AP Assignment.docx

by Mohamed Althaff Abdul Raheem

Submission date: 20-Sep-2022 10:30AM (UTC+0100)
Submission ID: 186110109
File name: 111163_Mohamed_Althaff_Abdul_Raheem_AP_Assignment_1545261_308993001.docx (4.18M)
Word count: 2706
Character count: 15863

Table of Contents

Task A 2

Application Requirement Specification 2

- Functional Requirement 2
- Functional Requirement 2

Task B 3

Diagrams 3

- Use Case Diagram 3
- Class Diagram 4
- Sequence Diagram 5

Task C 6

System design patterns 6

- Types of design patterns 6
- Singleton Design Pattern 7
- Factory Design Pattern 9
- Abstract Factory Design Pattern 12

Task D 13

Tasks E 14

- Test Plans – Related to Go Cheeta 14
- Test Cases – Related to Go Cheeta 15

Tasks F 24

User Manual 24

- Login page 24
- Customer Home page 25
- Driver Home page 25
- Customer Profile page 26
- Customer Booking Status page 26
- Customer Booking page 27
- Driver Profile page 27
- Driver Booking page 28

Tasks F 29

- Git repository creation 29
- Git repository Link 29
- Git commits creation/updates 29

References 30

Task A

Application Requirement Specification

Functional Requirement

- Signup / register (Customer)
- Login (Admin, Customer, Driver)
- View available vehicles
- Search for a respective category of vehicle
- Book a vehicle
- Track location
- Booking confirmation details
- View estimated fare
- Ride status
- View ride history
- Feedback sharing
- Logout

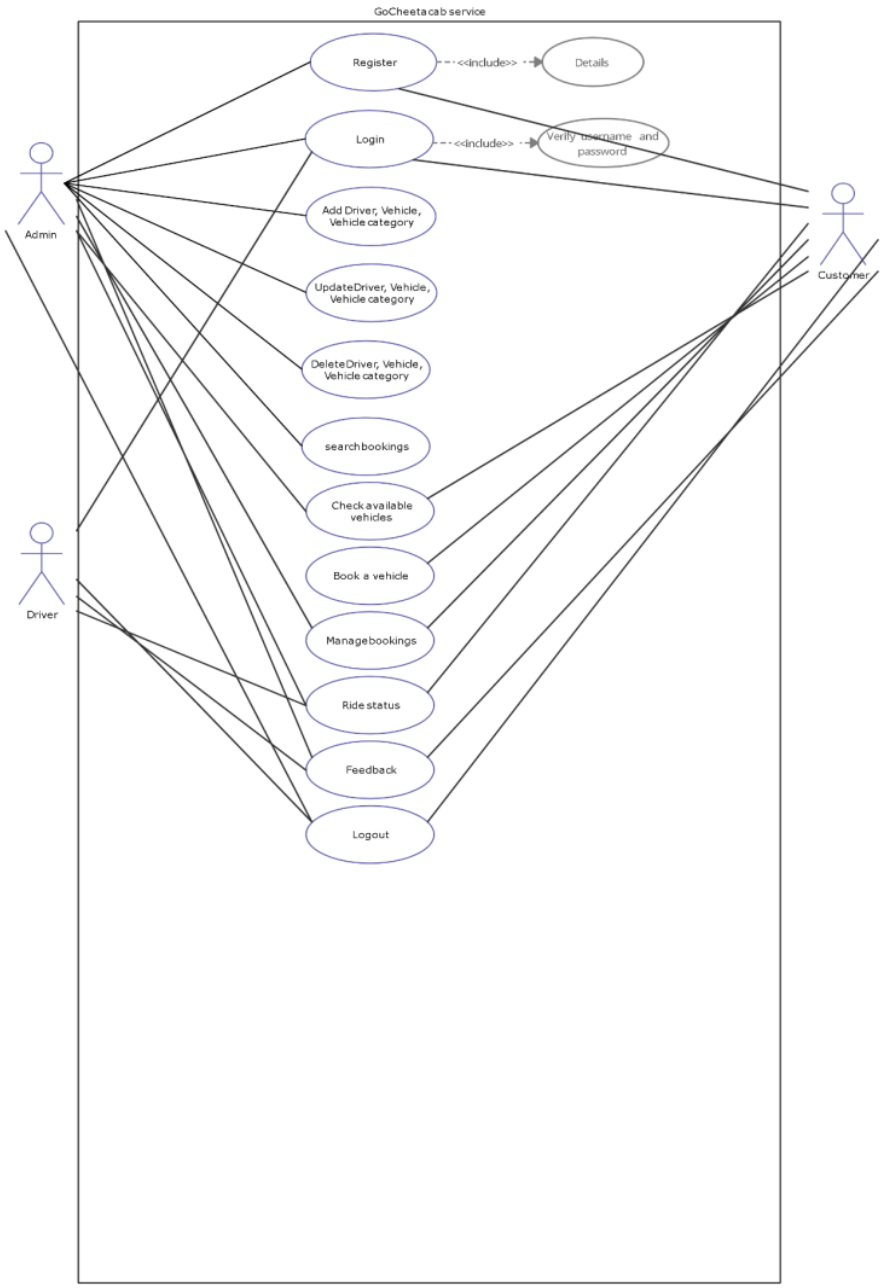
Functional Requirement

- Usability - These requirements are concentrated on how the user interface appears and how users interact with it. Color's, screen size, button size, etc., should all be described.
- Performance - Establish the system's operating speed requirements.
- Reliability - These specifications define system availability, such as whether the system must be available constantly.
- Security - Establish the security standards from a cybersecurity standpoint.

