**SOFTENG 325 Assignment 2: Architecture for Big Brother Driver (BBD)**

**Architecture Description**

**BBD-V Module Structure**

**Diagram 1:** “Uses” Structure for BBD-V (arrows represent “uses” relationships and the texts represent modules)

Get Location/Speed

Process Location/Speed

Location/Speed Repository

Send data to BBD-Ops

Manager

Encode Information

User Profile

Road Layout

**Diagram 1** is the “Uses” structure for the BBD-V architecture which consists of many modules.

The “**Get Location/Speed**” module is for acquiring the information from the GPS technology implemented. The information contains current location and the speed of the driver with the mobile device.

The “**Process Location/Speed**” module is for processing the information gathered by the first module into usable format assuming the raw information is not suitable (i.e. not efficient or not readable) by the device. The processed information will be stored in the “Location/Speed Repository” module.

The “**Location/Speed Repository**” is for storing information gathered. The speed and location of the driver will be saved based on the current time.

The “**User Profile**” module is for providing the profile of the user/driver in a suitable format.

The “**Road Layout**” module receives information about the speed limit and road-layout for the specified GPS location which stored in the repository.

The “**Encode Information**” module is for encoding/packaging information (speed, location, and user identifier) that is to be sent to BBD-Ops in a suitable format which has a small size with enough information so it can be transmitted fast and easily which also minimises the use of cellular data for the driver.

The “**Send data to BBD-Ops**” module is for sending the packaged information to the BBD-Ops where all the information is processed and correctly distributed.

The “**Manager**” module is for monitoring and controlling of the system and for processing some special circumstances. It will be providing the interface on the mobile devices for them to control and monitor the information gathered through the journey. It will be having a look at the repository and the road layout and send the information in correct order depending on the time to BBD-Ops. If it detects drivers erratically driving or going over the speed limit, it will send information with a special code to BBD-Ops which gives the priority so it can be delivered to the police fast. This module will be in charge of starting a new journey as well.

**BBD-M Module Structure**

Receive Information

Process Information

Repository

Display Information

Manager

Encode/decode Information

Diagram 2: “Uses” Structure for BBD-M (arrows represent “uses” relationships and the texts represent modules)

The **“Receive Information”** module is for receiving relevant information from BBD-Ops. The information differs depending on what type of user it is. This will be decided by BBD-Ops. The information being received will be based on the location or the registered driver by the user of BBD-M.

The **“Encode/Decode Information”** module is for decoding the information received from BBD-Ops into the appropriate format that can be processed.

The **“Process Information”** module uses “Encode/decode information” module to unpackage the information received and stores the data in the repository module so it can be viewed anytime by the user.

The **“Display Information”** module uses information stored in the repository module to show information that the user wants to access.

The **“Manager”** module is the central control unit which provides user interfaces. When the special report is received for police it will be prioritized and displayed on top of everything to notify the police user to penalise the unacceptable behaviour by the driver. The information of some driver will be provided with the approval of the driver or confirmation of the user being parents/guardian of the driver if it is not a police.

**BBD-Ops Module Structure**

The BBD-Ops structure is almost same as other “uses” module described above.

The **“Receive Information”** module is for receiving information packaged from the BBD-V application. The information contains information about the journey. The information also may contain a special flag to indicate immediate action may be required by police.

The **“Process Information”** uses **“Decode Information”** to unpackage the information received and store the information received to the **“Information Repository”** module. Using the given information, this module will calculate the length of the journey, the time taken, the maximum acceleration, the number of unnecessary lane changes, the amount of time spent over the speed limit, and the number of times the centre-line is crossed when not at an intersection after the journey has finished. If the flag is found, it will be notified to the “Manager” module who uses this module to send immediate notice to one of the BBD-M police users around the location detected.

Receive Information

Process Information

Information Repository

Send data to BBD-M

Manager

Encode Information

User Profile

Road Layout

Decode Information

Drivers

Diagram 3: “Uses” Structure for BBD-Ops (arrows represent “uses” relationships and the texts represent modules)

The **“Information Repository”** module stores processed information.

The **“Encode Information”** encodes/packages the information that requires to be sent to the specific users which will be decided by the “manager” module.

The **“User Profile”** module is information about the BBD-M users. The “Drivers” module is for the BBD-V users. The “User Profile” itself will be using a different structure for itself which is a generalisation. The module will be the super-class of two types of users. One is normal guardians/parents of registered drivers. Another type will be the police. Depending on what type of user it is, the information that is being sent and when the information is sent will be different.

The **“Road Layout”** module is for information about the speed limit and the lanes that will be continuously updated based on the temporary road works and permeant changes across whole New Zealand.

