Sprint Dates: 4/30 to 5/12

Items to mention

- Assigning Core Components

Schedule

Member	4/30	5/1	5/2	5/3	5/4	5/5	5/6
Jason	4	2	6	6	6	6	2
Jesus	0	6	15	15	0	0	0
Vi	4	4	4	4	4	4	4
Gio	4	4	6	6	6	6	4
Rainier	4	4	3	10	0	2	3

Member	5/7	5/8	5/9	5/10	5/11	5/12
Jason	4	2	6	6	6	6
Jesus	0	6	15	15	0	0
Vi	4	4	4	4	4	4
Gio	4	4	6	6	6	6
Rainier	4	4	3	10	0	2

Team Capacity

Members	Hours Available
Jason	32
Jesus	36
Vi	32
Gio	36
Rainier	26
Team Capacity (Total) =	162

Unavailable Days

Members	Unavailable Days	Reason
Jason		
Jesus		
Vi		
Gio		
Rainier	5/4	Work

Task Breakdown

Work Items Based on Product Backlog

Work Items	Effort Points (Hours)	Assigned Owners
Communication Establishment CE - 1	55	Gio
Communication Establishment CE - 2	55	Gio
Communication Establishment CE - 2	55	Gio
Inventory Management IM - 1	34	Rainier
Inventory Management IM - 2	34	Rainier
Car News Center Peer Review	2	Rainier
Car News Center CNC - 1	26	Vi
Car News Center CNC - 3	26	Vi
Vehicle Marketplace VPM - 1	16	Vi
Vehicle Marketplace VPM - 2	19	Vi
Vehicle Marketplace VPM - 3	12	Vi
Service Log SL - 1	40	Jason
Service Log SL - 2	40	Jason

Service Log SL - 3	40	Jason
Service Log SL - 4	40	Jason
Service Log SL - 5	60	Jason
Service Log SL - 6	30	Jason
Service Log SL - 7	30	Jason
Service Log SL - 8	70	Jason
Service Log SL - 9	40	Jason
User Administration UA - 1	21	Jason
User Administration UA - 2	21	Jason
User Administration UA - 3	21	Jason
Scrap Your Car Peer Review	2	Vi
Scrap Your Car SYC - 1	55	Jesus
Scrap Your Car SYC - 3	55	Jesus
Scrap Your Car SYC - 4	55	Jesus
Scrap Your Car SYC - 5	55	Jesus
Logging Web API	13	Jesus
Vehicle Profile VP - 1	34	Rainier
Vehicle Profile VP - 2	34	Rainier
Vehicle Profile VP - 3	34	Rainier
Vehicle Profile VP - 4	34	Rainier
Vehicle Profile VP - 5	34	Rainier

<u>Donate Your Car</u>

Donate Your Car (DYC - 1) (3)

1. Front end (3)
    a. A page will be created that will display all the charities with their descriptions and a button will be created next to each charity that will hold the link to them

Donate Your Car (DYC - 2) (3)

1. Front End (3)
    a. When the user clicks the button relevant to the charity it will create GetCarDetails that will return the necessary information to fill out the charity forms.
    b. Once the details are returned they are then redirected to the link that was in the button and a script will be used to fill out the online form

<u>Communication Establishment</u>

- **Communication Establishment (CE-1)(20)**
    1. xUnit tests (3)
        a. Create a test for retrieving the correct seller from the database that is associated to the marketplace post by the VIN number.
        b. Create a test that checks if a chat session was created in the database when a request was sent to the seller
    2. Manager Layer (5)
        a. Create CommunicationEstalishmentManager where only the VIN will be sent to the service layer that will use the VIN as the search parameter for ISearchParameters service.
        b. Manager layer will be used again once it has the seller's username and be used to create Notification object that will sen a message to the seller that a person wishes to communicate with them
    3. Models(1)
        a. Create the IChat interface that will be used by the services to create a chat session. Will contain values such as VIN, SessionID, BuyerUsername, SellerUsername, ViewStatus, MessageCreationTime and MessageContent
    4. Service Layer(3)
        a. The first service that be used is GetSellerInfo that will be in charged of retrieving the username of the seller from the database and send them a notification object that the buyer wishes to communicate
    5. Front End(5)
        a. Create a button that requests communication with the seller and will gather the VIN of the post and also the username of the buyer which will be the person who clicked the button.
    6. Entry Point(3)
        a. Create a GetSellerInfo that will use the username of the buyer and the vin of the post they are currently on. This will be sent to the CommunicationEstablishmentManager to search the database for the seller's username.

