

## **BIG 6 DELIVERY DOCUMENTATION**

---

# Software Requirements Specification

for

## BIG 6

Version 1.1

Prepared by

Carlyon Jones-Armour	620157742	carlyon.jonesarmour@mymona.uwi.edu
Candace Hendricks	620151637	candace.hendricks@mymona.uwi.edu
Andrew Freckleton	620146172	andrew.freckleton@mymona.uwi.edu
Brittney Rose	620146881	brittneyrose677@gmail.com
Aisha Forrester	620154561	Aisha.Forrester@gmail.com

**Course Instructor:** Dr. Ricardo Anderson

**Course:** COMP2140 – Introduction to Software Engineering

**Studio Facilitator:** Eyton Ferguson

**Date:** November 4, 2023

## TABLE OF CONTENTS

<b>1. Overall Description</b>	
1.1 Product Context and Need	1
1.2 Product Functionality	1-3
1.3 Stakeholders and Users Characteristics	3
1.4 Operating Environment	4
1.5 Design and Implementation Constraints	4
1.6 Assumptions and Dependencies	4-5
<b>2. Specific Requirements</b>	5
2.1 External Interface Requirements	5
2.1.1 Hardware Interfaces	6
2.1.2 Software Interfaces	7
2.1.3 Communications Interfaces	7
2.2 Functional Requirements	7-10
2.3 Behaviour Requirements	11
2.3.1 Use Case View	
<b>3. Other Nonfunctional Requirements</b>	12
3.1 Performance Requirements	12
3.2 Safety and Security Requirements	12
3.3 Software Quality Attributes	12-13
<b>Appendix</b>	14-19

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed	
1.1	Candace Hendricks  Andrew Freckleton  Carlyon Jones-Armour  Brittney Rose  Aisha Forrester	Final edits were made like grammar improvements, updated structure of the complete project.	12/01/12	
1.0	Candace Hendricks  Andrew Freckleton  Carlyon Jones-Armour  Brittney Rose  Aisha Forrester	Initial Draft	11/04/2023	

# 1 Overall Description

## 1.1 Product Context and Need

The proposed product is a website that our group, in collaboration with Big 6 Delivery, will be developing to enhance the company's operational efficiency and effectiveness.

Big 6 Delivery, founded by Phillip, is a well-established entity known for its emphasis on fast and efficient food delivery. Phillip's personal experience of waiting over three hours for a meal catalyzed the creation of this popular food delivery service on February 15, 2020, at UWI, Mona in Kingston, Jamaica. Initially serving the university community, the company's success led to its expansion across Kingston and St. Andrew. This growth, however, brought forth certain operational challenges, including increased customer demands, complaints of untimely deliveries, and bearers encountering delays in fulfilling orders.

As software developers, we have conducted a thorough analysis of these challenges and proposed the development of a food delivery application to address both current and potential future issues. The application will feature a user-friendly interface and incorporate real-time order-tracking capabilities. This strategic approach not only aims to elevate the overall customer experience but also to foster seamless communication between the company and its bearers. By proactively tackling these critical issues, we aspire to optimize the delivery process and contribute to the sustained success of Big 6 Delivery.

## 1.2 Product Functionality

### 1. Item Selection

- The system shall allow the user to add/remove more than one item from one or more merchant menus in their food cart to accommodate changes when ordering.
- The system shall allow the user to add, remove, and change the quantities of items in the cart.
- Upon successful validation of the customer's account, the system shall provide a means for the user to access and search for the food merchant of their choice.

### 2. Check-Out Display

- Upon successfully selecting the food items, the system shall allow the customer to view order details as selected
- The system shall allow the customer to insert specific instructions upon order, allowing the user to choose what to receive in the meal.
- The system shall display Food Merchant Cost, Delivery Amount, and GCT.
- The system shall give an option for the customer to insert a Company Coupon code/Company gift card or voucher on special occasions, if applicable.

- The system shall display the finalized total.

### 3. Delivery Method

- The system will integrate with Google Maps to determine the bearer's current location.
- The system shall access tracking data to estimate the time required for the delivery to reach the customer.
- The estimated delivery time shall be displayed to the customer during the ordering process
- The system should allow customers to give instructions to the courier of the delivery step to take when delivering.

### 4. Delivery Scheduling

- The system shall provide a scheduling option during the ordering process.
- The system shall allow users to select the time for their scheduled delivery within the day.
- The system shall ensure on-time delivery according to the scheduled time slot.
- The system should allow the user to choose the delivery time windows that align with their availability.
- The system should allow users to reschedule.

### 5. Survey

- The system shall prompt the user to complete the survey.
- The system shall provide a text box, for customer's recommendation or review.
- The system shall display a thank you message for completing the survey, if completed.

### 6. Purchase Details

- The system shall display various methods to pay for items.
- The system shall allow the user to enter and save credit card details.
- The system shall display the company's service agreement.
- The system shall display a prompt to confirm the purchase.

### 1.3 Stakeholders and Users Characteristics

- **Users**

The users that shall use this product includes but is not limited to students of The University of the West Indies. This market takes into account frequently users who may order regularly, users who may mostly utilize a particular product feature more than others, technically expert users as well as users with varying education background, experience and security or privilege level.

- **Stakeholder**

The stakeholders of this system includes the bearers, food merchants/suppliers, customers, legal entities, IT project manager and Software Developers.

- **Bearer**

The bearer is responsible for the efficient delivery of orders made by the customer and as such heavily relies on the delivery method system requirement which outlines the drop off location of the user as well as their phone number. The food merchants /suppliers of products to be sold to customers are important stakeholders that allow the user to choose from a range of products. The Food merchant cart is one that will reflect the actual merchant which was chosen by the customer to order from. There is also the customer who is the backbone of this system. They have access to Non-functional requirements such as creating their accounts, gift cards and proxy authorization. Not only this but functional requirements such as their cart, order details, choosing delivery methods and purchase details. The legal entity stakeholder provides a level of security for all parties involved in the system. Lastly, the IT project manager and software Developers would provide for maintenance of the system once it has been developed and launched.

- **Customer**

The stakeholders of the system from most to least pertinent are the customers, business analyst, project manager and project team.

## 1.4 Operating Environment

The Big 6 application will work in a specialized environment adapted to an application meant to run on Android devices such as smartphones and tablets. To guarantee seamless operation, a minimum of 1GB of RAM and a dual-core processor are required. The Big 6 software will be compatible with Android OS starting with version 5.0 Lollipop and above. This assures that it will work on a broad variety of Android devices. The app must work in tandem with other software components and services, such as accessing location data and other Google-specific services, using Google Play services payment gateway services which will allow for safe payment transactions. The app interfaces with a backend server, which handles order processing, menu retrieval, and user data storage.