The **“Send data to BBD-M”** module uses “Encode Information” to send information to BBD-M users. What kind of information to who will be decided by the “manager” module which has access to all information (not just the location and speed).

The **“Manager”** module controls and monitors everything. It compares driver identifier handed in by the BBD-V and its own database to confirm the driver and store the data in correct place under the category of the driver in the “Repository”. When new information is added, finds users related to the driver and encode the information and send it to the BBD-M using “Send data to BBD-M” module. It also sends quick emergency call/message to BBM-D police user. The police user is determined based on the “Road Layout” i.e. location of the driver so police from the closest station can be sent for quick action.

**Overall BBD Allocation Structure**

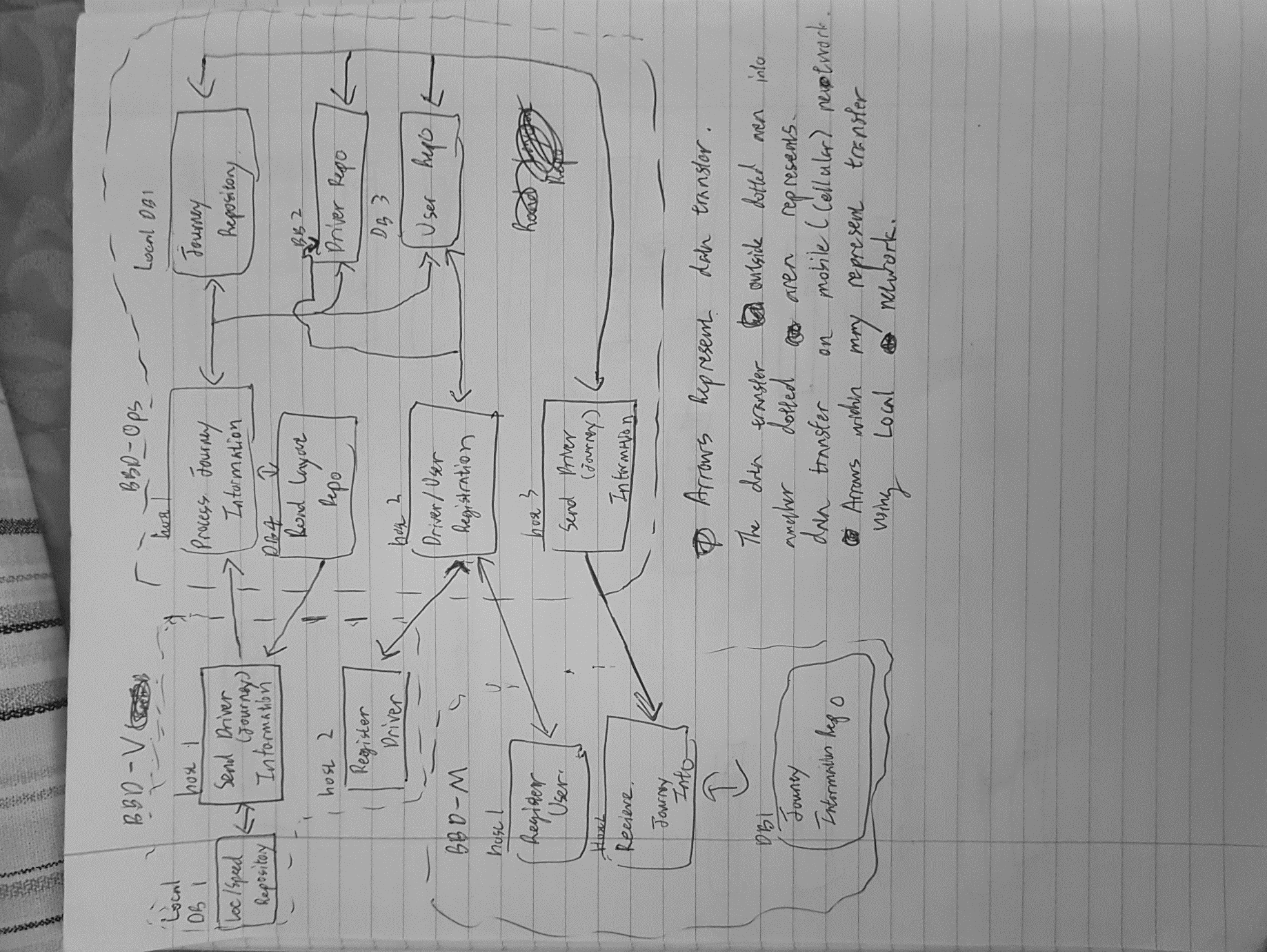


Diagram 4: Allocation Structure for BBD-Ops

Each box may represent combinations of some sub-processes to achieve the statement explaining the state.

