SENG 275

SOFTWARE TESTING

DR. NAVNEET KAUR POPLI

DEPT. OF ELECTRICAL AND COMPUTER ENGINEERING

University of Victoria

TEST CASES FROM USE CASES

University of Victoria

Use cases

- Use cases are used in UML to capture the usage requirements for a system.
- Use case diagrams are extremely useful in communicating with stakeholders.
- The **textual representation of use cases** is however useful for both **developers and stakeholders** because they describe what the actual user and system would do in response to a specific stimuli.
- In this lecture we will try to understand how use cases can be used to drive the testing process allowing a tester to develop test cases directly from user's usage of the system and essentially from the documentation of the use cases.

University of Victoria

Objectives of the lecture

- Explain the purpose of the use case diagram.
- Given a **use case**, construct an **activity diagram** showing the flow through the use case.
- Given an activity diagram, construct a **set of test cases** to fully exercise a use case.

Thug Bug vehicle security system-1980s

University of Victoria

Features

- Requirement to enter a **4 digit key code before starting the car**. Without entering the proper keycode the car will not start.
- Additionally the system featured a **vehicular hood lock.**
- The hood lock prevented the thieves from opening the hood unless they entered the same 4 digit keycode.
- This prevented the theft of devices like batteries or other things from underneath the hood.

University of Victoria

Keypad (standard numeric)

University of Victoria

Mechanism

- To start the car the user places the key in the accessory mode powering the security device and enters the 4 digit code onto the keypad for e.g. 1234
- If the keycode was incorrect the security system prevented the car from starting.
- It was novel and ingenious but also pretty simple.
- The system also required the entry of keycode to open the hood. To open the hood, the user turned the car again into the accessory position and enter the same 4 digit code 1234 and **hold the 0 button for 5 sec.** after which the hood would unlock.
- This system prevented a common problem in a simple manner.

University of Victoria

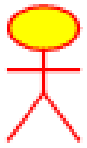
Use Case Modeling

- Use case modelling is a useful tool for providing a graphical representation of the software system's **requirements**.
- Diagram(s) provide overview of **actors** and **use cases**
- A **use case is a unit of functionality** in the system
- Because use case models are **simple** both in **concept and appearance**, it is relatively easy to discuss the correctness of a use case model with a **non-technical person** (such as a customer).
- Use cases are technically **classes** in object-oriented paradigm.

University of Victoria

Actors

- An actor is any **entity, human or otherwise**, that is **external** to the system being designed. Two symbols are available in the UML specification:



Name

An actor can be represented by a stick figure.

«actor»
Name

Alternatively, an actor may be represented by a class with the «actor» stereotype.

- The **stakeholders** should certainly be on the initial list of actors.
- Other potential actors include all the **computer systems** with which the system will interact, and any other **hardware devices** (including hardware such as **printers, bar-code readers,** and so on).

University of Victoria

Use case diagrams

-- represents what happens when actor interacts with a system.
-- captures functional aspect of the system.

Actor

Use Case

Relationship between actors and use case and/or between the use cases.

-- Actors appear outside the rectangle.

--Use cases within rectangle providing functionality.

--Relationship association is a solid line between actor & use cases.