## 1.5 Design and Implementation Constraints

### Constraints

The possibilities constraints to developers while creating a Big 6 will be restricted by several design and implementation difficulties such as Hardware restrictions, memory and Processor Capabilities, to ensure that the program runs well on devices with a dual-core CPU and at least 1GB of RAM, developers will need to optimize it. In particular, on older or less expensive Android devices, this restriction may necessitate careful memory management and effective coding techniques to maintain smooth operation. Interoperability of Operating Systems, and Android Version Compatibility allow Android devices running Android 5.0 Lollipop and later are intended to use the app. This restriction restricts the usage of features and APIs that were added to more current Android versions, which may force developers to offer other solutions or backward compatibility for older devices. Integrating the application with third-party payment gateways and Google Play Services is necessary for location-related functionalities. Developers must guarantee the correct handling of user data and payment security following industry standards, and this constraint may necessitate adherence to specific APIs, SDKs, and protocols provided by these services.

## 1.6 Assumptions and Dependencies

**Below are the assumptions and dependencies:**

- Assuming a specific number of merchants and items: If the system has a specific list of food merchants(KFC, Burger King, etc..) then realistically there will arise other food merchants and the system must then be updated to reflect them. A dependency here would include the fact that the menus of these food merchants will likely change and the system must be up to date with these changes.
- Assuming Google Map efficiency: The customer may use this feature for delivery estimation. However, if there is a case where it is unavailable or there are changes in Google Maps API, then this requirement will be severely affected.
- Assuming customer's location reliability: Customers may not enter the correct address, or location and this may lead to further issues when the bearer arrives at the location on the customer's account. Additionally, clear instructions may not be given by the customer directing the bearer on what to do when they arrive. For example, ringing the doorbell or waiting at a gate.



- Assuming that there will always be bearers: Realistically, there may be a shortage of available bearers which may significantly affect customer satisfaction and even lessen customers.
- Assuming reliability of payment gateways: The proper functioning of online systems may not always be smooth sailing and as such there should be anticipated unreliability and unavailability of these gateways.
- Assuming regulatory compliance: Relevant regulations such as food safety and data privacy are to be taken into account. Any changes here will limit the system's operation and it must be flexible enough to accommodate these changes

## 2 Specific Requirements

### 2.1 External Interface Requirements

#### 2.1.1 Hardware Interfaces

##### GPS Interface:

- *Logical Features:* This interface works with the hardware components of the GPS (Global Positioning System) that are present in tablets and smartphones. In order to precisely determine the user's location, "Big 6" uses GPS data, which is essential for location-based services and delivery tracking.
- *Physical attributes:* The GPS interface uses device-specific APIs (like iOS Core Location or Android Location Services) to access the device's GPS sensor. It uses information from the GPS hardware to give exact location data.

##### Camera Interface:

- *Logical Features:* The camera interface enables communication between the application and the hardware of the device's camera. It can be used for a number of things, such as taking pictures of food items, scanning QR codes, and confirming the identity of delivery drivers.
- *Physical characteristics:* Platform-specific APIs, such as the iOS AVFoundation framework or the Android Camera2 API, are used by the camera interface to interface with the device's camera sensor. It manages camera operations and takes pictures.

#### 2.1.2 Software Interfaces

##### Data Item and Messages

- *Resource Management Request (Memory Allocation):* The Big 6 app connects with the Android operating system to request memory allocation when a user opens it on their Android device. This guarantees that the application has enough memory to function properly. For example, when a user views a restaurant menu with crisp photos and thorough descriptions, the application needs more memory in order to properly load and display the content.
- *User Interaction Data (Touch Events):* The Android device records touch events when the user interacts with the application, such as by navigating the menu or choosing items for an order. For instance, the app receives touch event data, including touch coordinates, when a user taps on a particular menu item to view more information or add it to their cart. This allows the app to accurately register the user's input.
- *Display Output (Rendering UI):* The food delivery app interacts with the Android OS to display its user interface on the device's screen when the customer navigates through the app to place an order. The application transmits data items that define the design and visual elements for the menu, order summary, and delivery choices. The Android operating system makes sure that this

data is appropriately shown on the user's screen by adjusting for screen resolution and offering rendering performance feedback. Through the Android operating system, the application requests access to particular hardware elements, such as the device's camera and GPS. To open and operate these devices, messages are sent.

- *Network Communication (Placing an Order):* The Android OS starts a network communication when a customer places an order within the app. The application transmits network requests containing order details, delivery address and payment details to the server of the food delivery service. After processing the request, the server provides data items such as the order confirmation, the estimated time of delivery, and the status of the payment authorization.
- *Data Sharing Mechanism:* To access payment gateways, location services, and restaurant menus, "Big 6" can integrate with several external services and APIs. For instance, it can guarantee that customers receive their orders on schedule by using the Google Maps API to provide precise location and delivery tracking services. To store and retrieve customer profiles, order histories, and restaurant details, the app can establish a connection to a database system. Order tracking, fast access to menu items, and personalized user experiences are made possible by this. To handle user accounts, retrieve restaurant data, and process orders, "Big 6" talks with backend servers. User profiles, restaurant details, and order details are among the data shared with the backend. This communication guarantees that the app can process orders quickly and provide the most recent menu items.

### 2.1.3 Communications Interfaces

For a food delivery app like "Big 6," there are numerous standards and protocols that must be followed in order to ensure secure and effective data exchange. The application will communicate with external servers using the Hypertext Transfer Protocol (HTTP) and its secure variant, HTTPS. Sending and receiving data for order processing, menu retrieval, and user authentication requires this. Data security and encryption while in transit are guaranteed by HTTPS. Communication security will be needed for payment transactions. To ensure the secure handling of credit card data, the app will employ industry-standard payment protocols such as Payment Card Industry Data Security Standard (PCI DSS) and tokenization. Sensitive payment information will be shielded during transmission using encryption and secure communication channels.

"Big 6" will make use of push notification-specific communication standards, like Firebase Cloud Messaging (FCM) for Android. By enabling real-time updates and alerts to be sent to users' devices, these standards guarantee that users receive promotional messages and order status updates. In order to offer a responsive user experience, the application will strive for effective data transfer rates. Order status, menu items, and user data will all be updated in real-time through synchronization mechanisms, guaranteeing that customers always have access to the most recent information.

## 2.2 Functional Requirements

**Requirement ID#: 1.** Selecting options from the Food Merchant Cart

**Use Case:** Edit customer's food cart

**Rationale:** Customers may need to add/remove more than one item from one or more merchant menus in their food cart. This is to accommodate a change of mind in ordering.

**User Requirement:** The system shall allow the user to add, remove, and change the quantities of items in the cart

**System Requirements:**

**1.1:** Upon successful validation of the customer's account, the system shall provide a means for the user to access and search for the food merchant of their choice.