- **Communication Establishment (CE-2)(19)**
    1. xUnit tests(3)
        a. Create a test for retrieving chats from the database and will use the username of the user to get all chats associated with them

    b. Create a test that messages are saved in the the datastore and are able to be retrieved
2. Manager Layer(5)
    a. Will use GetChats sent by the entry point that will use IUserAccount to get the necessary information of the user such as the username and sent it to the service layer
    b. Manager layer will be used again with SendMessage that will use the IChat to create an object to be sent to the database that will contain the new message in the chat
3. Service Layer(3)
    a. Will send a notification object to the seller
4. Front-End(5)
    a. A page will be created to show all the chat sessions of the user and each chat session will be selectable.
    b. Once a chat has been selected it will display the previous messages of the chat session and the user is able to send new messages
5. Entry point(3)
    a. GetMyChats function will be created when the page is visited that will use IUserAccount to get the information necessary of the user in order to retrieve the chat sessions
    b. Then for when a user wants to send a new message PostSentMessage will be created that will use the IChat to create an object to send to the database.

- **Communication Establishment (CE-3)(19)**
    1. xUnit tests(3)
        a. Create a test that when the seller confirms the deal the Marketplace status is changed correctly to Sold
    2. Manager Layer(5)
        a. DealConfirmation will be created that will get the seller's username and the type of deal such as if the deal was canceled or confirmed.
        b. Also create a SendLocationMessage function that will only contain the location of their meetup
        c. For when deal is canceled DealCancellation will be created and will function the same as DealConfirmation but instead the deal type will be different
    3. Service Layer(3)
        a. A service called UpdateDealStatusSql that will change the status of the MarketplaceStatus to 1 by using the VIN or it the deal was canceled it will be change to 0
        b. Will use the InsertMessageSql service again but will only send the location
    4. Front-End(5)
        a. Create a button that confirms the deal
        b. Create a pop up after user confirms the deal that prompts user to enter

location of where to meet up
    c. Create a button that cancels the deal
5. Entry Point(3)
    a. If the user confirms deal it gets the sellers username and creates the deal type in the ConfirmDeal which will be a POST request
    b. If the user cancels the deal similar steps will be taken but the location is never prompted
    c. A post will also be created when the user is prompted to enter the location that will be used to to enter a new message into the chat.

Vehicle Profile

- **Vehicle Profile (VP-2)** **→ Rainier → 2EP**
    1. Front End → Rainier → 2EP
        a. Create a popup that will let the user see their vehicle information and a form for information that they want to change. Make, Model, Year, LicensePlate, Color, and Description are the only fields that should be editable. Two buttons will be added, “Submit” and “Cancel”
        b. Submit button will submit the vehicle profile with the updated values to the PostUpdateVehicleProfile entrypoint and update the existing vehicle profile in the session storage.
        c. Cancel button will return the user back to the vehicle profile view without calling the PostUpdateVehicleprofile HTTP request
- **Vehicle Profile (VP-3)** **→ Rainier → 2EP**
    1. Front End → Rainier → 2EP
        a. Add a new button on a vehicle profile that will let a user delete a vehicle
        b. Create a popup that will let the user confirm or deny that they want to delete their vehicle. Two buttons will be made, “Submit” and “Cancel”
        c. Submit button will submit the selected vehicle profile to the PostDeleteVehicleProfile entrypoint and delete the existing vehicle in the session storage.
        d. Cancel button will return the user back to the vehicle profile view without calling the PostDeleteVehicleProfile request