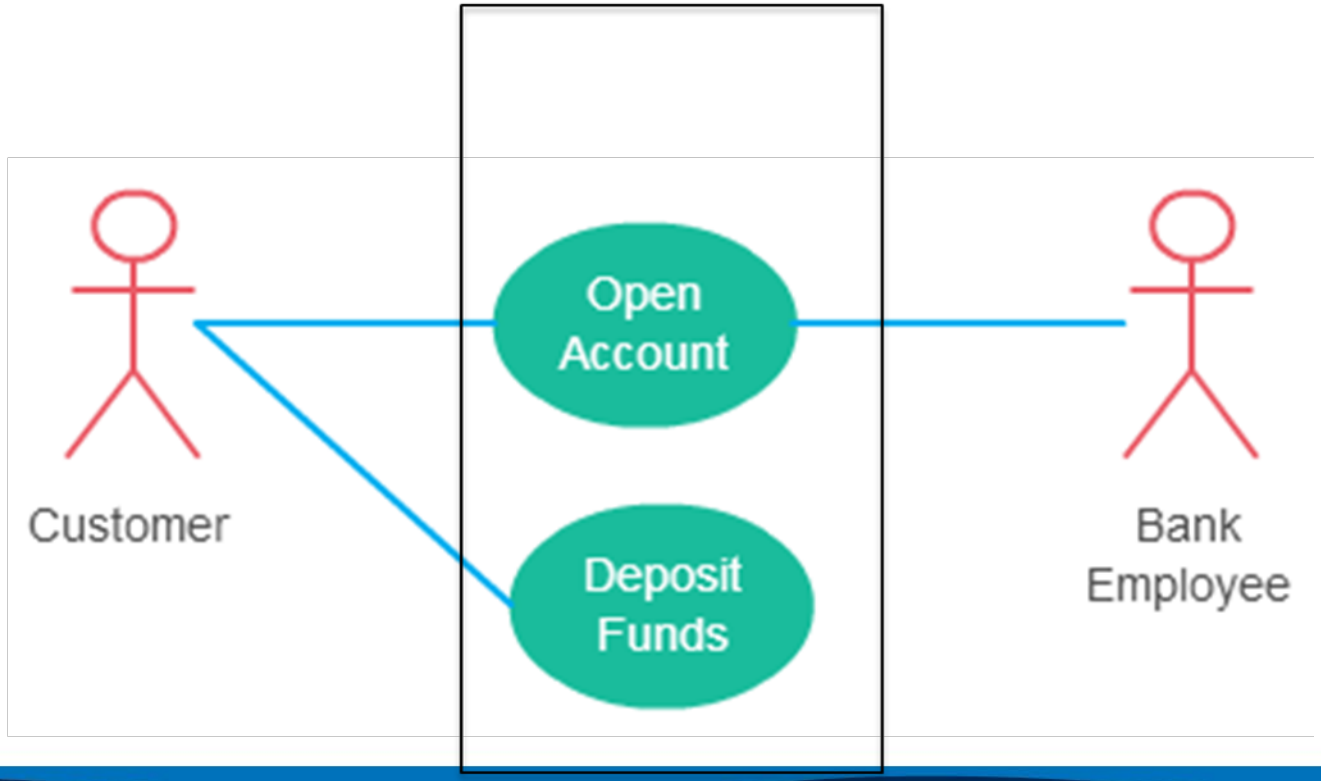
Primary and secondary actors

- **Primary actors: initiate the use** of the system. Left
- **Secondary actors:** are **reactionary** / supportive. They help to complete the use case. Right

Simple Use case diagram for a bank

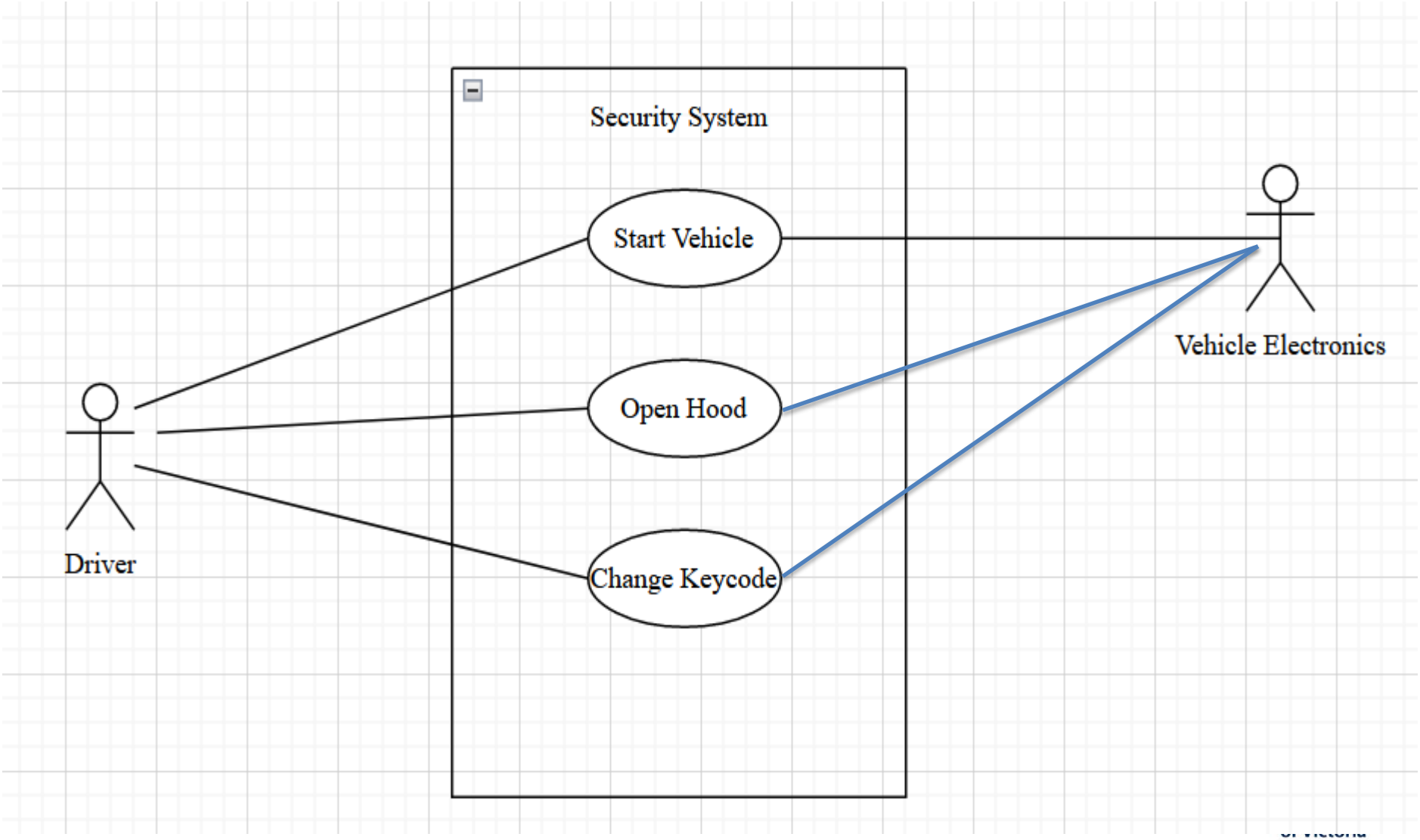
- A customer in a bank can open his account and deposit funds in the account.
- A bank employee can help them in opening the account.