**1.2:** The system shall provide a means for the user to add/ remove food items from the cart.

**Acceptance Criteria :**

1. 100% of the order details added must be saved
2. All modifications must be logged

**Relates to/Dependencies:** Presence of user's account. **Priority:** High. **Team Owner:** Brittney Rose

**Requirement ID#: 2.** Check Out-Order Information

**Use Case:** Display Order Information

**Rationale:** Customers may wish to redo their order after viewing the order details and the cost applied.

**User Requirement:** The system shall allow users to view order details and costs in the cart or restart the order if they wish.

**System Requirements:**

**2.1:** Upon successfully selecting the food items, the system shall allow the customer to view order details as selected

**2.2:** The system shall allow the customer to insert specific instructions upon order, allowing the user to choose what to receive in the meal.

**2.3:** The system shall display Food Merchant Cost, Delivery Amount, and GCT.

**2.4:** The system shall give an option for the customer to insert a Company Coupon code/Company gift card or voucher on special occasions, if applicable.

**2.5:** The system shall display the finalized total.

**Acceptance Criteria:** All data entered by the user must be saved

**Relates to/Dependencies:** Presence of user's order. **Priority:** High. **Team Owner:** Carlyon Jones-Amour

### **Requirement ID#: 3. Delivery Method**

**Use Case:** Choose preferred delivery option and assist in order tracking

**Rationale:** Customers may want an update on order status based on estimated delivery time given and direct bearers to the destination properly. This is for the convenience of the customer and to manage the expectation of the delivery.

**User Requirement:** The system should allow the customer to instruct how the delivery will be done.

#### **System Requirements:**

- 3.1 The System shall integrate with Google Maps to determine the bearer's current location.
- 3.2 The system shall access tracking data to estimate the time required for the delivery to reach the customer.
- 3.3 The estimated delivery time shall be displayed to the customer during the ordering process
- 3.4 The system shall allow customers to give instructions to the courier of the delivery step to take when delivering.

#### **Acceptance Criteria**

- 1. The user's location should be updated in near-real-time, reflecting any movements or changes.
- 2. The system should clearly indicate when the driver is en route to the delivery location.

**Relates to/Dependencies:** None **Priority:** High **Team Owner:** Aisha Forrester

### **Requirement ID#: 4. Delivery Scheduling**

**Use Case:** Schedule the time for delivery

**Rationale:** To accommodate users' busy schedules and pre-planned events, the system should allow for the scheduling of deliveries at specific times.

**User Requirement:** The system shall enable users to schedule food deliveries for specific date and time slots.

#### **System Requirements:**

- 4.1 The system shall provide a scheduling option during the ordering process.
- 4.2 Users shall be able to select the time for their scheduled delivery within the day

**4.3** The system shall ensure on-time delivery according to the scheduled time slot.

**4.4** To accommodate users' specific schedules, the system should allow them to choose delivery time windows that align with their availability.

**4.5** The system should allow users to reschedule.

**Acceptance Criteria:**

1. Users should be able to schedule deliveries during the order placement.
2. Users must have the option to select a specific date and time for delivery.
3. The system should ensure that deliveries are made within the scheduled time slot.

**Relates to/Dependencies:** None **Priority:** High **Team Owner:** Andrew Freckleton

**Requirement ID#: 5. Survey**

**Use Case:** Collect customer's feedback on purchases.

**Rationale:** To allow customers to make more informed decisions, the system should allow for the customers to describe their satisfaction with their purchases.

**User Requirement:** The system shall enable users to describe their experience and satisfaction/dissatisfaction with their purchase.