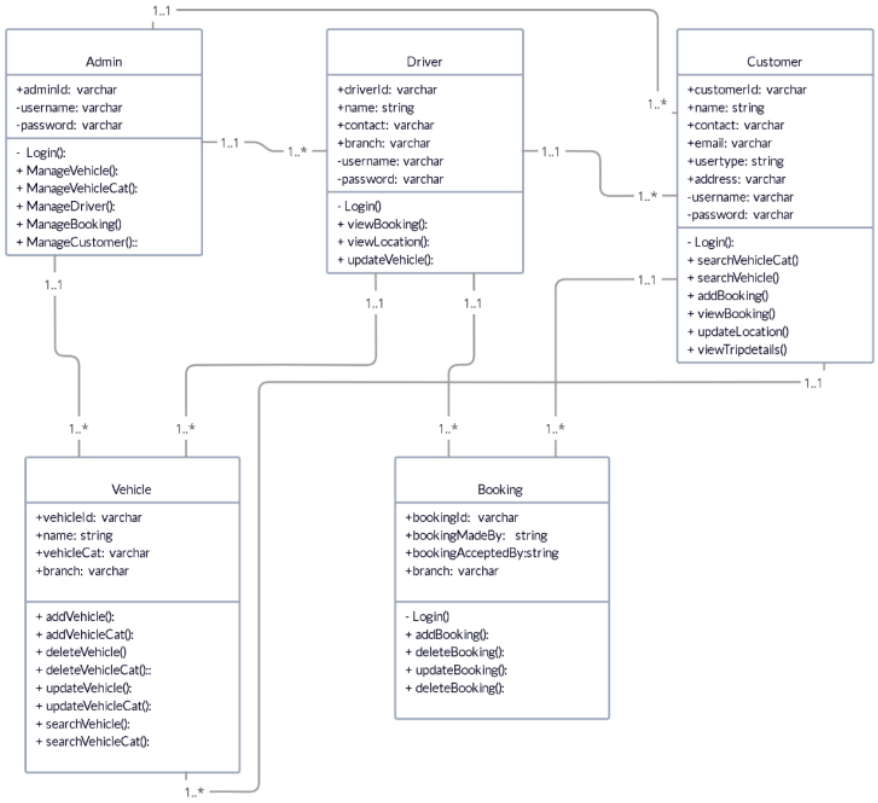
Task B

Diagrams

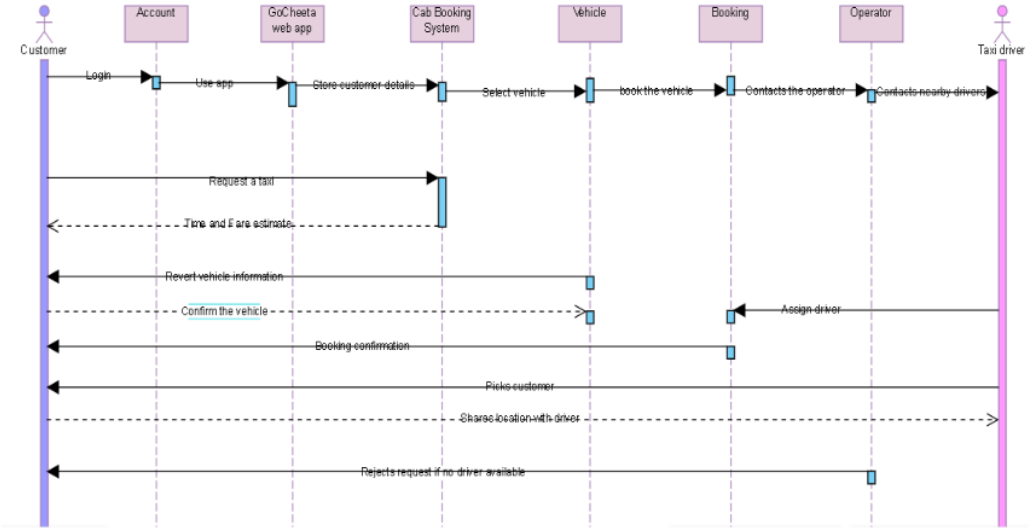
Use Case Diagram



Class Diagram



Sequence Diagram



Task C

System design patterns

Experienced object-oriented programmers apply a number of guiding concepts, with design patterns being one sample. In object-oriented systems, a design pattern is a structured approach of naming, motivating, and explaining a generic design solution. It outlines the issue, the fix, when to use it, and any negative effects. (geeksforgeeks.org, 2018) Additionally, it offers examples and implementation hints. by providing tested, proven development paradigms Design patterns could accelerate the process of development.

Considering focus on problems that the implementation phase, which is essential for effective application development, may not appear until later. Reusing design patterns reduces little errors that might cause major issues and increases the code's clarity for developers and designers who are knowledgeable about the patterns. Individuals routinely have limited experience with different digital design techniques on particular types of challenges. It would be challenging to use these methods for a wider variety of issues. Design patterns offer broad fixes that are given without the need for specifics pertaining to a particular problem. (sourcemaking.com, 2022)

Types of design patterns

1. Creational design patterns -The main idea of these process models is class initialization. Templates for constructing classes and methods for creating objects can be used to further decompose this design. While object-creation designs effectively use delegation, class-creation patterns effectively leverage inheritance all across the setup phase. (Rahman, 2019)
2. Structural design patterns - The creation of classes and objects is the main focus among these process models. Interfaces are built via inherited using structural class development strategies. To provide additional functionality, object composition methods utilizing appropriate infrastructure patterns are discussed.

3. Relational design patterns: Those schema center on how items in a class communicate with one another. The behavioural styles that emphasize inter-object communication the most. (Rahman, 2019)

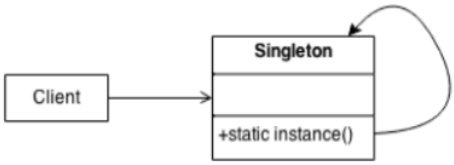
Singleton Design Pattern

The following code is one of the simplest process models for Java. Given that it is arguably the greatest methods for producing an item, this particular style falls under the category of a creational pattern. According to the Singleton design, each separate, distinct class instance contains a common resource. (2022; TutorialsPoint.com) Under this case, actual data linked to storage and resources is arbitrated access. It's no longer necessary to share the connection because a class method provides access to this instance This may be used by every other object that has accessibility to the Singleton's class header. (freecodecamp.org, 2018)

This prototype outlines the composition of a class with a single occurrence limit. A Singleton isolates a special resource and makes it accessible throughout the whole program. A persistent storage, a piece of hardware, a network service, or any other entity that can be treated as a singular object or service might be the resource.

A gadget running an iOS application is one illustration of Cocoa touch. Each iPhone or iPad will only have one screen and a battery for an app that is now running. Since it offers a single channel for interacting with the fundamental features, UI Device is a Singleton class in this context. A race issue or deadlock may result from this kind of mismatch if the unique resource has a writable configuration. Singletons serve as a control by assuring ordered access to the shared resource since they are unique.

Structure -



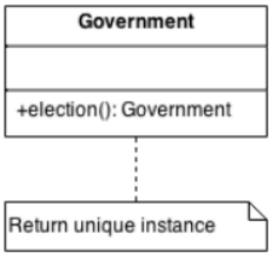
Set access and "initialization on first use" within the control of the class of the single instance. It is a private static attribute with just one instance. A static public method, the accessor function.