Use case diagram for a bank



University of Victoria

Use case diagram of Thug Bug Security System



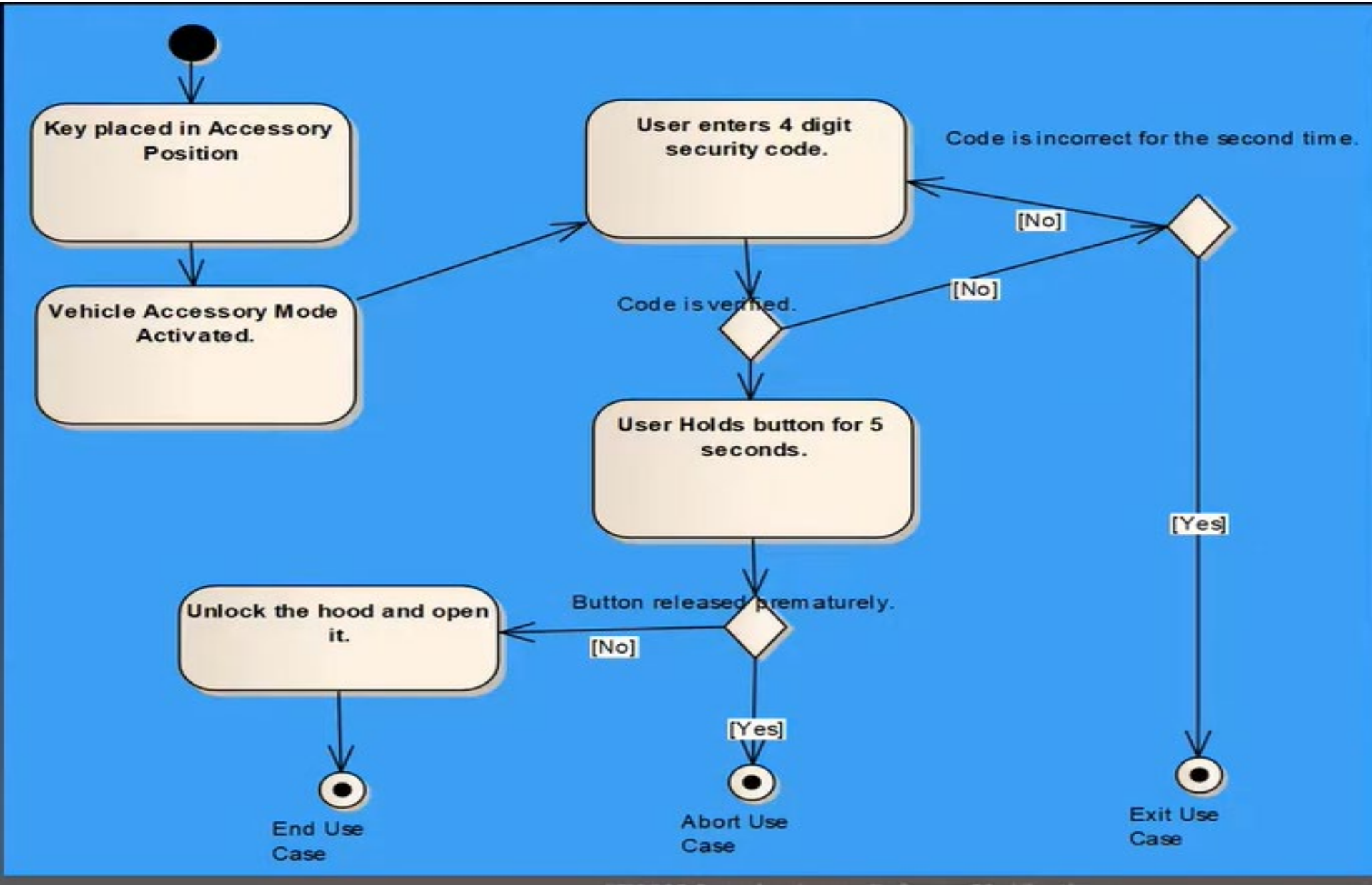
Use Case specification for the Open hood Use Case