**System Requirements:**

**5.1** The system shall prompt the user to complete the survey.

**5.2** The system shall provide a text box, for customer's recommendation or review.

**5.3** The system shall display a thank you message for completing the survey, if completed.

**Acceptance Criteria:** Users should be able to provide feedback on purchases.

**Relates to/Dependencies:** None **Priority:** Low **Team Owner:** Brittney Rose

**Requirement ID#: 6. Purchase Details**

**Use Case:** Allow users to select payment methods.

**Rationale:** To allow customers to choose a preferred method to pay for items.

**User Requirement:** The system shall enable users to choose a method to pay for items selected.

**System Requirements:**

**6.1** The system shall display various methods to pay for items.

**6.2** The system shall allow the user to enter and save credit card details.

**6.3** The system shall display the company's service agreement.

**6.4** The system shall display a prompt to confirm the purchase.

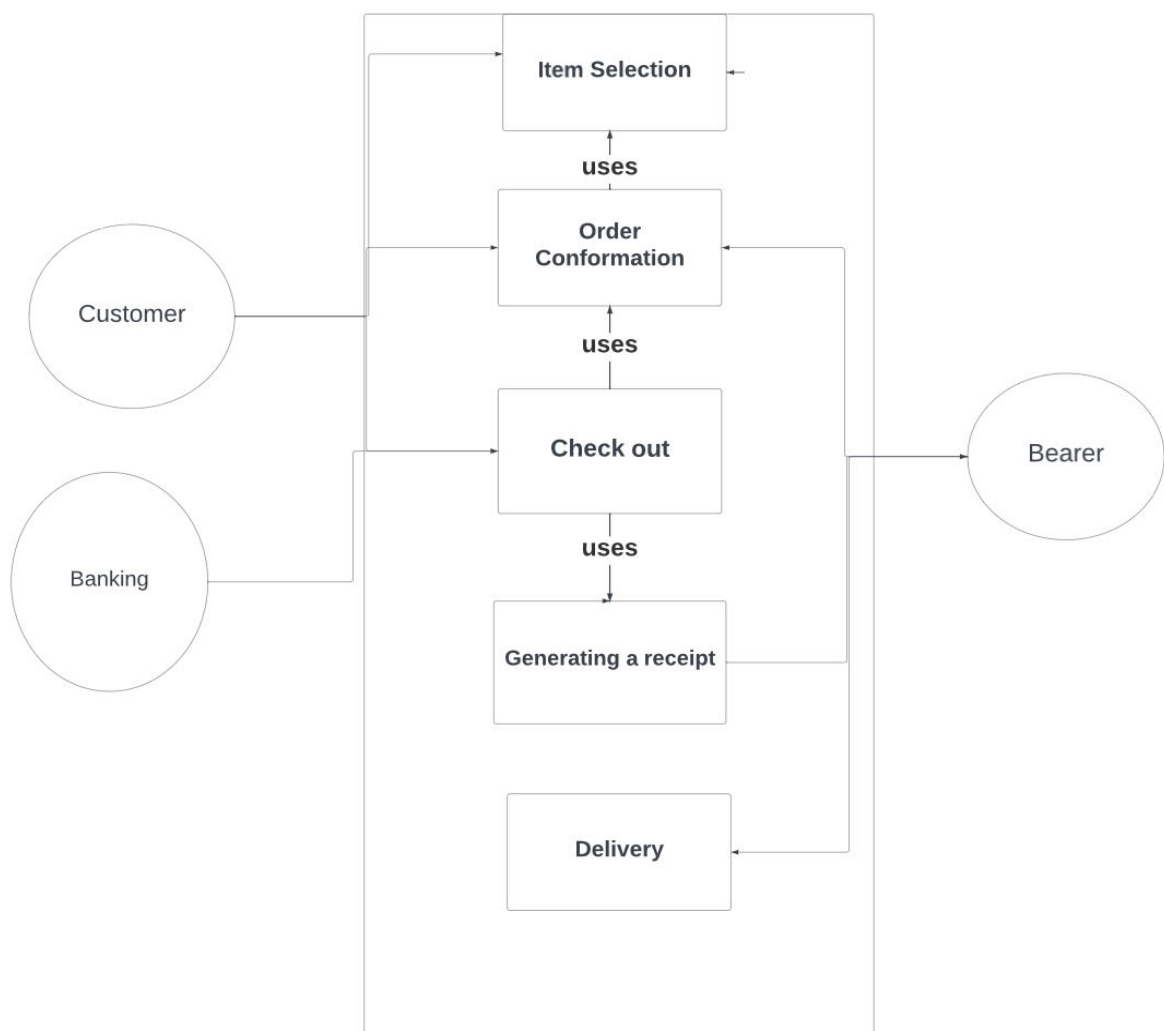
**Acceptance Criteria:**

1.Users should be able to select payment methods.

**Relates to/Dependencies:** Item Selection **Priority:** High **Team Owner:** Candice Hendricks

## 2.3 Behaviour Requirements

### 2.3.1 Use Case View






## 3 Other Non-functional Requirements

### 3.1 Performance Requirements

**Requirement 1:** Content on the product will load within 5 seconds for the majority of users.

**Rationale:** Content, such as: restaurant/merchant descriptions, images and menus, that load slowly can negatively affect the users experience leading to the user's frustration.

**Requirement 2:** Users shall receive delivery estimation time within 10 seconds of order.

**Rationale:** This will enhance the users experience as an estimated time can help to avoid frustration and uncertainty regarding the delivery.

**Requirement 3:** Users shall receive a prompt confirming their order within 30 seconds.

**Rationale:** This will prevent the users' uncertainty regarding whether the order was successful.

### 3.2 Safety and Security Requirements

The client expects a high level of security as the product deals with the users' financial and personal data. Some of the safety requirements needed are:

1. The product must adhere to the necessary privacy standards for financial information. The system shall not store private or sensitive data unless specified by the user. If it is stored, the data must be encrypted.
2. The product should implement user identity authentication and authorization so that users can only have access to their specific roles, such as: customer or bearer.

Other major security requirements include:

- Explicitly stating the privacy policy for the collection and use of the user's data.
- Ensuring that user passwords are strong by specifying minimum password lengths and complexity.

### 3.3 Software Quality Attributes

- This software shall be maintainable by modularization. This means breaking varying parts of the system into manageable chunks. Not only this but ensuring to use version control tools like GitHub to have more control over the code documentation and collaboration.
- The software shall be flexible by the use of accurate parameterization. This allows for more control over the behaviour of the system and changes can easily be made without affecting the entire system.
- The software shall be testable by accurate unit testing. This will include writing tests for functions individually and executing them in isolation.
- This software shall be reliable by ensuring to gather clear and concise requirements from stakeholders and ensure error handling so the system does not break under pressure.



## Appendix

*Dates of interview and who was interviewed, questionnaires, interview logs, observation activities, client responses*

### Interview Questions

#### Interview: #1

*Date: September 15, 2023*

*Interviewers: Carylon-Jones Armour, Candace Hendricks*

*Interviewee: Mr. Philip - C.E.O*

- 1. Can you provide us with a comprehensive overview of Big 6 Delivery, including its mission, target audience, and key operations?*
- 2. From your perspective, what are the most frequently reported pain points that customers experience when using Big 6 Delivery's services?*
- 3. Are there any upcoming developments or expansions in the pipeline for Big 6 Delivery that might potentially introduce new challenges or issues that we should consider in the website design?*
- 4. Could you share insights into any pain points that employees encounter with the current system, and how these might be contributing to delays in the delivery process?*
- 5. In your opinion, what are some key features or functionalities that you believe could be incorporated into the website to significantly enhance the efficiency of Big 6 Delivery's operations?*
- 6. a. How would you envision the ideal customer experience when using the website for placing and tracking orders?*  
*b. How can we ensure that this aligns with the company's overall goals and values?*
- 7. a. In your opinion, if there was a feature that could be incorporated into the current website to expedite deliveries, what would that feature be?*  
*b. What do you believe would be its impact on enhancing the efficiency of our delivery process?*

#### Interview: #2, #3

*Date: October 1, 2023*

*Interviewer: Carlyon Jones Armour - Interviewee: Ms. Jimel Ferguson (Big 6 customer)*

*Date: October 12, 2023*

*Interviewer: Candace Hendricks - Interviewee: Ms. Dennis Clarke (Big 6 customer)*

- 1. How often do you use Big 6's services?*
- 2. What are the most important factors for you when using a food delivery service like Big 6 Delivery?*
- 3. Can you describe any specific pain points or challenges you've encountered while using the current Big 6 Delivery service?*

4. *Are there any features or functionalities that you wish were available in the app to enhance your experience?*
5. *How would you rate the overall user-friendliness and navigation of the current Big 6 Delivery app?*
6. *Can you share any positive experiences or standout features from using the Big 6 Delivery service?*
7. *What additional information or features would you find helpful when placing orders through the app?*
8. *Have you ever experienced delays or issues with order deliveries, and if so, what do you think might have caused them?*
9. *Are there any specific preferences or options you would like to see in the app for customizing orders?*
10. *How important is real-time order tracking to you when using a food delivery service?*

Interview #4, #5:

NB: To maintain confidentiality, employees were assigned pseudonyms for the purpose of this interview log in the SRS document.