Car News Center

- **Car News Center (CNC-1)** → Vi → 10EP
    1. Entry point for Get request to get all news articles for any vehicles in the datastore → Vi → 2EP
    2. JS script method to handle extracting data and request from API
    ` → Vi → 3EP
    3. JS script method to display the information → Vi → 3EP
    4. XUnit test: → Vi → 2EP
        a. 1 for fail case when news articles are not received from API → Vi → 1EP

    b. 1 for fail case when vehicles are not retrieved successfully from datastore
→ Vi → 1EP

- **Car News Center (CNC-3)** → Vi → 11EP
    1. Entry point for Get request to show all current alerts in datastore → Vi → 2EP
    2. Entry point for Post request to create new alerts to datastore
→ Vi → 2EP
    3. JS Script for extracting and generating view for displaying all notifications in datastore → Vi → 6EP
    4. XUnit test: → Vi → 1EP
        a. 1 for fail case when notifications object are not retrieved from datastore
→Vi → 1EP

<u>Vehicle Marketplace</u>

- **Vehicle Marketplace (VPM-1)** → Vi → 7EP
    1. Entry point for Post request to upload vehicle to marketplace
→ Vi → 2EP
    2. JS Script to generate a page when the upload process is successful
→ Vi → 3EP
    3. XUnit test: →Vi → 2EP
        a. 1 for fail case when vehicles is not uploaded successfully to the datastore
→Vi → 1EP
        b. 1 for success case when vehicles are uploaded successfully to datastore
→Vi → 1EP
- **Vehicle Marketplace (VPM-2)** → Vi → 7EP
    1. Entry point for Post request to delete vehicle from marketplace
→Vi → 2EP

    2. JS script to generate page when the delete process is successful
→Vi → 3EP
    3. XUnit test: →Vi → 2EP
        a. 1 for fail case when vehicle is not deleted successfully from the datastore
→Vi → 1EP
        b. 1 for success case when vehicle is successfully deleted from datastore
→Vi → 1EP
- **Vehicle Marketplace (VPM-3)** →Vi → 12EP
    1. XUnit test: →Vi → 4EP
        a. 1 for fail case when vehicles are not fetched successfully from the datastore →Vi → 1EP
        b. 1 success case for pagination when retrieve all vehicles from datastore

→Vi → 1EP

c. 1 fail case for pagination when retrieve all vehicles from datastore
→Vi → 1EP

d. 1 success case for all vehicles fetched successfully from datastore
→Vi → 1EP

Scrap Your Car **→ Jesus** **→ 36 EP**

- **Scrap Your Car (SCY-1)** **→ Jesus** **→ 13 EP**
    1. XUnit Tests (6)
        i. Test to validate parts service can create
        ii. Test to validate part service can retrieve matching
        iii. Test to validate part service can retrieve all (UID)
        iv. Test to validate part service can remove
        v. Test to validate listing service can create
        vi. Test to validate listing service can retrieve matching
        vii. Test to validate listing service can retrieve all
        viii. Test to validate listing service can update
        ix. Test to validate listing service can remove
        x. Test to validate Manager default parts list
        xi. Test to validate Failing database error
    2. Service and Target(5)
        i. Creation of DTO's, interfaces + classes
        ii. Implementation so parts service can create
        iii. Implementation so part service can retrieve matching
        iv. Implementation so part service can retrieve all (UID)
        v. Implementation so part service can remove
        vi. Implementation so listing service can create
        vii. Implementation so listing service can retrieve matching
        viii. Implementation so listing service can retrieve all
        ix. Implementation so listing service can update
        x. Implementation so listing service can remove
    3. Manager + Entry Point(2)
        i. Creation of Interface + Class
        ii. Implement Create Default parts list