- **Use Case Name**
  - Open Hood
- **Summary**
  - In this use case, the driver of the vehicle successfully opens the hood of the vehicle.
- **Actor**
  - Driver
- **Precondition**
  - Key is inside the ignition.
- **Postcondition**
  - The hood of the vehicle is opened.
- **Flow** :
  1. The user places the key in the accessory position. The car activates vehicle electronics.
  2. The user enters a four digit security code on the keypad. System verifies code is correct.
  3. User presses and holds the 0 button for 5 seconds to unlock the hood.
- **Alternatives** :
  - 2.a Code entered is incorrect. Hood remains locked. User may try reentering security code one more time.
  - 2.b Code entered is incorrect for the second time. Hood remains locked, and user must cycle ignition to off before trying again.
  - 3a User does not hold button for 5 seconds. Use case is aborted.

Activity diagram-show the information in a visual format

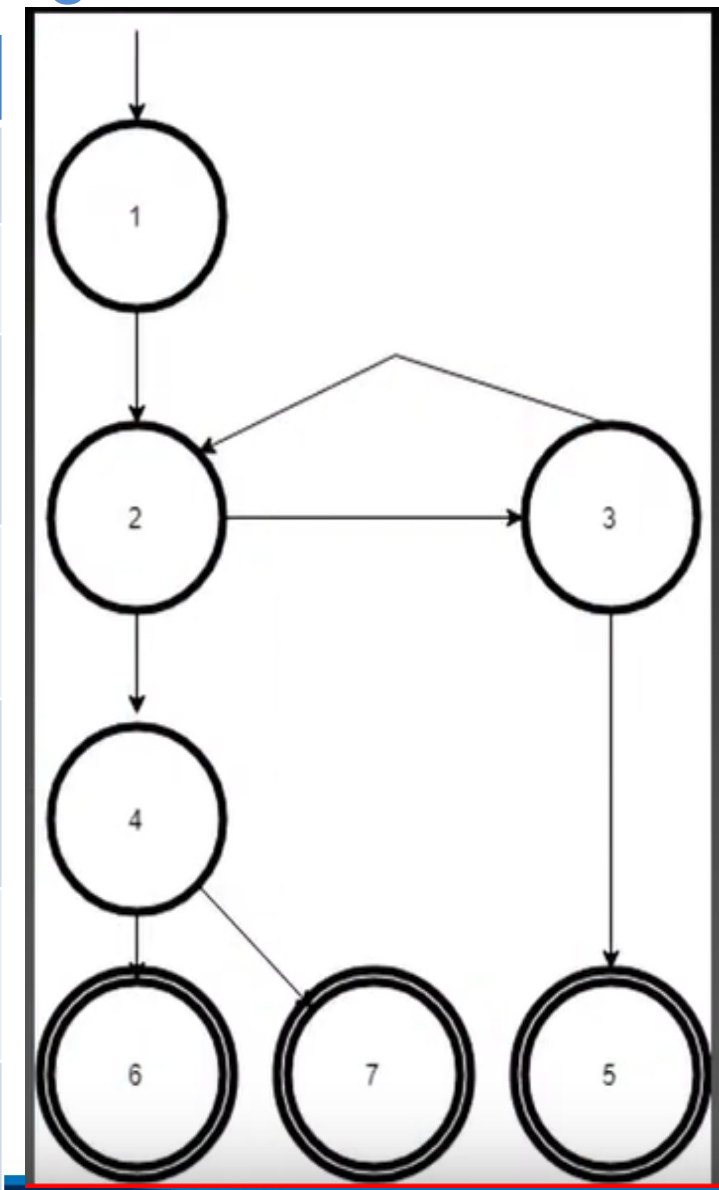
- Activity diagram is also a UML diagram.
- It is basically a flowchart through a use case.
- It has 2 kinds of nodes:
  - Action states are derived from use case descriptions
  - Sequential branches are derived from alternative flows.
- Flow between steps become edges.

Activity diagram for ‘Open Hood’ use case



Graph generated from the activity diagram

Node	Action
1	User turns key to accessory position.
2	User enters the keycode.
3	User enters the code incorrectly the first time and thus is given a chance to enter the code again.
4	User enters the correct keycode of 1234.
5	User enters the keycode incorrectly the second time too and thus the hood remains locked.
6	User presses and holds the 0 button for 5 seconds and thus the hood is unlocked.
7	User presses and holds the 0 button for less than 5 seconds and thus hood remains locked.