Date: September 21, 2023

Interviewer: Andrew Freckleton -Interviewee: Mr. John Brown (employee)

Interviewer: Brittney Rose -Interviewee: Mr. James Green (employee)

1. *Can you describe your role and responsibilities within Big 6 Delivery, and how it contributes to the overall operations of the company?*
2. *From your perspective, what are some of the most significant challenges or pain points that you face in your day-to-day tasks?*
3. *How do you currently interact with the existing system for order management and delivery, and are there any specific features or functionalities that you find particularly useful or frustrating?*
4. *Have there been instances where you've encountered delays in order processing or delivery? If so, can you provide some context or potential causes for these delays?*
5. *In your opinion, what improvements or enhancements could be made to the current system that would help streamline operations and improve efficiency?*
6. *Are there any additional functionalities or tools that you believe would be beneficial in improving communication and coordination among team members?*
7. *Can you share any specific examples of successful initiatives or practices within Big 6 Delivery that you believe contribute to the company's overall success and customer satisfaction?*

Questionnaire:

*Instructions: Kindly provide your responses to the following questions with careful consideration, understanding that your input will be handled with the highest level of confidentiality and held in the strictest confidence.*

- 1. How would you rate the current user-friendliness of the existing order management system?*
  - a) Very user-friendly*
  - b) Somewhat user-friendly*
  - c) Neutral*
  - d) Somewhat difficult to use*
  - e) Very difficult to use*
- 2. How satisfied are you with the current level of communication and coordination among team members?*
  - a) Very satisfied*
  - b) Satisfied*
  - c) Neutral*
  - d) Dissatisfied*
  - e) Very dissatisfied*
- 3. Do you feel that the current system adequately addresses customer feedback and concerns?*
  - a) Yes, completely*
  - b) Yes, to some extent*
  - c) Neutral*
  - d) No, not sufficiently*
  - e) No, not at all*
- 4. How often do you encounter delays in order processing or delivery in your role?*
  - a) Very rarely*
  - b) Occasionally*
  - c) Sometimes*
  - d) Frequently*
  - e) Very frequently*
- 5. How important is real-time order tracking to you in ensuring efficient delivery operations?*
  - a) Very important*
  - b) Important*
  - c) Neutral*
  - d) Not very important*
  - e) Not important at all*
- 6. Are there specific types of data or information that you believe should be prioritized in the order management system for efficient decision-making?*
  - a) Customer information*
  - b) Order status and details*
  - c) Delivery schedules and routes*
  - d) Inventory levels*
  - e) Other (please specify)*
- 7. How would you rate the current system's ability to handle peak demand periods or high traffic situations?*
  - a) Excellent*
  - b) Good*

- c) Fair*
- d) Poor*
- e) Very poor*

*8. Would you prefer a mobile application or a web-based platform for order management and delivery coordination?*

- a) Mobile application*
- b) Web-based platform*
- c) No preference*

*9. Which of the following features do you believe would significantly improve the efficiency of order processing and delivery at Big 6 Delivery? (Select all that apply)*

- a) Real-time order tracking*
- b) Automated route optimization*
- c) Customer feedback integration*
- d) Inventory management tools*
- e) Employee performance analytics*

*10. How would you rate the current level of employee training and onboarding for using the existing order management system?*

- a) Excellent*
- b) Good*
- c) Fair*
- d) Poor*
- e) Very poor*

*11. How confident are you in the security measures in place to protect sensitive customer and company data within the current system?*

- a) Very confident*
- b) Confident*
- c) Neutral*
- d) Not very confident*
- e) Not confident at all*

*12. Are there any specific compliance or regulatory requirements that you believe the order management system should adhere to?*

*13. Which aspects of the current system's user interface do you find most effective in facilitating your tasks?*

- a) Navigation*
- b) Order tracking*
- c) Data entry*
- d) Reporting and analytics*
- e) Other (please specify)*

*14. How would you rate the current system's ability to adapt to changing market trends and customer preferences?*

- a) Excellent*
- b) Good*
- c) Fair*
- d) Poor*
- e) Very poor*

*15. What role do you believe data analytics and reporting should play in optimizing Big 6 Delivery's order management process?*

- a) Critical role
- b) Important role
- c) Neutral
- d) Limited role
- e) No role

16. Are there any specific integration requirements you believe the order management system should have with other existing tools or platforms used by Big 6 Delivery?

17. Can you describe your typical workflow or tasks related to Big 6 Delivery operations?

18. What do you believe are the most critical challenges faced by Big 6 Delivery in terms of order processing and delivery?

19. Are there any specific features or functionalities that you feel are essential for an efficient order management system?

20. Please share any additional suggestions or insights you have that could improve the order processing and delivery process at Big 6 Delivery.

#### Observation Activities:

##### LOGS:

Time: 9th October 2023

Action: Order Food

Observation:

- Due to the use of university restaurants there was an absence of some of the restaurants at certain times when other competing delivery services had access.
- The waiting time took more than an hour.
- Reported wait time updates did not reflect reality.

Time: 12th October 2023

Action: Order Food

Observation:

- .Website responsiveness was slower than usual.
- Inconsistencies in web site display were observed due to device specs.
- The waiting time took more than an hour and a half.
- Reported wait time updates did not reflect reality.
- Displayed features (e.g Star ratings) were non functioning

Time: 15th October 2023

Action: Web Interface Troubleshooting

Observation:

- Inconsistent theme
- Inconsistent Image quality
- Missing content (text/images/pages)
- Beginner level design

Time: 19th October 2023

Action: Customer communication and support

Observation:

- Limited support channels
- Unappealing presentation of chatbox
- Unnecessary links that leads to the same page
- Might not be appropriate for persons who are not tech-savvy





---

# *Software Design Specification*

*for*

## *Big 6 Food Delivery Application*

*Version 1.1*

*Prepared by*

*Candace Hendricks  
Andrew Freckleton*

*6201516371  
620146172*

*<e-mail>  
<andrew.freckleton02@gmail.  
com>*

*Carlyon Jones Armour  
Brittney Rose  
Aisha Forrester*

*620157742  
620146881  
62014561*

*<carlyonja@gmail.com>  
brittneyrose677@gmail.com  
<e-mail>*

***Course Instructor:*** *Claudia Allen*

***Course:*** *COMP2140 – Introduction to Software Engineering*

***Studio Facilitator:*** *Mr. Eyton Ferguson*

***Date:*** *November 18, 2023*

*Using this template:*

*<items in parentheses and or in blue are descriptions of what is to be done or examples. You should remove them and replace with details of your project. Delete this note before submission >*

## ***TABLE of CONTENTS***

Project Overview.....	1
Architectural Design.....	2
General Constraints.....	3
Alternatives Considered.....	4
System Architecture Diagram.....	5
Architectural Description.....	6
Architecture Justification.....	7
Architecture Decomposition.....	8
Component Decomposition.....	9
Structural Design.....	10
Design Notes.....	11

## 1.0 Project Overview

*<Describe the client, the problem to be solved, and the intended users. Explain the context in which your software will be used, i.e. the big picture. (1 – 3 paragraphs). THIS SECTION SHOULD REFLECT SOME INFORMATION FROM THE SRS. It may also be useful to include the user requirements (not system requirements) and the non-functional requirements that are important to the architecture decisions you will make later. >*

The client for this software development project is Big 6, a well-established entity founded by Phillip, known for its emphasis on fast and efficient food delivery. Phillip's personal experience of waiting over three hours for a meal catalysed the creation of this popular food delivery service on February 15, 2020, at UWI, Mona in Kingston, Jamaica. Initially serving the university community, the company's success led to its expansion across Kingston and St. Andrew. This growth, however, brought forth certain operational challenges, including increased customer demands, complaints of untimely deliveries, and bearers encountering delays in fulfilling orders. In response to these challenges, a comprehensive analysis has been conducted, leading to the decision to develop a food delivery application. The primary objective is to streamline the ordering and delivery processes, enhance customer satisfaction, and optimize overall operational efficiency for Big 6 in the face of both current and potential future challenges.