Disadvantage in Singleton design pattern:

- Unit testing is hampered by Singletons: If the object and the methods connected to it are too deeply linked to be tested without creating a class specifically for the Singleton, a Singleton may make it difficult to write testable code.
- Because singletons are freely available across the code base, they may be overused, which results in hidden relationships. Additionally, it becomes challenging to follow as its reference is not entirely clear when passed to multiple methods. (freecodecamp.org, 2018)

Example:

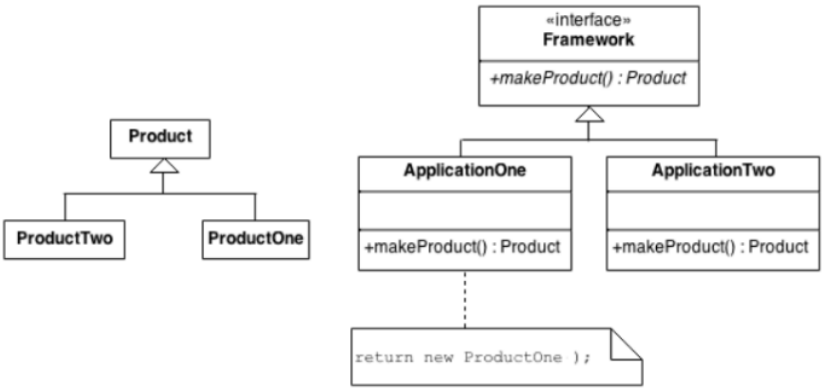
A single incident of a class is guaranteed by the Singleton pattern, which also offers a singular point of access to that instance globally. It is called after the singleton set, which is a set with just single element by definition. The presidency of the United States is a Singleton position. The United States Constitution outlines the procedures for electing presidents, sets term limits, and establishes the transition process. There can only ever be one functioning president at a time due to this. The term "The President of the United States" is a universal point of access that identifies the person in the office, irrespective of the personal identity of the current president.



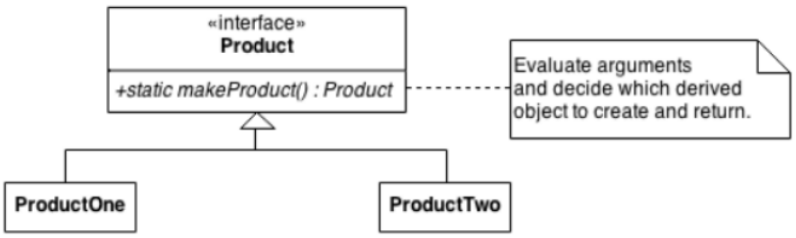
Factory Design Pattern

This process model is one of Java's most well-liked design patterns. This type of design pattern is a creational pattern since All of this delivers one of the greatest methods for producing an object. Create an interface for creating objects (such a Java interface or an abstract class), and following the factory design pattern, let the subtypes determine whichever class to create. An interface's factory function enables a class to delay development in favor of one or more concrete subclasses. These frameworks explain the construction of an object, hence they come under the category of creational design patterns. (geeksforgeeks.org, 2022)

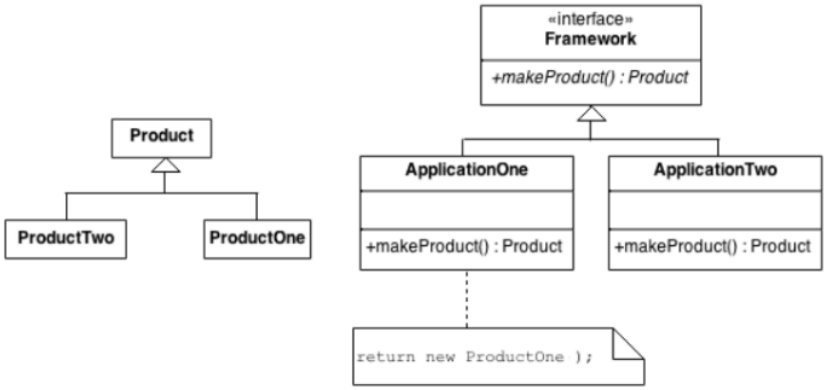
Structure - The Gang of Four's (below) discussion on Factory Method implementation has a lot in common with Abstract Factory. The presentation in this chapter concentrates on the strategy that has since gained popularity as a result. (sourcemaking.com, 2022)



One concept of a factory method that is becoming more and more common is: a class static method delivering an item of the class's "class" type. It can, instead, provide an instance of a subclass instead of a constructor as the real object. As opposed to a builder, it is possible to reuse an existing object rather than creating a new one. Factory methods, as opposed to constructors, might be given various, more evocative names. (Example – Colors. make_RGB_color(float red, float green, float blue) and Color .make_HSB_color(float hue, float saturation, float brightness)

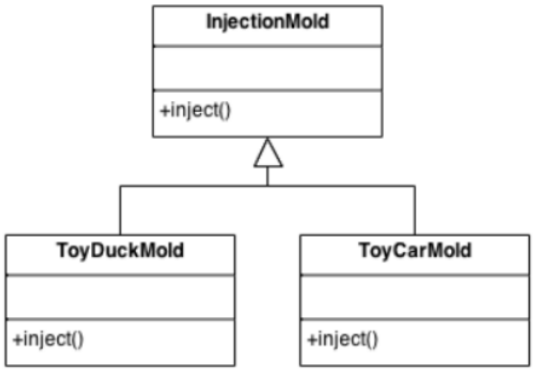


The operational specifics of derived classes are completely independent of the client. We can now create things that are polymorphic.



Example:

Even though Factory Method offers an interface for creating objects, it gives subtypes the freedom to decide which classes to initialize to a constructor. Pushes for injection molding to display this pattern. Manufacturers utilize plastic molding extract, which is treated before the plastic is poured into the appropriate molds, to make plastic toys. The sort of toy is determined by the mold (car, action figure, etc.).



