario, a solution would be to offer the customer a similar bike for the agreed period, and then take temporary note of this somewhere offline, and later update the system once it is back online, but the system would have to contain something to deal with the fact that the rented bike is not the one that was originally promised

The system description requires that in the event that no bikes of the user's preference are available for the given dates, the system should widen the date range criterion to 3 days before the start date and 3 days after the end date. However, it is not clear what behaviour the system should exhibit if there are no matches found in this date range either. One possible resolution would be that the system returns the message that no available bikes were found. An alternative would be that the system recursively widens the date range by the same 6 day amount up to a limit, stopping as soon as there is a match. I would be unwise to recurse without limit, as there could conceivably be a scenario where the there would never be the desired number and/or types of bikes available, and this would cause the system to crash as it recurses indefinitely.

7. Self-Assessment

7.1 Assessment

§1 Identify Stakeholders	**15%**
• Identify core stakeholders in the system	5%
• Identify additional stakeholders	5%
• Describe how the system affects each stakeholder	5%
§2 Describe System State	**10%**
• Include state essential to the operation of the system	5%
• Include additional state mentioned in the description	5%
§3 Describe Use Cases	**37%**
• Identify use cases	10%
• Describe use cases using the appropriate templates	27%
§4 Use Case Diagram	**15%**
• Correctly use UML use case notation	5%
• Include key actors and use cases	5%
• Identify connections between actors and use cases	5%
§5 Describe Non-Functional Requirements	**10%**
• Identify non-functional requirements within the context of the system	7%
• Provide reasoning for assessing non-functional requirements	3%
§6 Ambiguities and subtleties	**5%**
• Identify some ambiguities in the system description	3%
• Discuss potential options for resolution in ambiguities	2%
§7 Self-assessment	**5%**
• Attempt a reflective self-assessment linked to the assessment criteria	5%

7.2 Justificaction

Overall, our paper paints a clear picture of what the bike rental system requirements should be. The main and secondary stakeholders are clearly identified and reasoning is provided for each stakeholder. The functional requirements in the system are clearly identified and explained in detail.

Regarding the use cases, the use cases are identified in order. However, the use cases which require full detail might perhaps need clearer wording and slightly more information. The use case diagram is coherent.

The non-functional requirements are listed in enumerated lists and are clearly identified. In addition, we have explained how assessing certain requirements would be useful. Even though there would be some ambiguities in the system, we have explained possible solutions to those particular ambiguities.

7.3 Addressed Marker's Comments

In regards to addressing the marker's comments, our team has listed and described three additional stakeholders into the requirements documentation. In addition, for the 'System State' category, we have described information of what the system should record in regards to four categories: Customer, Bike Status, Provider, and Booking.

Our team has also explained the two 'full template' use cases in fuller detail. For 'Book Quote' use case, we added in three additional extensions and added in notes to make the use case have a full template. For the use case 'Pick Up Bike', we added notes for a non-functional requirement. The non-functional requirements section was changed significantly- with five categories described in detail. We added in an additional paragraph in the 'Ambiguities' section where by combining extensions from the use cases may conflict with one another. In addition, we also added in potential options to resolve those ambiguities without being vague.