- **Scrap Your Car (SCY-3)** **→ Jesus** **→ 2 EP**
    1. XUnit Tests (1)
        i. Test to validate "List Individual Part"a Manager Layer
        ii. Test to validate Failing database error
    2. Manager + Entry Point (1)
        i. Implementation so Manager can create individual parts

- **Scrap Your Car (SCY-4) → Jesus → 6 EP**
    1. XUnit Tests (2)
        i. Validate target returns correct values and order
        ii. Validate target has proper pagination
        iii. Validate target no results show up, if no values match
        iv. Validate service returns no results if no search values present
        v. Test to validate Failing database error
    2. Service and Target (2)
        i. Creation of DTO's, Interfaces, and classes
        ii. Implementation of search function
    3. Manager + Entry Point (2)
        i. Implementation of manager layer functions
        ii. Implementation of endpoints
- **Scrap Your Car (SCY-5) → Jesus → 11 EP**
    1. XUnit Tests (5)
        i. Validate service can create buy requests
        ii. Validate service can retrieve User's buy requests
        iii. Validate service can retrieve Incoming buy requests
        iv. Validate service can retrieve Matching buy requests
        v. Validate service can update buy requests
        vi. Validate service can delete buy requests
        vii. Validate manager can create buy requests
        viii. Validate manager can get buy requests
        ix. Validate manager can update buy requests
        x. Validate manager can delete buy requests
        xi. Test to validate Failing database error
    2. Service and Target (4)
        i. Creation of DTO's, Interfaces, and Classes
        ii. Implement create buy request functionalities
        iii. Implement retrieve outgoing buy request functionalities
        iv. Implement retrieve incoming buy request functionalities
        v. Implement retrieve matching buy request functionalities
        vi. Implement update buy request functionalities
        vii. Implement delete buy request functionalities
    3. Manager + Entry Point (2)
        i. Implementation of Manager Layer creates
        ii. Implementation of Manager Layer retrieves
        iii. Implementation of Manager Layer updates
        iv. Implementation of Manager Layer deletes
- **Scrap Your Car (SCY-6) → Jesus → 4 EP**

1. XUnit Tests (2.5)
    i. Validate service can approve buy requests
    ii. Validate service can reject buy requests
    iii. Validate manager can check if buy requests are valid
    iv. Validate manager can approve buy requests
    v. Validate manager can reject buy requests
    vi. Test to validate Failing database error
2. Service and Target (1)
    i. Implement approve buy request
    ii. Implement deny buy request
3. Manager + Entry Point (0.5)
    i. Implement functionality to validate business rules, and pass to correct service layer function

<u>Logging Web API</u> **→ Jesus** **→ 5 EP**

- **Front End Function**
    1. Testing: "CreateLog()" Front end Testing
        - 5 EP

<u>Service Log</u>

- **Service Log (SL 1 – 4)** **→ Jason** **→ EP**
    1. Check for way to enforce 100 SL limit in Sql → Jason → EP
    2. Create Valid Service Log test case → Jason → EP
    3. Create Invalid Service Log test case → Jason → EP
    4. Create multiple valid service logs test case → Jason → EP
    5. Create createSL manager function → Jason → EP
    6. Implement business logic in createSL function → Jason → EP
    7. Create SL Web API project → Jason → EP
    8. Create SL controller → Jason → EP
    9. Create SL controller function CreateSL → Jason → EP
    10. Manually test SL 1 - 4 with swagger → Jason → EP
    11. Create Front - End default forum for SL input → Jason → EP
    12. Implement CreateSL JS function → Jason → EP
    13. SL 1 Front - End logic for handling requests → Jason → EP

- **Service Log (SL 5)** **→ Jason** **→ EP**
    1. Create retrieve 10 valid SL test case → Jason → EP
    2. Create retrieve <10 valid SL test case → Jason → EP
    3. Create retrieve 0 valid SL test case → Jason → EP
    4. Create retrieve incrementing page valid test case → Jason → EP
    5. Create retrieveSL manager function → Jason → EP