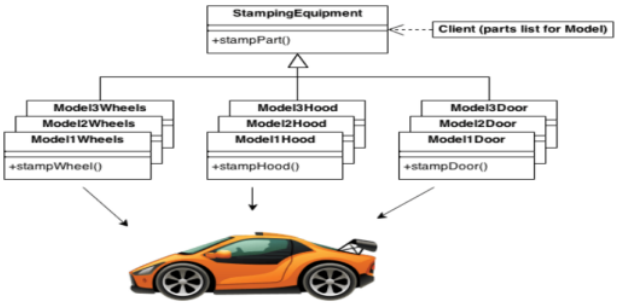
Abstract Factory Design Pattern

The abstract factory design pattern is among the creational patterns. Abstract factory pattern is a degree of refinement above factory pattern and is almost equivalent to factory pattern. Abstract factory patterns are centered around a super-factory that produces further workplaces. The use of the abstract factory pattern has given us a framework that makes it possible to create things that follow a common theme. Therefore, at runtime, any necessary concrete factory that can manufacture the specified type of things is connected with the abstract factory. (geeksforgeeks.org, 2021)

Problem - System dependency must be contained in an application for it to be accessible. These "platforms" might be a database, operating system, windowing system, etc. Too frequently, this encapsulation is not planned ahead of time, and a large number of #ifdef case statements with choices for all systems that are now supported begins to spread throughout the code like rabbits. (sourcemaking.com, 2022)

Structure - Each Factory Method for an item is specified by the Abstract Factory. The newer provider and the relevant, platform-specific product classes are both included in each Factory Method. A class that derives from the Factory is then used to model each "platform". (sourcemaking.com, 2022)

Example - The Abstract Factory's function is to offer a creation interface for groups of related items without defining specific classes. The sheet metal striking machinery used in the production of Japanese cars has this pattern. An Abstract Factory uses the stamping machinery to make vehicle body pieces. For various automobile models, the same equipment is used to stamp right- and left-handed doors, right- and left-handed front fenders, right- and left-handed hoods, etc. The concrete classes generated by the equipment may be altered in 3 minutes by using rollers to replace the stamping dies.



Task D

Task / Source code is attached to the document

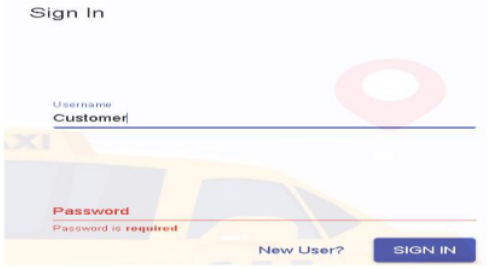
Tasks E

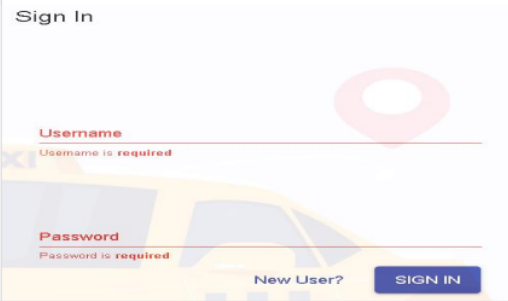
Test Plans – Related to Go Cheeta

Test case ID	Test case name	Description	Expected results
TC 01	Check login	Check login with correct user name and password.	Login successfully
TC 02	Check login	Check login with correct user name and wrong password.	Login failed
TC 03	Check login	Check login with wrong user name.	Login failed
TC 04	Check Logout	Check logout in Driver UI	Logout successfully
TC 05	Check Logout	Check logout in Customer UI	Logout successfully
TC 06	Registration	Input Details Customer / Driver	Successfully Registered
TC 07	Registration	Empty Fields	Fill the details
TC 08	Dashboard	View Dashboard Customer	Display Dashboard
TC 09	Search	Search Vehicle / Booking	Display Details
TC 10	View	View Booking Information	Display Booking Details
TC 11	View	View Profile Information	Display Details
TC 12	Booking	Booking Vehicles when it is Available	Booked Successfully
TC 13	Search	Search Vehicle / Booking	Display Details
TC 14	View	View Booking Information	Display Booking Details
TC 15	Add	Add Vehicles Manually	Added Successfully
TC 16	Edit	Edit Vehicles Availability / Details	Edited Successfully
TC 17	Delete	Delete Vehicle Information	Deleted Successfully

Test Cases – Related to Go Cheeta

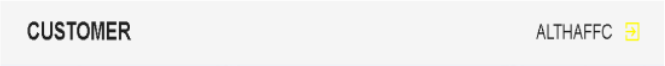
Test case ID	TC 01
Test case name	Login Check
Description	login with correct user name and password.
Test data	Username: AlthaffD or AlthaffC, Password: driver/customer
Expected result	Login successfully
Actual result	expected result
Screen shot	
Conclusion	Login function works without disruption.

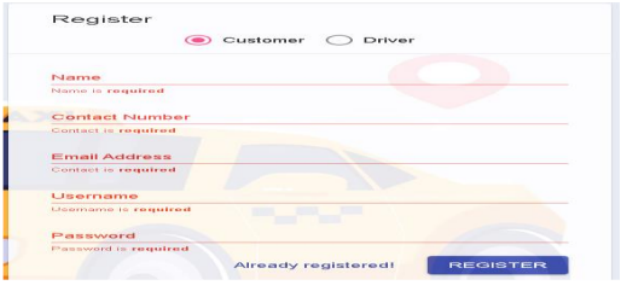
Test case ID	TC 02
Test case name	Login Check
Description	login with correct user name and wrong password.
Test data	Username: AlthaffD or AlthaffC, Password: 12
Expected result	Login failed
Actual result	expected result
Screen shot	
Conclusion	Login failed.

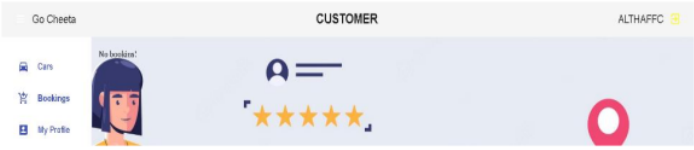
Test case ID	TC 03
Test case name	Check login
Description	Check login with wrong user name
Test data	Username: Abc
Expected result	Login failed
Actual result	expected result
Screen shot	
Conclusion	Login failed

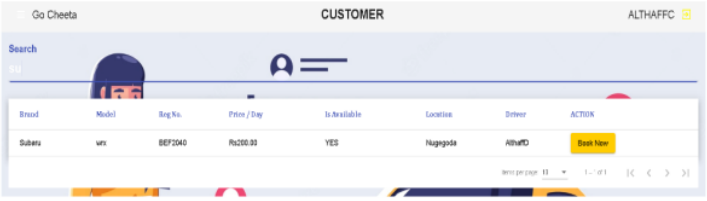
Test case ID	TC 04
Test case name	Check Logout
Description	Check logout in Driver UI
Test data	Press Logout icon next to user name
Expected result	Logout successfully
Actual result	expected result
Screen shot	
Conclusion	Logout successfully