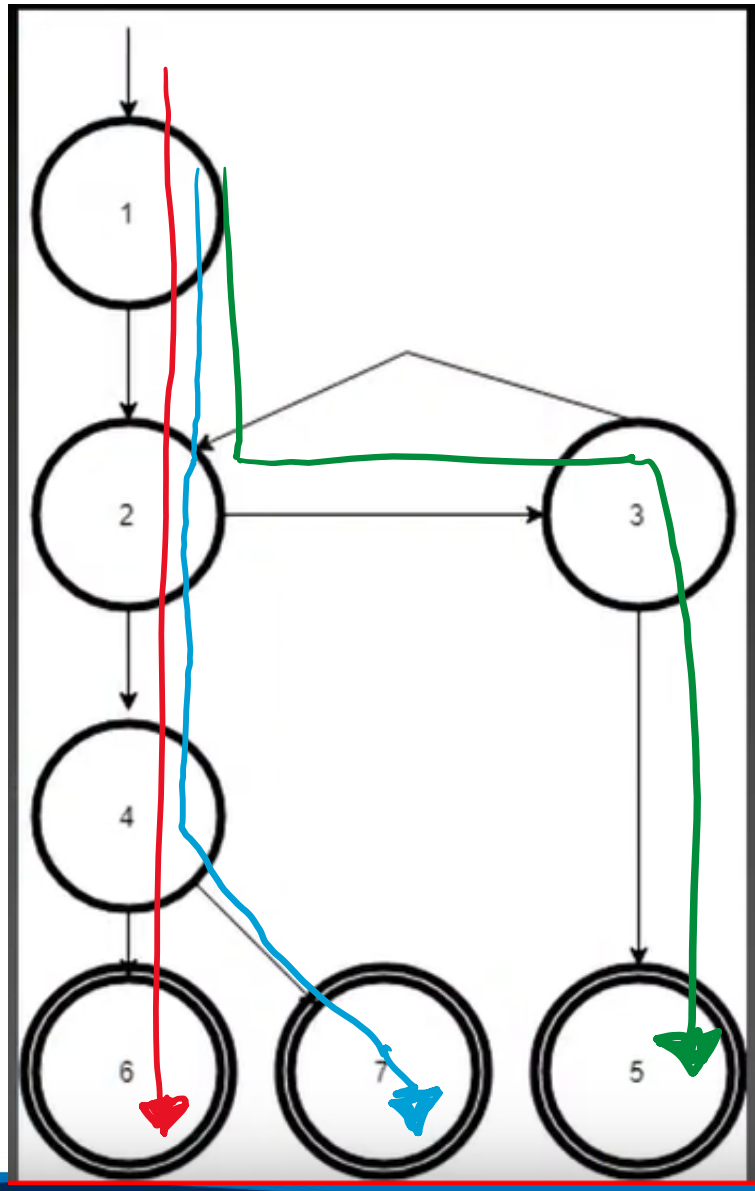


Node Coverage

Node	Action
1	User turns key to accessory position.
2	User enters the keycode.
3	User enters the code incorrectly the first time and thus is given a chance to enter the code again.
4	User enters the correct keycode of 1234.
5	User enters the keycode incorrectly the second time too and thus the hood remains locked.
6	User presses and holds the 0 button for 5 seconds and thus the hood is unlocked.
7	User presses and holds the 0 button for less than 5 seconds and thus hood remains locked.