The intended users of the application encompass bearers responsible for order fulfillment, customers of Big 6, and the Chief Executive Officer (CEO). Bearers will utilize the application to efficiently manage and fulfill delivery orders, leveraging features like prioritization and route optimization. Customers, on the other hand, will interact with the platform to easily place orders, track their deliveries in real-time, and receive timely updates through a notification system. The CEO will oversee the application, gaining insights into the company's delivery operations and ensuring the software aligns with strategic goals.

User requirements for the application include seamless order placement and tracking for customers, an efficient interface for bearers to manage orders, a robust notification system for timely updates, and secure payment processing. Additionally, there is a need for secure user authentication mechanisms to protect sensitive information. Non-functional requirements encompass aspects such as performance, scalability, reliability, usability, and availability. The application must handle a large number of concurrent users, scale to accommodate future growth, be reliable with a backup and recovery system, feature intuitive user interfaces, and be available 24/7 to meet customer demands. These requirements will guide the architecture decisions for the successful development and deployment of the food delivery application.

## **2.0 Architectural Design**

### **2.1 General Constraints**

*<Describe global limitations or constraints that have a significant impact on your system design. Examples include hardware and software environments, interface requirements, external data representations, performance requirements, network requirements, etc. (1 – 3 paragraphs)>*

The architectural design of the Big 6 Food Delivery Application must contend with several global constraints that significantly influence the system's development. Firstly, given the nature of a food delivery service, performance requirements are paramount. The application must operate seamlessly under high user loads, handling concurrent requests efficiently to ensure a smooth and timely ordering and delivery experience. This necessitates a robust and scalable infrastructure that can adapt to fluctuations in demand.

Another critical constraint is the need for secure and reliable payment processing. The system must adhere to industry standards for financial transactions, implementing encryption protocols and secure authentication mechanisms to protect user payment information. Compliance with relevant payment processing regulations and standards is imperative to ensure the trust and confidence of both customers and bearers.

Additionally, the architectural design must consider network requirements, particularly in regions where Big 6 operates. The application should be designed to perform optimally in areas with varying network conditions, ensuring a consistent user experience even in regions with lower connectivity. Moreover, given the expansion plans of Big 6, the architecture must be flexible and capable of accommodating future integrations, such as partnerships with new restaurants or changes in the delivery network. These constraints will guide the design decisions, influencing the selection of technologies, frameworks, and infrastructure components to meet the specific challenges posed by the operational context of Big 6.

### **2.2 Alternatives Considered**

*<Discuss at least 2 alternative architectural PATTERN/MODEL considered and explain why they were not as appropriate as the one chosen. (1 – 4 paragraphs). Use the details about each pattern i.e. when used, advantages, disadvantages to discuss why they are considered>*

In the evaluation of alternative architectural patterns for the Big 6 Food Delivery Application, the Repository Pattern and the Client-Server Pattern were considered, with the following analysis:

#### **Repository Pattern:**

The Repository Pattern was initially explored due to its strengths in promoting a centralised data access layer, providing a structured and organised approach to handling data interactions. This pattern encourages a separation of concerns, facilitating maintainability and testability in the application's data access logic. However, it became apparent that the Repository Pattern, while advantageous for certain types of applications, could introduce potential bottlenecks in scenarios requiring rapid and concurrent access to real-time data.

This limitation influenced the decision to explore alternative patterns better suited for the dynamic nature of food delivery operations.

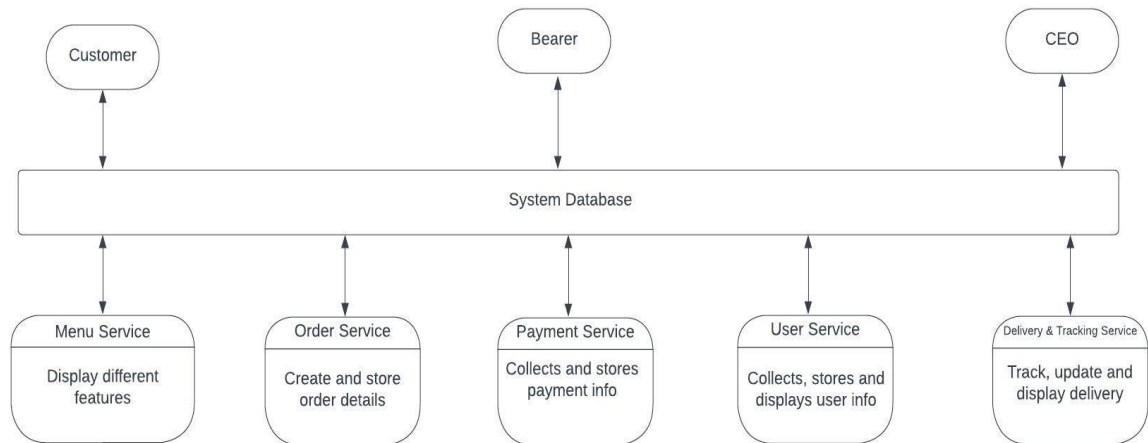
#### Client-Server Pattern:

The selected architectural pattern for the Big 6 Food Delivery Application is the Client-Server Pattern, which offers a range of advantages aligned with the project's requirements. One notable benefit is the clear separation of concerns between clients (customer, C.E.O, and bearer interfaces) and the server. This segregation enables the development of specialised client interfaces tailored to the distinct needs of customers and bearers, enhancing user experiences. Moreover, the Client-Server Pattern supports efficient communication between clients and the server, ensuring real-time updates, quick response times, and improved overall system performance.

Additionally, the Client-Server Pattern facilitates scalability, allowing the application to handle a large number of concurrent users seamlessly. The server-centric approach centralises critical business logic and data management, making it easier to maintain and scale the system as the user base grows. This scalability is crucial for accommodating the expanding operations of Big 6 and ensuring a responsive and reliable food delivery service.

In summary, while the Repository Pattern offers advantages in terms of structured data access, the dynamic and real-time requirements of a food delivery service led to the selection of the Client-Server Pattern. The latter's benefits in terms of separation of concerns, efficient communication, and scalability make it the more suitable architectural choice for the Big 6 Food Delivery Application.