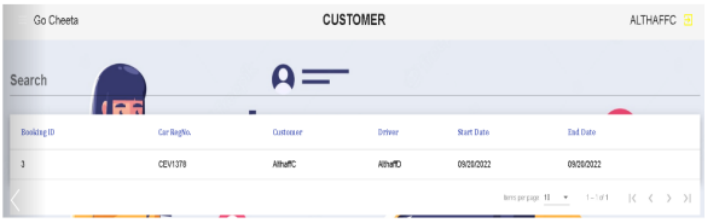
Test case ID	TC 05
Test case name	Check Logout
Description	Check logout in Customer UI
Test data	Press Logout icon next to username
Expected result	Logout successfully
Actual result	expected result
Screen shot	
Conclusion	Logout successfully

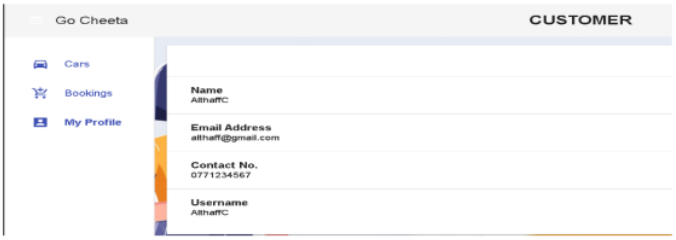
Test case ID	TC 06
Test case name	Registration
Description	Input Details Customer / Driver
Test data	Input Proper details
Expected result	Successfully Registered
Actual result	expected result
Screen shot	
Conclusion	Successfully Registered

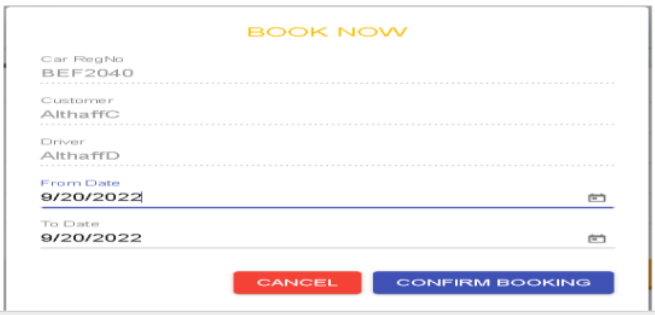
Test case ID	TC 07
Test case name	Registration
Description	Empty Fields
Test data	Fill the details Correctly without missing a detail
Expected result	Fill the details
Actual result	expected result
Screen shot	
Conclusion	Display Fill the details Error Message

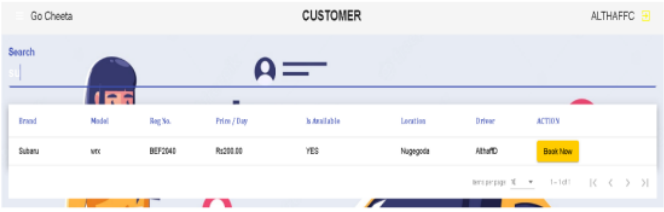
Test case ID	TC 08
Test case name	Dashboard
Description	View Dashboard Customer
Test data	Display Functions in Dashboard
Expected result	Display Dashboard
Actual result	expected result
Screen shot	
Conclusion	Display Dashboard Details

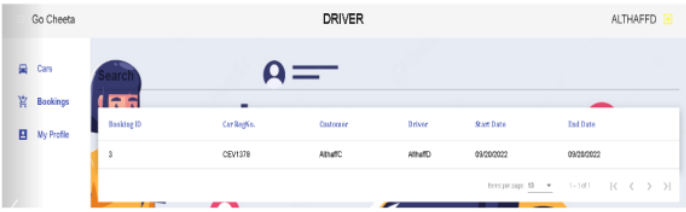
Test case ID	TC 09
Test case name	Search
Description	Search Vehicle / Booking
Test data	Search anything in a short manner
Expected result	Display Details
Actual result	expected result
Screen shot	
Conclusion	Display Searched Details

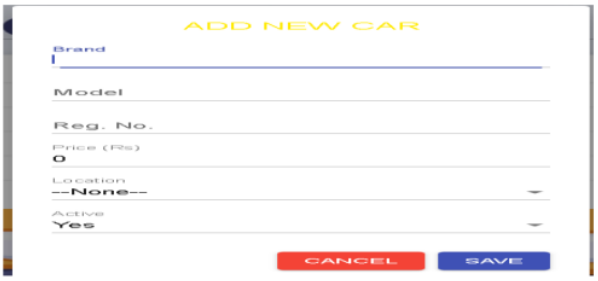
Test case ID	TC 11
Test case name	View
Description	View Booking Information
Test data	View the Date of Booked Details
Expected result	Display Booking Details
Actual result	expected result
Screen shot	
Conclusion	Display Booking Details

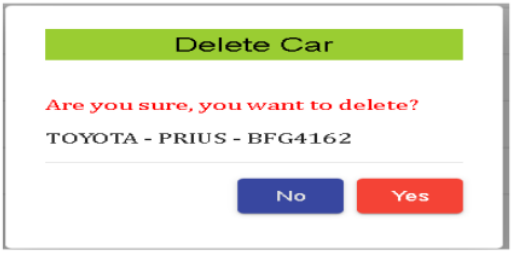
Test case ID	TC 12
Test case name	View
Description	View Profile Information
Test data	View the information's provided
Expected result	Display Details
Actual result	expected result
Screen shot	
Conclusion	Display Profile Information's

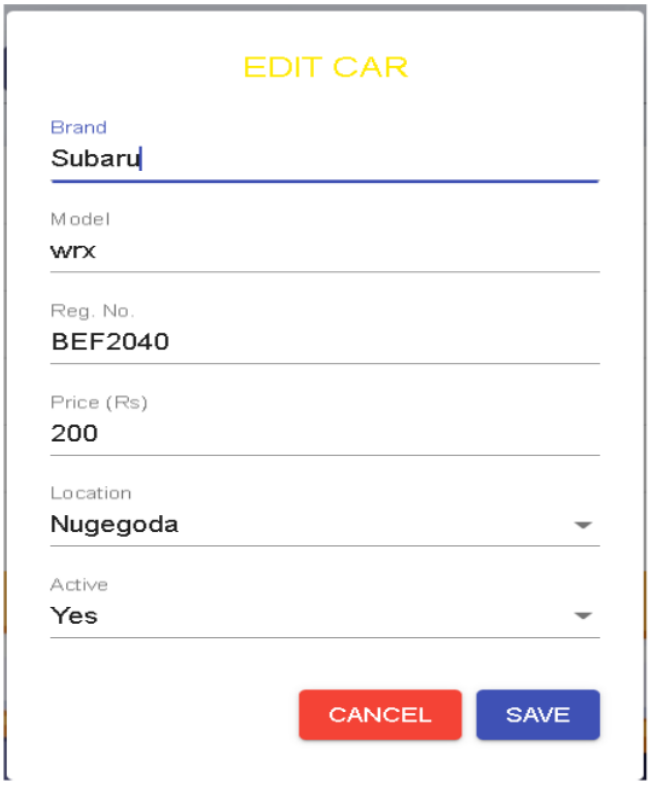
Test case ID	TC 13
Test case name	Booking
Description	Booking Vehicles when it is Available
Test data	Booking the available vehicles
Expected result	Booked Successfully
Actual result	expected result
Screen shot	
Conclusion	Booked Successfully