The structure is “Deployment” structure which shows how software processes/modules are assigned and communication element that are involved (Explained in diagram 4). There are two main processes that is carried out separately. One is for recording a journey and another process is for online-registering.

Both processes will be run on separate mobile devices as creating journey and registration cannot occur simultaneously.

Online-registration processes will be queued in personal devices running either on BBD-V or BBD-M. The data is passed to processors in BBD-Ops which communicate with its local databases to confirm that the queue is acceptable (i.e. no duplicate user/driver, no permission to add driver not related to the one who requested) which costs a lot to look through the big database.

Manager will be responsible for detecting and monitoring the system in BBD-Ops. As the problem with the driver will be calculated by the BBD-V by looking through the road layout database in BBD-Ops and signal BBD-Ops to dispatch police to deal with the problem. It is not sure whether detecting problem within the BBD-V system which burdens the user and detecting problem within the BBD-Ops by processing the received information from BBD-V as there will a lot of data to be analysed at the same time. The initial decision made is to let BBD-V do the reporting. This may change later depending on the performance.

The journey information when the journey has ended is passed from BBD-V to BBD-M via BBD-Ops. There is one exception when the information is directly passed to BBD-M for the case where the driver is same as the user of BBD-M.

**Tactics**

Tactics for Performance:

* **Increase computational efficiency:** using an efficient algorithm to encode or decode data to be transferred between the applications. The algorithm should be minimising the size of the data and it should be reasonably fast to meet the performance requirements.
* **Reduce computational overhead:** deciding whether to co-locate some processes which will end up with reduction of the overhead of the remote communication. Less communication cost also means that the data can be transferred faster to meet the performance requirements.
* **Manage event rate**: deciding on how frequently to update journey information from BBD-V to BBD-Ops for the drivers that need to be tracked in real-time.
* **Control frequency of sampling:** choice to be made on how often the data should be recorded. More frequent means more accurate data but it will lead to high cost and slower performance.
* **Introduce concurrency:** having multiple servers to process incoming data from many drivers. This will allow many journey information from multiple BBD-V drivers to be processed and get it ready for BBD-M at the same time within the required limit.
* **Increase available resource:** having many processors is essential but having faster processors is also essential as well. It may cost more but it must be done to meet the performance requirements.
* **Maintain multiple copies of either data or computations:** have multiple copies of data and servers. By placing multiple copies of data and server around country, the computation will be done faster and more effectively as the distance from the mobile device and the server decreases. It also ensures the security of data loss or corruption due to many reasons (i.e. natural disaster).

Tactic for modifiability:

* **Split module:** the hardware (displays and user interface) and the computations are well split into separate module which minimises the needs for changing computations when the hardware (a mobile device that needs to be used) is different. So, when new device is released, only the layout and the look of the user interface has to be adjusted which can be done quite quickly i.e. can be done within 40 hours of development and testing.

Tactic for security:

* **Verify message integrity**: confirming that data has not been changed or corrupted to deliver correct information to the related users.
* **Authenticate actors:** authenticate the actor on the mobile device to be the authorised person by making the user trying to access BBD-M to enter a password that matches.
* **Limit access**: Access to the journey information is limited by the BBD-Ops who distribute the information to specific BBD-M users. This reduces probability of sending the information to un-authorised people most of the time i.e. BBD-M user does not have access to all the information on the database in BBD-Ops.
* **Encrypt data**: Make sure the data being sent is encrypted which can only be decrypted by special token or key which is given to authorised users of BBD-M. The BBD-M user with no right would not be able to decrypt data and view it.
* **Lock computer:** the app is being used on personal mobile devices that will be locked and prevent unauthorised people from using the device and the app.

**Justification**

The journey information will be transmitted to BBD-Ops every 1 seconds by “Manager” module in BBD-V. The vehicle moving at 100kph travels about 28 metres in 1 second which is a lot. To observe some drivers in real time, it would not be very useful if the data would be delivered in long interval with not enough data.

To meet the performance requirements the illegal behaviour of drivers needs to be reported to the police within 5 seconds of the occurrence of the problem. There are 2 choices as I explained in earlier section. The one that I have decided to use here is to detect the problem using BBD-V system, not with BBD-Ops. The mobile devices where BBD-V would be running on will have different hardware and it is most likely to be less powerful than the ones that is going to be used for BBD-Ops. Running processes on the mobile device will be slow if the device is bad and may cause a delay in the reporting system. The problem with the BBD-Ops is that it will be dealing with thousands of Journeys information that needs to be processed at the same time. The processors may not be able to handle the traffic quite effectively and fast enough. It would be too expensive to do the calculation of detecting problems from the thousands of information. Whereas, the mobile device will not be in use which means that most of its computational power can be used by BBD-V to do lots of jobs which may be more cost effective and responsive.