

Individual Evaluation

Tudor Stefan Haba

2420645

Table of Contents

Project Brief.....	3
Week breakdown	3
Monday - Planning	3
Tuesday.....	4
Wednesday	5
Thursday - Testing	6
Week retrospective	6
What went well	6
What went bad	6
What I would do differently	6
Appendix	7
Links	7
Figures	7
Minutes	8
Mins of Meetings – 16th December.....	8
[1] Mins of Meeting - Morning scrum 17th December	8
[2] Mins of meeting (Morning scrum) - 18th December.	9
[3] Mins of meeting (19th December).....	10
[4] Retrospective - 20th December	12

Project Brief

The goal of this project was to have a booking system that can be used by customers as well as staff. Part of the challenge of this project was working in a team of 4 in a short period of time. We decided that our MVN [Fig.1] should include the ability for customer to register an account, login into the system make a booking and browse all vehicles. The staff should be able to add and remove vehicles, list all vehicles and book a vehicle for repairs or maintenance.

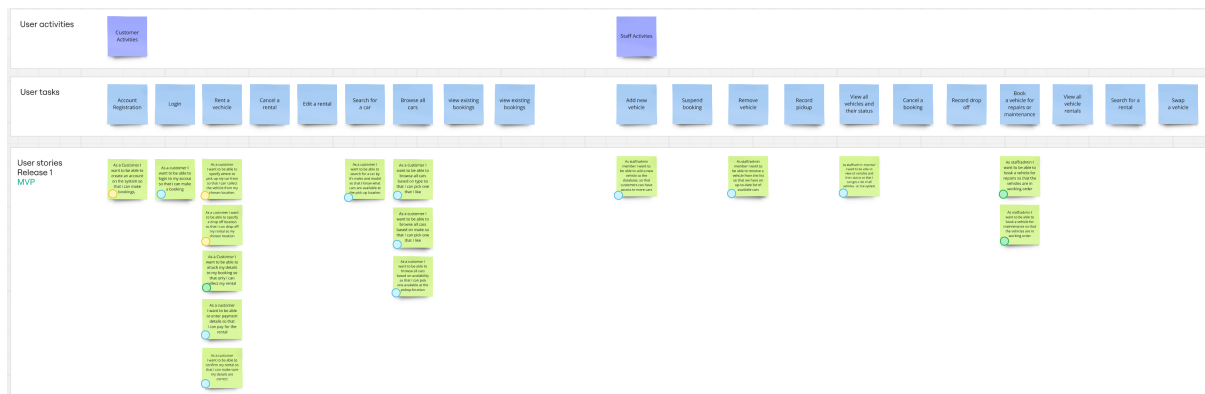


Fig.1 MVP

Week breakdown

Monday - Planning

After reading the project brief, we decided to start planning the database. We had an initial discussion of how the database should look like and we started planning it out on a whiteboard. When we were all happy with the breakdown of the database in the individual tables, we used Visual Paradigm to design an ERD [Fig. 2]. As a team we discussed in detail what attributes we should add to every table and what the primary key should be as well as discussing what the relationship of the tables would be like.

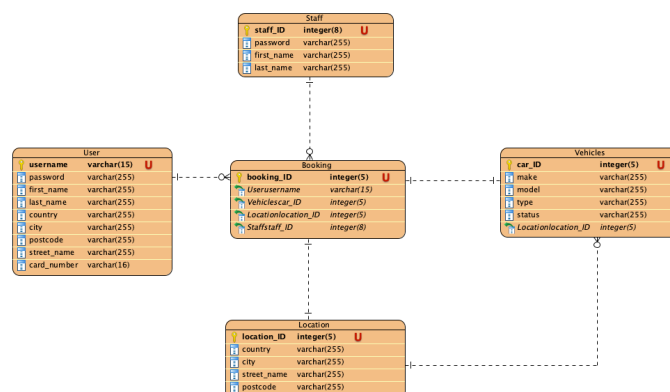


Fig. 2 ERD

As a team we used agile and we discussed how to break down the project brief into smaller user stories. The software we used to create a user board and add our user stories was Miro. We then had a discussion around the room on what the MVP product should contain and what should go into future releases. Creating a booking was our main task for the MVP, editing a booking was something we decided should go into the second release and any extra feature not included in the brief that we came up with went into the third release. We deemed making a booking critical for the system and the other stories less important for our objective.

Every task has then been added to a Trello backlog so that we can take up stories from it in the upcoming standups. We used planning poker to assign point for every task on Miro starting from 1 to 5 requiring the most effort [Fig. 3].

Tuesday

On the standup [1] we held for Tuesday the first task I wanted to work on was implementing the design of the database in pgAdmin as I have less experience with this and I wanted to learn more. Ryan had the same idea and since we both wanted to implement the database, we thought that pair programming would be perfect as working on the same SQL file would be difficult. Each of us spent 5 to 10 minutes after which we would switch so this way I worked on implementing some of the database as well as implementing some of the mock data.