6. Implement business rules in retrieveSL function → Jason → EP
7. Create SL controller function RetrieveSL → Jason → EP
8. Manually test SL 5 with swagger → Jason → EP
9. Create SL default view → Jason → EP
10. Implement SL retrieve SL function → Jason → EP
11. Create SL default view pagination logic → Jason → EP
12. Create response handling logic for SL 5 → Jason → EP

- **Service Log (SL 6) → Jason → EP**
    1. Revise SL - 6 Sql Target → Jason → EP
    2. Revise SL - 6 Service layer function → Jason → EP
    3. Create successful SL 6 edit test case → Jason → EP
    4. Create ModifySL manager function → Jason → EP
    5. Implement ModifySL business rules in manager → Jason → EP
    6. Create SL controller function ModifySL → Jason → EP
    7. Test SL 6 with swagger → Jason → EP
    8. Create Edit SL input forum → Jason → EP
    9. Create Edit SL Js function → Jason → EP
    10. Create response handling logic for SL 6 → Jason → EP

- **Service Log (SL 7) → Jason → EP**
    1. Create successful SL 7 test case → Jason → EP
    2. Create DeleteSL manager layer function → Jason → EP
    3. Implement DeleteSL business rules in function → Jason → EP
    4. Create SL controller function DeleteSL → Jason → EP
    5. Test SL - 7 with Swagger → Jason → EP
    6. Create DeleteSL Js function → Jason → EP
    7. Create “Are you Sure” function → Jason → EP
    8. Create response handling logic for SL 7 → Jason → EP

- **Service Log (SL 8) → Jason → EP**
    1. Create category filter in view for filtering SL’s → Jason → EP
    2. Create Date filter in view for filtering SL’s → Jason → EP

User Administration

- **UA - 1 → Jason → EP**
    1. Revise CreateAccount _Pass sequence diagram → Jason → .5
    2. Revise CreateAccount_Datastore_Fail sequence diagram → Jason → .5
    3. Revise CreateAccount_Business_Fail sequence diagram → Jason → .5
    4. Create UA - 1 class diagram → Jason → 1

5. 4/24 meeting Vong for feedback on UA - 1 LLD → Jason → .1
6. Rewrite CreateValidAccount method → Jason → 2
7. Rewrite GenerateDefaultClaims method → Jason → 2
8. Create function call to emailing service → Jason → 2
9. Add missing log functionalities in BRD to service → Jason → 2
10. Create new User Administration Manager → Jason → 1
11. Create new User Administration Manager interface → Jason → 1
12. Check for valid business rule (DOB) in old code → Jason → 1
    i. Date of Birth
    ii. Username
    iii. Account Type

- **UA - 2 → Jason → EP**
  1. Create RecoverAccount success sequence diagram → Jason → .5
  2. Create SendAdminRequest success sequence diagram → Jason → .5
  3. Create ConfirmRequest success sequence diagram → Jason → .5
  4. Create RecoverAccount_Fail sequence diagram → Jason → .5
  5. Create SendAdminRequest_Fail sequence diagram → Jason → .5
  6. Create ConfirmRequest_Fail sequence diagram → Jason → .5
  7. Create UA - 2 Class diagram → Jason → .5
  8. 4/24 meeting Vong for feedback on UA - 2 LLD → Jason → .1

- **UA - 3 → Jason → EP**
  1. Create DeleteAccount success sequence diagram → Jason → .5
  2. Create DeleteAccount_Datastore_Fail sequence diagram → Jason → .5
  3. Create DeleteAccount_Business_Fail sequence diagram → Jason → .5
  4. Create UA - 3 Class diagram → Jason → .5
  5. 4/24 meeting Vong for feedback on UA - 3 LLD → Jason → .1
- **UA - 4 → Vi →**
  1. Create UserManagment View success sequence digram → Vi → 1EP
  2. Create UserManagment Edit Account success sequence digram → Vi → 1EP
  3. Add new method to Target to return all user accounts information from datastore → Vi → 3EP
  4. Add new method to Target to let admin edit user accounts in datastore → Vi → 5EP
  5. Create service to return all user accounts information from datastore → Vi → 1.5EP
  6. Create service to call on target for editing user account → Vi → 1.5EP
- **UA - 5 → Vi →**
  1. Create RequestData success sequence diagram → Vi → 1EP