Test case ID	TC 15
Test case name	Search
Description	Search Vehicle Model / Booking Details
Test data	Search anything in a short manner
Expected result	Display Details
Actual result	expected result
Screen shot	
Conclusion	Display Searched Details

Test case ID	TC 17
Test case name	View
Description	View Booking Information
Test data	View Booking Information
Expected result	Display Booking Details
Actual result	expected result
Screen shot	
Conclusion	Display Booking Details

Test case ID	TC 19
Test case name	Add
Description	Add Vehicles Manually
Test data	
Expected result	Added Successfully
Actual result	expected result
Screen shot	
Conclusion	Added Vehicles data Successfully

Test case ID	TC 21
Test case name	Delete
Description	Delete Vehicle Information
Test data	Delete Vehicle Information
Expected result	Deleted Successfully
Actual result	expected result
Screen shot	
Conclusion	Deleted Successfully

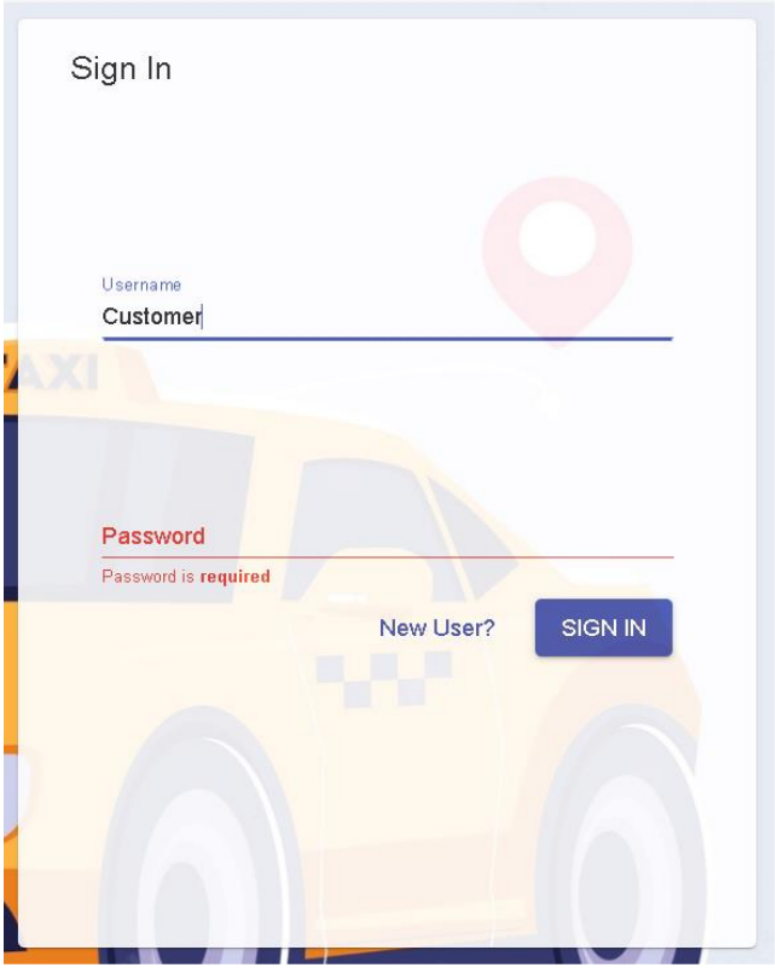
Test case ID	TC 20
Test case name	Edit
Description	Edit Vehicles Availability / Details
Test data	Edited Successfully
Expected result	Updated Successfully
Actual result	expected result
Screen shot	
Conclusion	Updated Successfully

Tasks F

User Manual

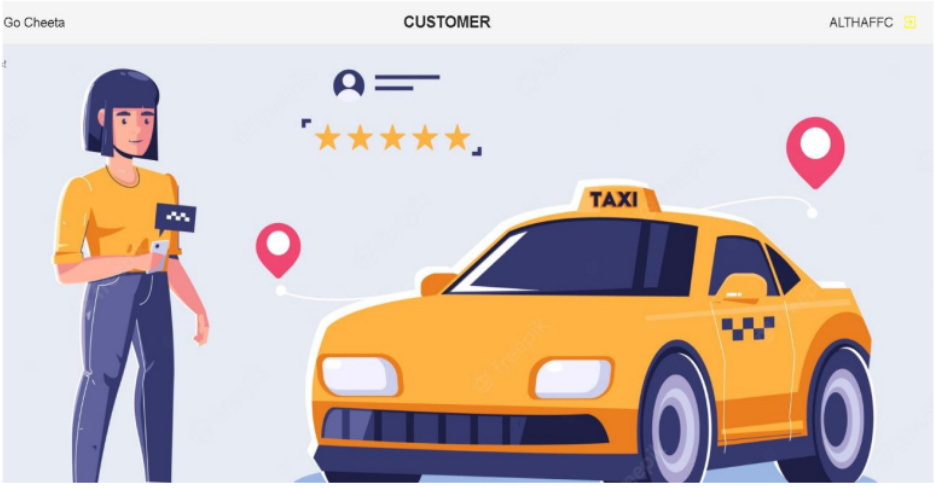
Login page

User is required to enter the correct username and password to log in/access the application.