1-2-4-6
1-2-4-7
1-2-3-5

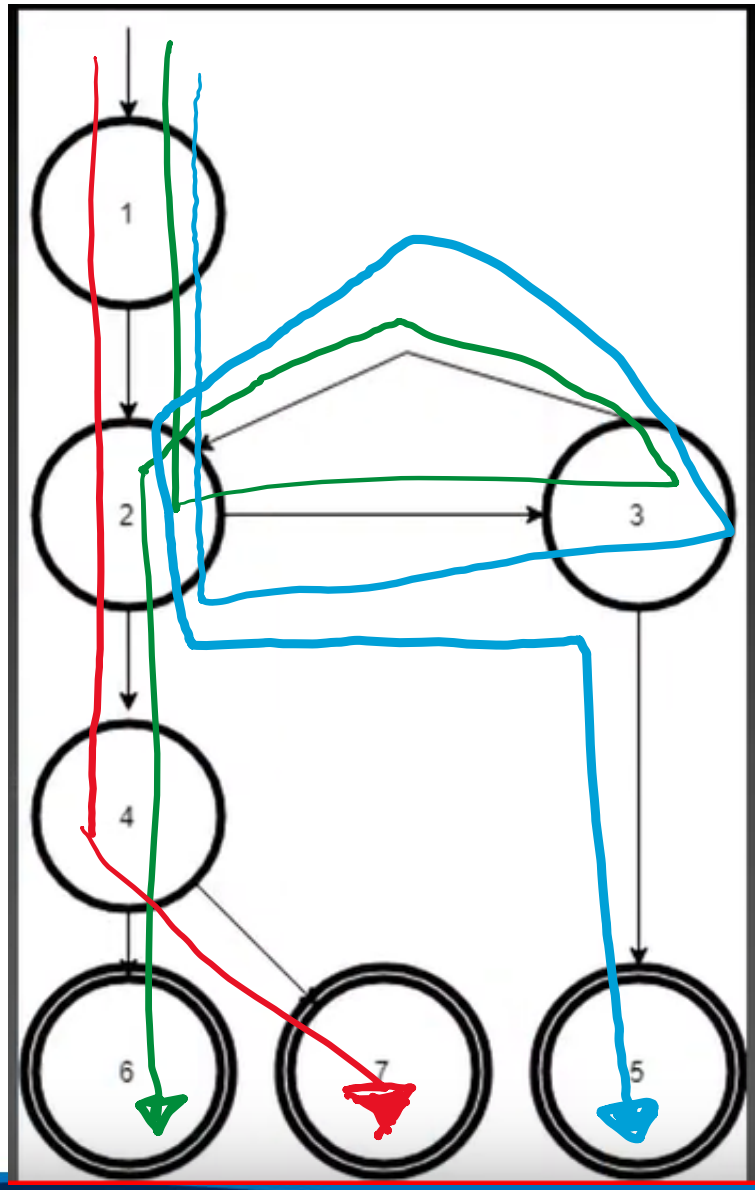
} Complete node coverage



Path	1-2-4-6	1-2-4-7	1-2-3-5
Initial Conditions	Key is in ignition. Keycode is set to 1234	Key is in ignition. Keycode is set to 1234	Key is in ignition. Keycode is set to 1234
Test Steps	1. User turns key to accessory position. 2. User enters the correct keycode of 1234 3. User presses and holds the 0 button for 5 seconds.	1. User turns key to accessory position. 2. User enters the correct keycode of 1234 3. User presses and holds the 0 button for less than 5 seconds.	1. User turns key to accessory position. 2. User enters the incorrect keycode of 1124. 3. The condition of re entering keycode after this step is not exercised.
Expected Result	Hood is unlocked.	Hood remains locked.	??? Node coverage does not exercise the entire use case.

Edge coverage

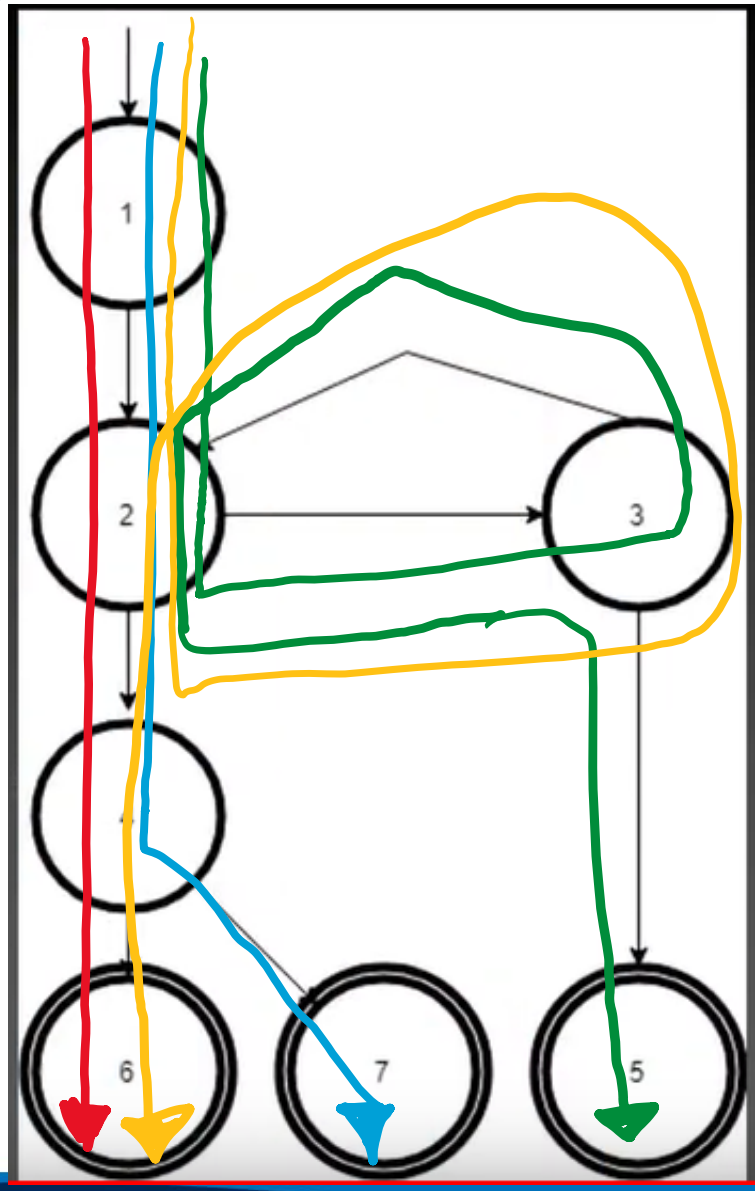
Node	Action
1	User turns key to accessory position.
2	User enters the keycode.
3	User enters the code incorrectly the first time and thus is given a chance to enter the code again.
4	User enters the correct keycode of 1234.
5	User enters the keycode incorrectly the second time too and thus the hood remains locked.
6	User presses and holds the 0 button for 5 seconds and thus the hood is unlocked.
7	User presses and holds the 0 button for less than 5 seconds and thus hood remains locked.