2. Add new method to Target to return all information relating to a specific account → Vi → 3EP
3. Create a service to return all information relating to a specific account → Vi → 2EP

- **UA - 6 (9)**
    1. Create the LLD for UA-6 with the success and fail cases (3)
        a. Create the UpdateAccount success sequence diagram
        b. Create the UpdateAccount fail sequence diagram
        c. Create UA - 6 Class diagram
    2. Create the Back-End (10)
        a. Create the necessary manager based on the LLD
        b. Create the necessary service based on the LLD
        c. Create the necessary target based on the LLD
        d. Create the necessary entry point based on LLD
    3. Create the Front(3)
        a. Create the necessary scripts which will most likely be a POST that will update the user account with parameters being Address, Name, and Phone number

<u>Claim Service</u>

- **Create User Claims → Rainier → 4 EP**
    1. Create LLDs for Pass Scenario
        a. Valid claim, claimscope, and user ID
        b. Valid claim is written to database
    2. Create LLDs for Fail Scenario
        a. Invalid user is passed in. No user ID is passed in
        b. Invalid claim passed in. Claim is not a valid claim in database
        c. Database error. Could no write to database
- **Modify User Claims → Rainier → 4 EP**
    1. Create LLDs for Pass Scenario
        a. Valid claim, claimscope, and user ID
        b. Valid claim is updated in database
    2. Create LLDs for Fail Scenario
        a. Invalid user is passed in. No user ID is passed in
        b. Invalid claim passed in. Claim is not a valid claim in database
        c. Database error. Could not write to database
- **Delete User Claims → Rainier → 4 EP**
    1. Create LLDs for Pass Scenario
        d. Valid claim, claimscope, and user ID
        e. Valid claim is deleted from database
    2. Create LLDs for Fail Scenario

a. Invalid user is passed in. No user ID is passed in
b. Invalid claim passed in. Claim is not a valid claim in database
c. Database error. Could no write to database

System Observability

- **SO - 1** **→ Rainier** **→ 6EP**
  1. Create LLDs for Pass Scenario → Rainier → 1EP
  2. Create LLDs for Fail Scenarios → Rainier → 2EP
     a. Data is invalid
     b. Datastore Error
  3. Create xUnit tests for Pass Scenario → Rainier → 4EP
     a. All data passed in is valid
     b. Number of login attempts is valid
     c. Number of account creation attempts is valid
     d. Top 3 views of a website is valid
     e. Top 3 most visited views is valid
     f. Top 3 most registered cars is valid
     g. Number of vehicle creation attempts is valid

Task Distribution

Members	**Hour // EP remaining**	**Tasks**
Jason		•
Jesus	20 - 40.5-6.5 = -27	• **SYC (1,3-6) Design (40.5 EP)** • **Logging Web API (6.5)**
Vi	28-47= -19	• **Car News Center (21EP)** ○ CNC-1 (10EP) ○ CNC-3 (11EP) • **Vehicle Marketplace (26EP)** ○ **VPM-1 (7EP)** ○ **VPM-2 (7EP)** ○ **VPM-3 (4EP)**
Gio	36-64-9= -37	**Donate Your Car (DYC - 1) (3)** • Front End(3) **Donate Your Car (DYC - 2) (3)** • Front End(3) **Communication Establishment (CE-1)(20)**