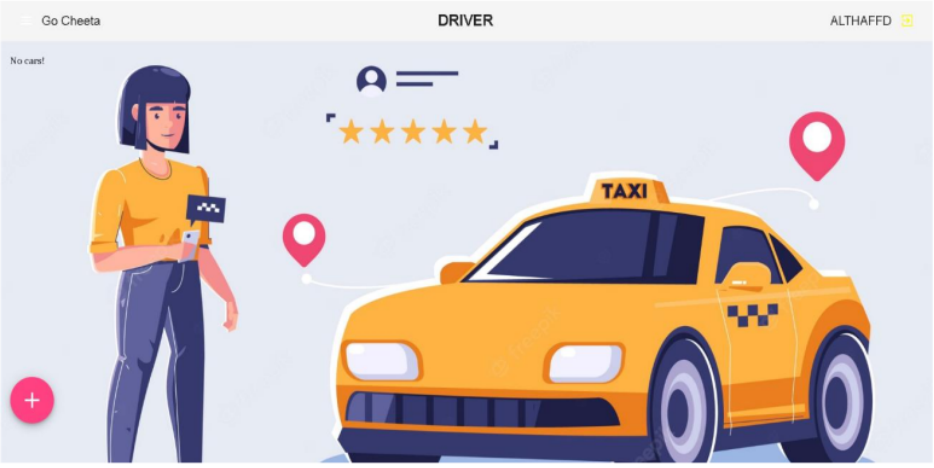
Customer Home page

User can select provided options and manage their profile accordingly.



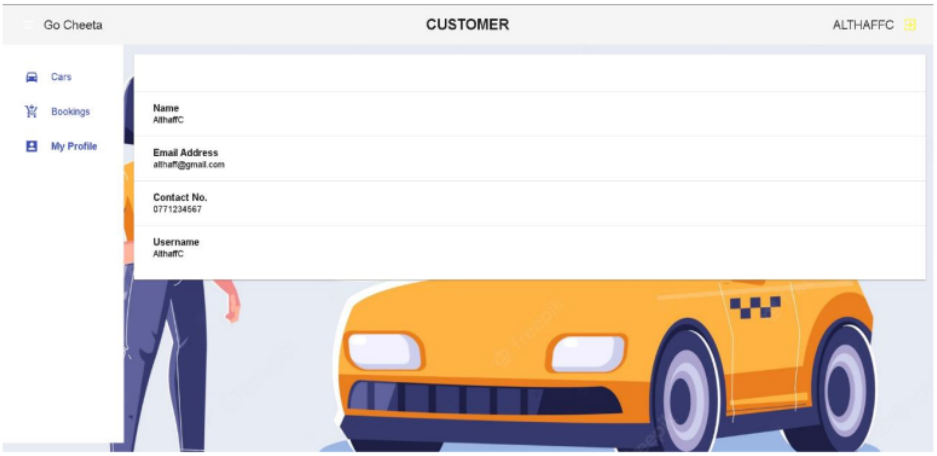
Driver Home page

User can update their vehicle/profile according



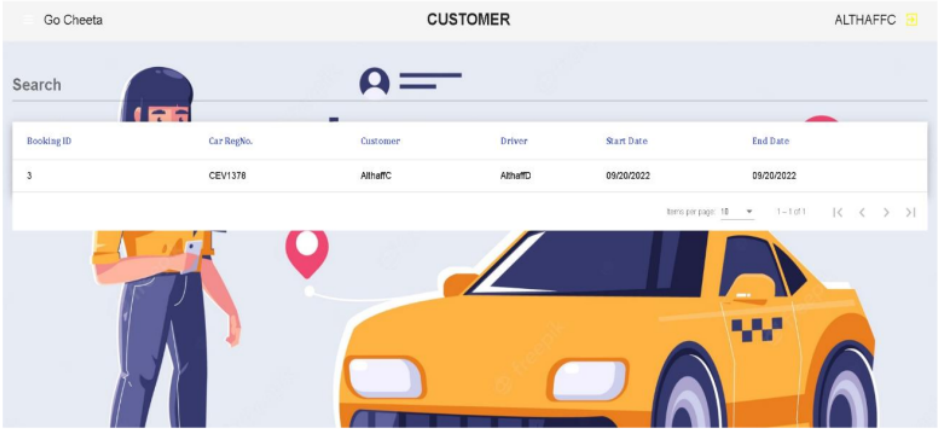
Customer Profile page

User can view their Profile details but will not be able to edit them due to security reasons.



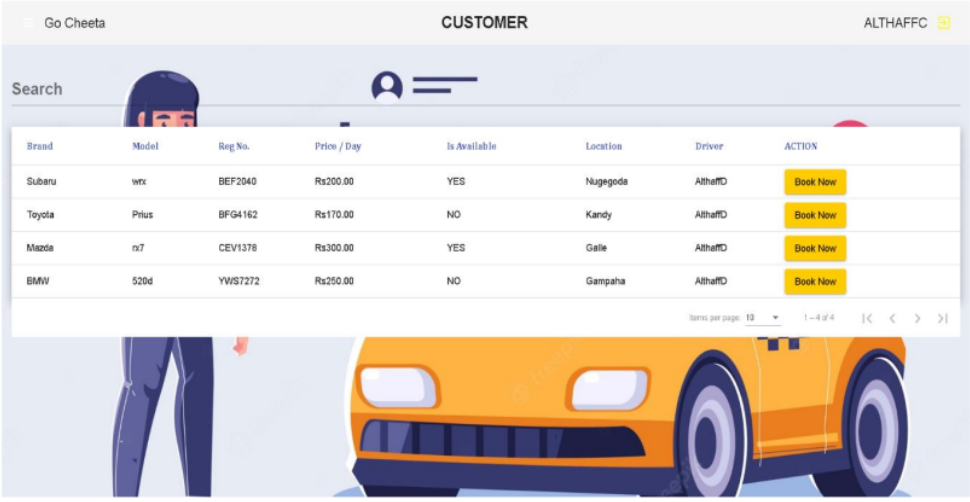
Customer Booking Status page

User can view their Booking status and booking details



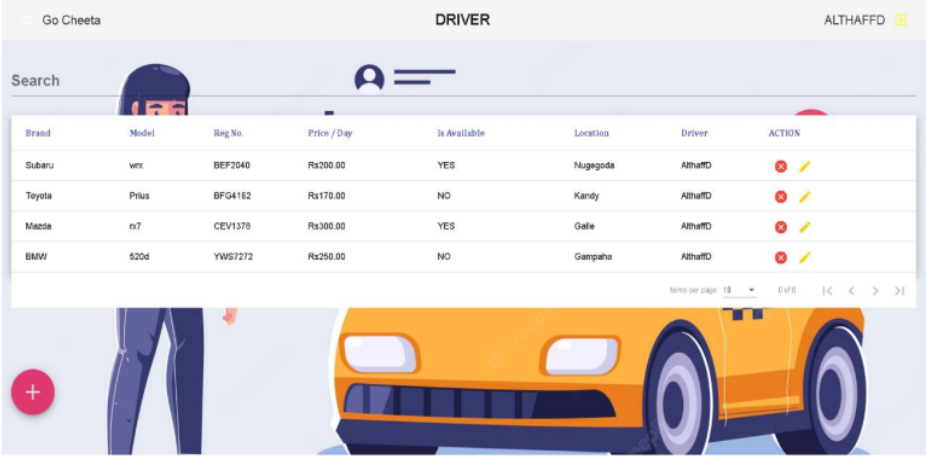
Customer Booking page

User can book their desired vehicle by clicking on them.