## 2.3 System Architecture Diagram



### 2.3.1 Architectural Description

#### 1. Main Menu Module:

1.1 Explore and Search: Users should be able to explore local restaurant menus.

Users should be able to find certain foods or restaurants using the search capability.

1.2 Menu Item Specifics: Each menu item should provide specific information such as a description, price, ingredients, and nutritional information.

Users can view menu item photos.

#### 2. User Module:

2.1 Registration and Authentication: Users should be able to register for the app using their email, phone, or social network credentials. The system must include secure authentication procedures (for example, passwords and fingerprints).

2.2 Manage User Profiles: Users should be able to create, update, and delete their profiles. Personal information, delivery addresses, and preferred payment methods should all be included in the profile.

2.3 Order History: Users should be able to view a full order history, which includes previous orders, receipts, and order statuses.

Users should be able to repeat past orders in the system.

#### 3. Order Module:

3.1 Place Order: Users should be able to add products to their shopping basket and then place an order. Before completing the purchase, the system should provide an order summary.

3.2 Delivery Options: Users can select between delivery and pickup. Selecting a delivery time slot is one of the delivery possibilities.

3.3 Payment: Users can pay for orders securely using a variety of payment options (credit card, digital wallets, etc.). If applicable, the system should support cash on delivery.

3.4 Order Confirmation: Users will receive an order confirmation email that includes information such as expected delivery time and order number. Email and push notifications should be used to send confirmation.

#### 4. Tracking Module:

4.1 Real-time Order Tracking: Users can view the status of their orders in real-time. The system sends information such as order preparation, ready for delivery, and delivery.

4.2 anticipated Time of delivery (ETA): Customers are given an anticipated delivery time for their orders. The ETA should be dynamically updated based on the delivery person's current position.

4.3 Delivery Person Information: Users can examine delivery person information, such as name and contact information.

#### 5.Payment Module:

5.1 Secure Transactions: The system must ensure that payment transactions are secure and encrypted. Payment receipts should be sent to users via email and in the app.

5.2 Multiple Payment Methods: Accept a variety of payment methods, such as credit/debit cards, digital wallets, and other local payment methods.

5.3 Promotions and Discounts: During the checkout process, users can use promo codes and discounts. The system should be in charge of validating and implementing promotional offers.

### **2.3.2 Architecture Justification**

*<Give justification for the architecture presented. Why was this the best way to organise the components among other options, what makes this architecture a good approach for your system>  
0*

*Use design pattern and why it is best to use*

The selection of a Client-Server design pattern for the Big 6 Food Delivery Application is grounded in its suitability to efficiently organize and distribute the system's functionalities. This architectural choice was made after careful consideration of various options, and it emerges as the most effective approach for the following reasons:

#### 1. Organized Service Distribution:

- The Client-Server pattern allows the system's functionalities to be organized into distinct services, with each service residing on a different server. This organization aligns with the modular structure identified in the Software Design Specification, where components like OrderProcessor, PaymentProcessor, UserProfileManager, and others are logically separated. Such a modular arrangement enhances maintainability and promotes a clear separation of concerns.

#### 2. User Roles and Responsiveness:

- The system involves different user roles, including bearers, the CEO, and customers. By adopting a Client-Server architecture, dedicated client interfaces can be developed for each user role. This ensures that interfaces are tailored to the specific needs of bearers, the CEO, and customers, providing a more responsive and user-friendly experience. Each client



interacts with the server to perform role-specific operations, contributing to a more intuitive and efficient user interaction.

### 3. Scalability and Distribution:

- The Client-Server pattern accommodates scalability requirements effectively. The servers, hosting various services, can be distributed across a network. This enables the system to handle a large number of concurrent users and increase demands by efficiently distributing the workload. As the company expands, additional servers can be added to support the growing user base and maintain optimal performance.

### 4. Centralized Functionality:

- The centralization of core functionalities on the server side ensures that general functionalities are available to all clients without redundant implementation. Common services like order processing, payment handling, and user profile management are implemented once on the server, reducing duplication of effort and enhancing system consistency.

### 5. Real-time Communication and Updates:

- The Client-Server pattern facilitates real-time communication between clients and the server. This is crucial for a food delivery application where timely updates on order status, delivery estimates, and other notifications are essential. The centralized nature of the server allows for efficient communication, ensuring that information is consistently updated and synchronized across all clients.

In summary, the Client-Server architecture is the preferred choice for the Big 6 Food Delivery Application due to its ability to organize functionalities into services, accommodate different user roles, support scalability, centralize core functionalities, and facilitate real-time communication. This approach enhances system responsiveness, maintainability, and overall user experience in alignment with the specific requirements outlined in the Software Design Specification.

## ***3.0 Architecture Decomposition..***

### 3.1 Component Decomposition – *Classes OR Modules* <not both so delete one from this heading>

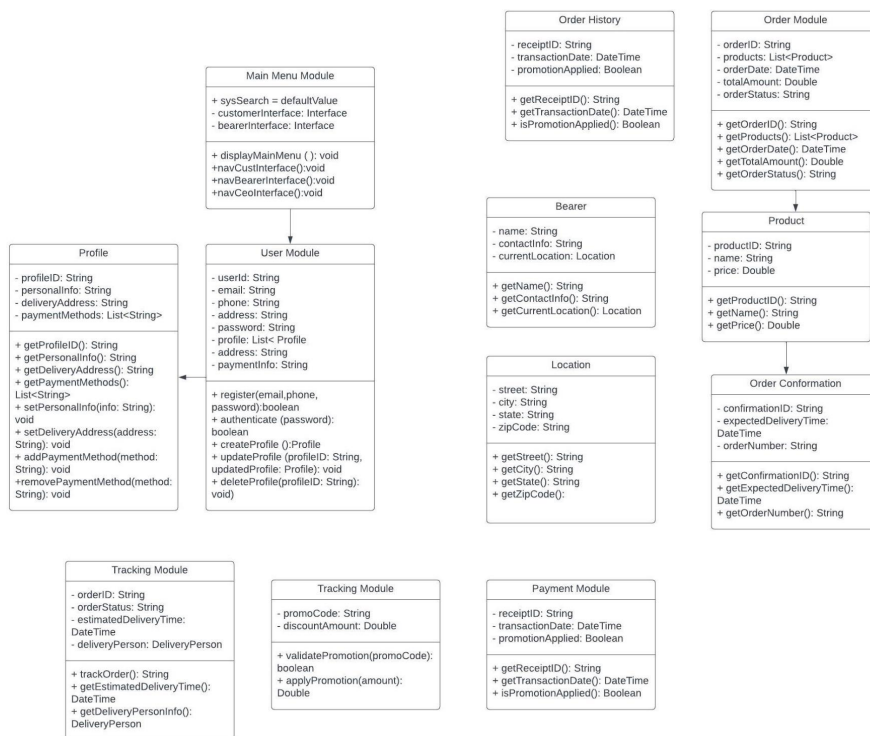
<Using the below table, provide the name and description for each class OR module that will be used to fulfil the design of each component specified in 2.3. For traceability, you should include the list of requirements IDs which will be fulfilled by the specified component, and by extension the class or modules specified. Once any part of the requirement is being addressed in the component, it should be listed (you should use system requirements ID in the case where the entire user requirement is not being addressed in the component).>