Edge pair (Specified Path) coverage

Node	Action
1	User turns key to accessory position.
2	User enters the keycode.
3	User enters the code incorrectly the first time and thus is given a chance to enter the code again.
4	User enters the correct keycode of 1234.
5	User enters the keycode incorrectly the second time too and thus the hood remains locked.
6	User presses and holds the 0 button for 5 seconds and thus the hood is unlocked.
7	User presses and holds the 0 button for less than 5 seconds and thus hood remains locked.

1-2-4-6
1-2-4-7
1-2-3-2-5
1-2-3-2-4-6



Test cases to fully exercise the use case

Path	1 – 2 – 4 – 6	1 – 2 – 4 - 7	1 – 2 – 3 – 2 - 4 - 6	1 – 2 – 3 – 2 – 3 - 5
Initial Conditions	Key is in the ignition Keycode is set to 1234.	Key is in the ignition. Keycode is set to 1234.	Key is in the ignition Keycode is set to 1234.	Key is in the ignition Keycode is set to 1234.
Test Steps	1. User turns key to accessory position. 2. User enters the correct keycode of 1234. 3. User presses and holds the 0 button for 5 seconds.	1. User turns key to accessory position. 2. User enters the correct keycode of 1234. 3. User presses and holds the 0 button for less than 5 seconds.	1. User turns key to accessory position. 2. User enters the incorrect keycode of 4321. 3. User reenters correct keycode of 1234 4. User presses and holds the 0 button for 5 seconds.	1. User turns key to accessory position. 2. User enters the incorrect keycode of 4321. 3. User reenters incorrect keycode of 4321
Expected Result	Hood is unlocked.	Hood remains locked	Hood is unlocked	Hood remains locked

Assignment- Doggles Daycamp

- The purpose of the Kennel Management System (KMS) is to provide a management system for Doggles Daycamp. This system must safely and securely store information about customers, pets, and kennel bookings. The system must publicly show the availability of vacancies in a particular Doggles Daycamp kennel in a calendar view, and the services the Doggles Daycamp location offers which can include walking and bathing. Information about a kennel booking must be available to the customer who booked, and should be anonymized for anyone else, unless they are an employee with authorization to view this information. The system must incorporate a livestream feature for registered customers to see their pets in real time while they are away.
- The system must also provide users the ability to make a financial donation to Doggles Daycamp.