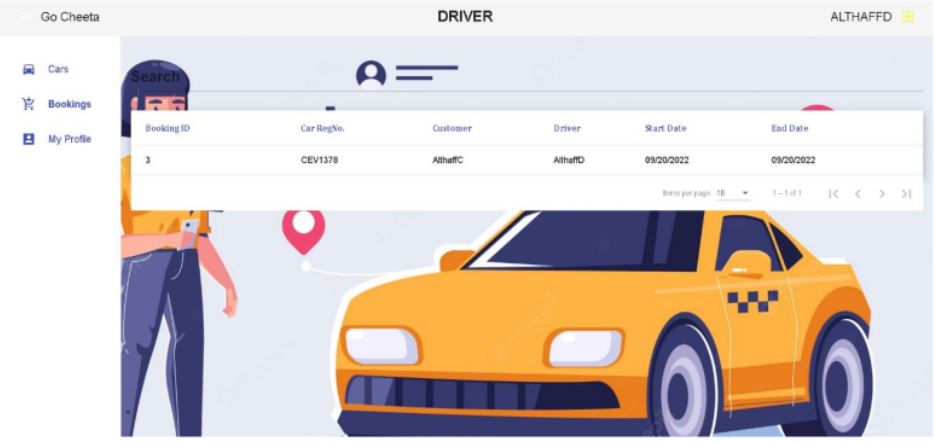
Driver Profile page

User can view their profile details



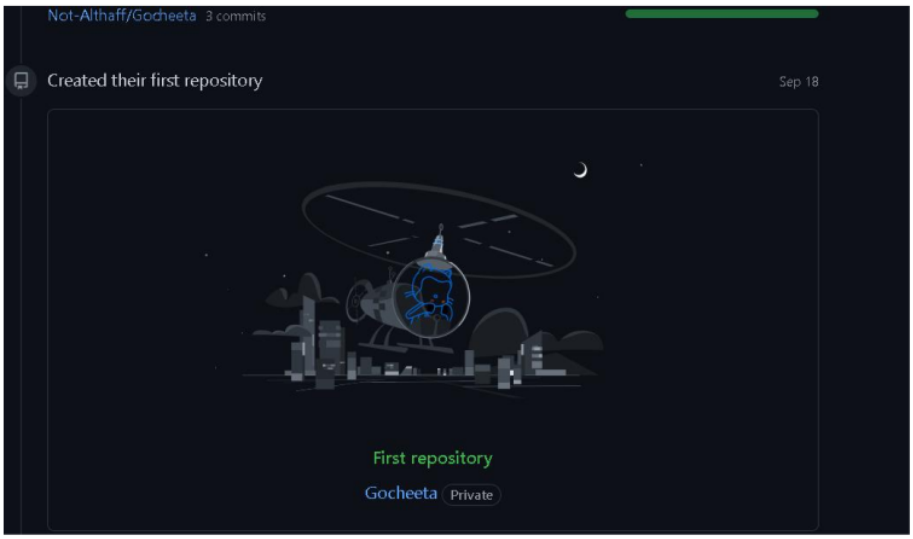
Driver Booking page

User can view how many bookings they have for their vehicle



Tasks F

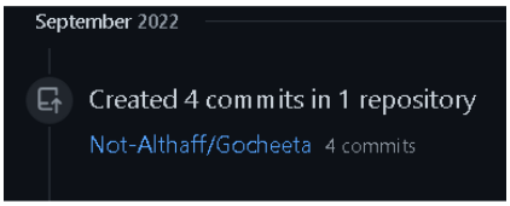
Git repository creation



Git repository Link

https://github.com/Not-Althaff/Gocheeta.git

Git commits creation/updates



References

freecodecamp.org, 2018. *Let's examine the pros and cons of the Singleton design pattern*. [Online]
Available at: https://www.freecodecamp.org/news/singleton-design-pattern-pros-and-cons-e10f98e23d63/#:~:text=Singletons%20hinder%20unit%20testing%3A%20A,class%20dedicated%20to%20the%20Singleton.
[Accessed 15 September 2022].

geeksforgeeks.org, 2018. *Software Design Patterns*. [Online]
Available at: https://www.geeksforgeeks.org/software-design-patterns/
[Accessed 15 September 2022].

geeksforgeeks.org, 2021. *Abstract Factory Pattern*. [Online]
Available at: https://www.geeksforgeeks.org/abstract-factory-pattern/
[Accessed 16 September 2022].

geeksforgeeks.org, 2022. *Factory method design pattern in Java*. [Online]
Available at: https://www.geeksforgeeks.org/factory-method-design-pattern-in-java/
[Accessed 16 September 2022].

Rahman, S., 2019. *The 3 Types of Design Patterns All Developers Should Know*. [Online]
Available at: https://www.freecodecamp.org/news/the-basic-design-patterns-all-developers-need-to-know/
[Accessed 15 September 2022].

sourcemaking.com, 2022. *Abstract Factory Design Pattern*. [Online]
Available at: https://sourcemaking.com/design_patterns/abstract_factory
[Accessed 16 September 2022].

sourcemaking.com, 2022. *Design Patterns*. [Online]
Available at: https://sourcemaking.com/design_patterns
[Accessed 15 September 2022].

sourcemaking.com, 2022. *Factory Method Design Pattern*. [Online]
Available at: https://sourcemaking.com/design_patterns/factory_method
[Accessed 16 September 2022].

tutorialspoint.com, 2022. *Design Pattern - Singleton Pattern*. [Online]
Available at: https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm
[Accessed 15 September 2022].

AP Assignment.docx

ORIGINALITY REPORT

21%	10%	4%	20%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Wales Institute, Cardiff Student Paper	18%
2	people.dsv.su.se Internet Source	1%
3	Submitted to Oxford Brookes University Student Paper	1%
4	Submitted to Higher Education Commission Pakistan Student Paper	1%
5	medium.com Internet Source	1%
6	sourcemaking.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		