Example ONLY:

Requirement ID	Architecture Component	Class / <Module> Name	Description
Req. 1 Req. 3 Req. 6.1	User Interface	SystemUI  SystemTXT  SystemSearch	Provides the layout and functions for users to manage access to various features upon system startup  This will provide a minimal text-based interface that will allow system setup, to create the initial system users and modify administrative privileges.  Enable users to conduct searches using specific keywords based on their profiles. For customers, it allows searching for available companies based on order times. For bearers, it facilitates searching for available orders to deliver or searching orders by order number for updating customer profiles on delivery. The CEO can use it to check sales, orders for the day, and other relevant information.
Req. 2.3 Req 1.4 Safety & Security	Authorization	AdminAuth	Handles secure user authentication, ensuring the protection of sensitive information and access control based on user roles and identity.
	User Profile	ProfileSys	Manages user profiles for both bearers and customers, enabling the updating of account information and displaying personalised data from the database based on the user's role.
	Order Management	OrderProcess	Manages the processing of incoming orders, including order prioritization, assignment to bearers, and updating order statuses in real-time.
	Database Management	DatabaseManager	Manages the storage and retrieval of data in the application database, ensuring data integrity, security, and optimal performance.

Requirement ID	Architecture Component	Class / <Module> Name	Description
	Notification System	Notif	Manages a robust notification system, sending timely updates to both customers and bearers regarding order status, delivery estimates, and other relevant alerts.
	Payment Processing	PaymentProcessor	Facilitates secure and reliable payment processing, adhering to industry standards for financial transactions and ensuring the confidentiality of user payment information
	C.E.O Dashboard	DashboardMa	Provides a dedicated interface for the CEO to monitor and analyze key performance indicators, sales data, and other relevant metrics for strategic decision-making.
	GiftCertificationAcceptor	GiftCert	Facilitates the purchase and redemption of gift certificates within the application, allowing customers to buy and use certificates for their orders.
	Delivery Scheduling	ScheduleSys	Implements a calendar system for delivery scheduling, displaying date and time options for orders, with a limit of no more than a weekly order.
	Address Verification	AddVer	Verifies delivery addresses during order placement, ensuring accuracy and providing support for bearers to navigate to the correct location.

### ***3.2 Structural Design – Class Diagram / Structure Chart not both>***



### 3.2.1 Design Notes

<Explain the diagram in terms of relationships between classes/modules, any assumptions and constraints which impacted the diagram design or why certain relationships were specified in the diagram>

**\*\*Explanation of Diagram:\*\***

#### 1. **\*\*Main Menu Module:\*\***

- **\*\*Explore and Search:\*\*** The Main Menu Module allows users to explore local restaurant menus. The 'MainMenu' class is responsible for displaying menu items and enabling search functionality ('searchMenuItems'). The 'MenuItem' class provides specific information about each menu item, including description, price, ingredients, and nutritional information. The 'MenuSearch' class facilitates the search capability.

- **\*\*Menu Item Specifics:\*\*** The 'MenuItem' class encapsulates information about each menu item, fulfilling the requirement for specific details.

#### 2. **\*\*User Module:\*\***

- **\*\*Registration and Authentication:\*\*** The 'User' class manages user registration and authentication ('register', 'authenticate'). The system ensures secure authentication procedures with password and fingerprint options.

- **\*\*Manage User Profiles:\*\*** Users can create, update, and delete profiles through the `User` class (`createProfile`, `updateProfile`, `deleteProfile`). Personal information, delivery addresses, and preferred payment methods are included in the `Profile` class.
- **\*\*Order History:\*\*** The `OrderHistory` class allows users to view their order history (`viewOrderHistory`) and repeat past orders in the system (`repeatOrder`).

### 3. **\*\*Order Module:\*\***

- **\*\*Place Order:\*\*** The `Order` class handles placing orders (`placeOrder`) by adding products to the shopping basket and providing an order summary before purchase.
- **\*\*Delivery Options:\*\*** Users can select delivery or pickup options, including choosing a delivery time slot. The `Order` class manages these delivery possibilities.
- **\*\*Payment:\*\*** Payment transactions are handled securely through the `PaymentSystem` class, which accepts various payment methods (`processTransaction`).
- **\*\*Order Confirmation:\*\*** The `Order` class manages order confirmation, sending details such as expected delivery time and order number via email and push notifications.

### 4. **\*\*Tracking Module:\*\***

- **\*\*Real-time Order Tracking:\*\*** The `OrderTracking` class enables users to view the real-time status of their orders (`trackOrder`).
- **\*\*Anticipated Time of Delivery (ETA):\*\*** The anticipated delivery time is dynamically updated based on the delivery person's current position, managed by the `OrderTracking` class (`getEstimatedDeliveryTime`).
- **\*\*Delivery Person Information:** The `DeliveryPerson` class provides information such as name and contact details for users to examine (`getDeliveryPersonInfo`).

### 5. **\*\*Payment Module:**

- **Secure Transactions:** Secure payment transactions are ensured through the `PaymentSystem` class (`processTransaction`), which also handles sending payment receipts to users via email and in the app (`sendReceipt`).
- **\*\*Multiple Payment Methods:\*\*** The `PaymentSystem` class accommodates various payment methods, such as credit/debit cards, digital wallets, and other local methods.
- **\*\*Promotions and Discounts:\*\*** The `Promotion` class validates and applies promo codes and discounts during the checkout process (`validatePromotion`, `applyPromotion`).

### Assumptions and Constraints:

- The diagram assumes a centralized system architecture where classes within each module interact directly.
- Security measures for authentication and payment transactions are crucial, and the diagram reflects the segregation of these concerns into the `User` and `PaymentSystem` classes.
- The system assumes the availability of email and push notification services for order and payment confirmations.
- The diagram implies real-time tracking relies on the delivery person's device providing location updates.

### Design Choices:

- The design utilizes a modular approach with distinct classes for each module, promoting encapsulation and maintainability.
- Relationships are specified to reflect the flow of functionalities, such as placing an order, tracking it in real-time, and making secure payments.

- The use of class hierarchies, like the `MenuItem` and `Profile` classes, allows for flexibility in managing specific details for different entities.

This design aims to provide a clear and organized structure for the Big 6 Food Delivery Application, ensuring that each module handles its specific functionalities effectively.