		• xUnit(3) • Manager Layer(5) • Models(1) • Service Layer(3) • Front End(5) • Entry Point(3) **Communication Establishment (CE-2)(19)** • xUnit(3) • Manager Layer(5) • Models(1) • Service Layer(3) • Front End(5) • Entry Point(3) **Communication Establishment (CE-3)(19)** • xUnit(3) • Manager Layer(5) • Models(1) • Service Layer(3) • Front End(5) • Entry Point(3) **Update User Account(UA-6)(16)** • Create LLD for UA-6 (3) • Create Back-End(10) • Create Front-End (3)
Rainier	26 - (2 + 2) - (6) - (4 + 4 + 4) = 4	**Vehicle Profile** • VP-2 (2 EP) • VP-3 (2 EP) **System Observability** • SO-1 (6 EP) **Claim Service** • Create Claims (4 EP) • Modify Claims (4 EP) • Delete Claims (4 EP)

Final Analysis

Do we accept the estimates for:

Example

Jesus	
Jason	
Vi	
Gio	
Rainier	

Service Log

SL - 1 2 3 & 4 (5 EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

SL - 5 5 EP	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

SL - 6 - 5 EP	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes

Rainier	Yes

SL - 7 5 EP	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

SL - 8 5 EP	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

SL - 9 5EP	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

User Administration

UA - 1 20 EP

Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

UA - 2 10 EP	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

UA - 3 15 EP	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Donate Your Car

DYC-1 (11EP)	
Jesus	Yes
Jason	Yes
Vi	Yes

Gio	Yes
Rainier	Yes

DYC-2 (10EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Communication Establishment

CE-1 (20EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Car News Center

CNC-1 (10EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

CNC-2 (11EP)

Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Vehicle Marketplace

VPM-1 (7EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

VPM-2 (7EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

VPM-3 (12EP)	
Jesus	Yes
Jason	Yes
Vi	Yes

Gio	Yes
Rainier	Yes

Vehicle Profile

VP-1 (5 EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

VP-2 (4 EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

VP-3 (4 EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

VP-4 (2 EP)	
Jesus	Yes

Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

VP-5 (2 EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Logging Web API

Logging API (12EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

System Observability

SO-1 (4EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

IM-2 (4EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Claim Service

Create Claims (4EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Modify Claims (4EP)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Delete Claims (4EP)	
Jesus	Yes
Jason	Yes
Vi	Yes

Gio	Yes
Rainier	Yes

Scrap Your Car

SYC-1 (12)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

SYC-3 (4.5)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

SYC-4 (8)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

SYC-5 (10.5)

Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

SYC-6 (5.5)	
Jesus	Yes
Jason	Yes
Vi	Yes
Gio	Yes
Rainier	Yes

Are we within our Sprint Capacity?

Work Item	**New Estimate**
Donate Your Car DYC-1	3
Donate Your Car DYC-2	3
Communication Establishment CE - 1	20
Communication Establishment CE - 2	19
Communication Establishment CE - 2	19
Car News Center CNC - 1	2
Car News Center CNC - 3	5

Vehicle Marketplace VPM - 1	5
Vehicle Marketplace VPM - 2	5
Vehicle Marketplace VPM - 3	5
Service Log SL - 1 - 4	10
Service Log SL - 5	10
Service Log SL - 6	10
Service Log SL - 7	10
Service Log SL - 8	10
User Admin. UA - 1	15
User Admin. UA - 2	15
User Admin. UA - 3	15
User Admin. UA - 4	15
User Admin. UA - 5	15
User Admin. UA - 6	16
Scrap Your Car SYC - 1	10
Scrap Your Car SYC - 3	10
Scrap Your Car SYC - 4	10
Scrap Your Car SYC - 5	10
Scrap Your Car SYC - 6	10
Vehicle Profile VP - 1	2
Vehicle Profile VP - 2	2
Vehicle Profile VP - 3	2
Create claims	4
Modify claims	4
Delete claims	4

Total	297

Will We Add More Work

No, we will not add more work due to being over capacity.