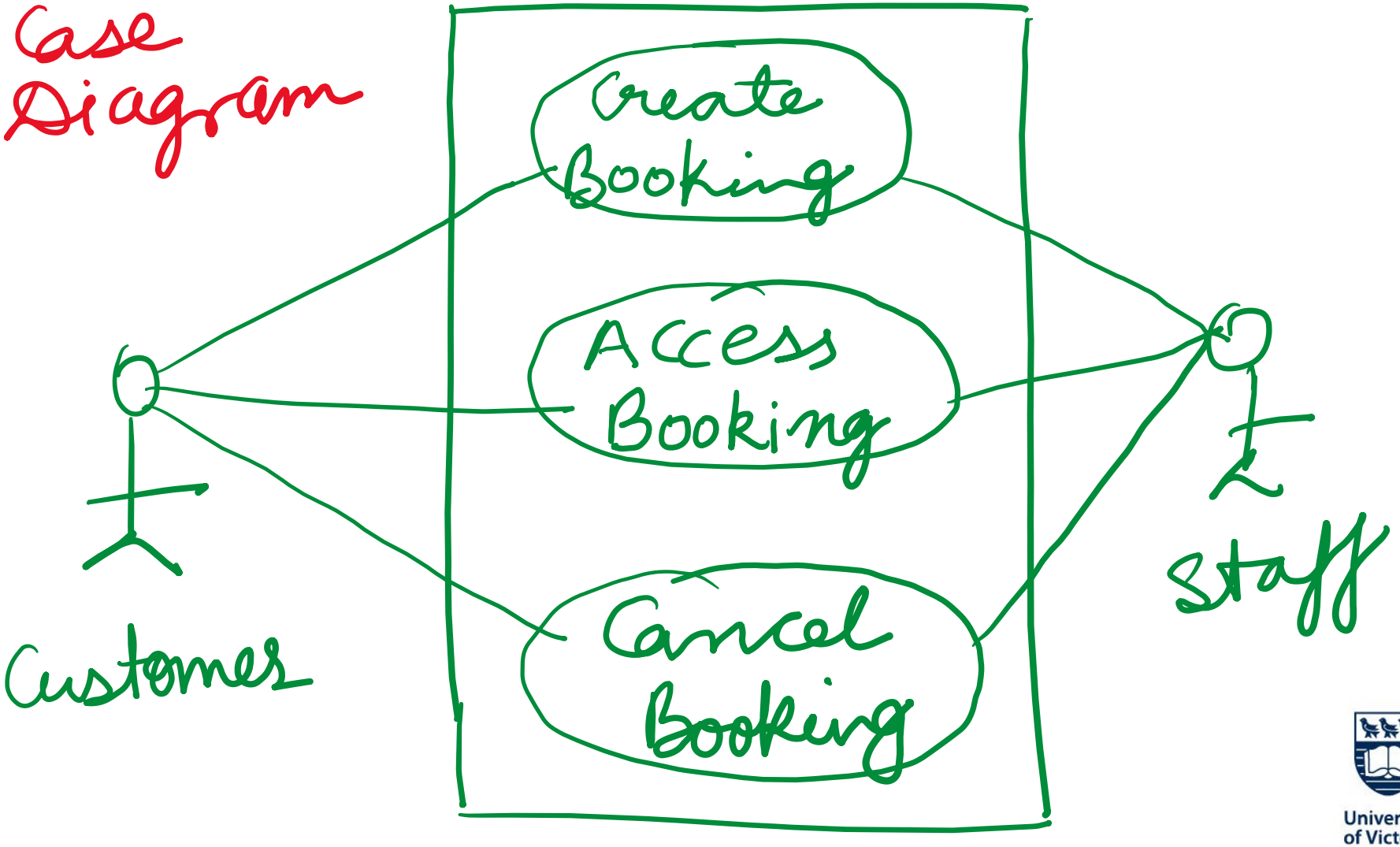
Doggles Daycamps Kennel Booking feature

- **Description and Priority:** The booking feature is where registered users or employees processing a walk-in booking will input their information and pay for a booking at a Doggles Daycamps location.
- It is accessed through the calendar in the kennel information interface.
- Inputted customer information is used to keep the system database up to date.
- Since booking represents the main income source of Doggles Daycamp, it is a high priority.

University of Victoria

Kennel Booking System

Use Case Diagram



University of Victoria

UC-4-1: Create a Booking

ID: UC-4-1

Description: A user creates a kennel booking.

Actors: User, System Database, Payment Service

Preconditions: N/A

Main Flow:

1. User inputs full name, contact phone number, contact email, pet name, pet breed, pet color, pet allergies, pet medications, into the respective text inputs on the booking form.
2. User selects a date from a calendar widget to indicate when they will pick up their pet.
3. User selects all optional add-ons they wish to add to their booking, using checkboxes present for dog walking and bathing.
4. The user clicks “Submit” at the bottom of the form.
5. User clicks “Confirm” on a confirmation prompt showing their inputted booking details.
6. User selects a payment service to pay for their booking and is redirected to the selected payment service. Successful payment redirects the user to the User Bookings page and prompts a system message “Booking Confirmed”.
7. Booking is sent to System Database and associated with the user’s database entry.
8. A confirmation email is sent to the user’s email address, containing receipt and booking details.

Postconditions:

- The user has a new kennel booking associated with their user entry in the system database.
- The user receives a confirmation email with the details of their booking.

University of Victoria

UC-4-1: Create a Booking

Alternative Flow(s):

5a: If the user does not fill out one or more fields

1. System message “One or more required fields is empty” appears.
2. Continue from 4.

7a: If payment processing fails

1. System message “Payment could not be processed” appears.
2. Prompt user to select “Try Again” or “Cancel”.
3. If “Try Again” is selected, continue from 7. Otherwise, quit the booking process.

- Create an activity graph for the feature.
- Create test cases which will efficiently test the system.