After I finished working on the database the next story I picked from the backlog was registering a new user [Fig. 4]. This was a short task that required to get user input and then store it on the database. The task was initially marked as a 3/5 by the team but after implementing it I feel it should be 1/5 based on how much effort it actually required. The next story I picked up was adding a new vehicle to the database from the staff's perspective [Fig. 5]. Whilst doing this task I realised we don't currently have a way for the staff to login into the system. I wanted to use the login that was already implemented by someone else instead of adding a separate login for the staff. I added another function that would take in the username that was entered and it would check against the staff database first to see if it matches any staff IDs. If the username matched then I returned the staff ID back and confirm in the login function that this username is a staff member and display the staff menu instead of the customer one. After setting the staff login I could go back and start working on adding new cars to the database. This was asking for user input and storing it. However, any vehicle needs to have a location as location is a foreign key. I had to make a new function that would print all the items from a table. I wanted this function to be generic so it could be used by other people as well if needed, what I did was pass in the name of the table I want to retrieve the data from and then print a list of everything in that table. Now the staff was able to view everything required to add a new vehicle so I could finish up my task by storing everything to the database. As a team

we decided this was a 1/5 task however we did not take into consideration all the other function needed to get it to work and I believe this should be closer to 3/5 based on the amount of time it ended up taking.

Wednesday

In the daily standup [2], the first user story I picked for the day was customers being able to search for a vehicle by entering the make and model [Fig. 6]. For me this was a quick task as all I did was take in the user input and query the database to return a list of records matching the make and model entered by the user. The next two stories I picked for today was allowing staff to book vehicles for repairs [Fig. 7] and to book vehicles for maintenance [Fig. 8]. The reason I picked up both tickets was because they were very similar to each other, and I talked with the team to make a function that would handle both depending on the arguments passed. The function I did required a string to be passed which was either 'in repair' or 'in maintenance' and then it would update the vehicles status to what has been picked prior. Whilst doing these two tickets I realised that I needed to list all the vehicles currently stored so the staff could pick the ID of the vehicle they wanted to update. This was another story [Fig. 9] that no one was working on so I picked this ticket up as well. Initially I was not displaying the location along with the vehicles but after some team discussion it was requested that I add this as well as it would make it easier for the staff to see exactly what vehicle they are updating but it would also be useful for the booking system another team member was working on.

At this point every ticket in the MVP was either in progress or completed so I picked up a ticket from the next release. The final story I was working on for the day was allowing a staff to list all the details of an individual booking [Fig. 10]. I created a function that would join multiple tables together in order to retrieve all the information I needed for the booking. What I encountered during the implementation of this was because we set up the database with future implementations in mind bookings could either be staff created or user created. When retrieve data for customer created ones I needed a separate SQL query that would not join on the staff table as it cannot be joining on null. I added a function that would take in the booking ID and check whether the booking was staff created, if it had a staff ID, or if it was customer created, no staff ID. Using that function I then use the correct SQL query to retrieve the data.

Thursday - Testing

On Thursday during the standup [3] I decided that I would look into testing. I started out by doing some JUnit testing on the helper function I added. The tests I was able to do before I came into a big issue were checking that the staff login returns the expected output and checking whether a booking is created by a staff or not. The issue I came across was the way we implemented the code made testing very difficult as most of our function are voids and take in no arguments. I then started looking into Mockito. After I managed to set it up for the project I tried doing some test using it, and left the evidence in the repository, but I came up onto another issues. As our functions ask for user input and store it to the database Mockito tests would get stuck in an infinite loop asking for input. This made it impossible for us to test the code as we would have to change how we handle the functions and the limited timeframe did not allow it.

Week retrospective

On Friday the team held a week retrospective [4].

What went well

The use of git and GitHub as a team was really good, we did frequent commits and we didn't have many problems with conflicts even though we were all working on the same file. Communication between the team was also great, we helped each other with any issues or bugs we encountered.

What went bad

One of the things that went bad was the problem one of our team members was having with pgAdmin. Wednesday we were one person down for most of the day. The other main issue we had was testing our code.

What I would do differently

The first thing I would change would be to write our code in a more testing friendly manner. Meaning that I would suggest having more arguments passed into function and having the function return something that would be easy to test using our current knowledge. Using classes would have been another good example of how the code would be more testable as we would have to pass data between classes.

What I feel lacked was our experience with testing and leaving testing to the last day didn't help. If we started testing earlier, perhaps as soon as we finished implementing a story, we would have had more time to realise we are making it hard for ourselves to test the code. If we caught the problem early it would have been easier for us to change how we were coding and make it more test friendly.

Appendix

Links

GitHub: <https://github.com/RLE97/Car-Rental-Fury>

Miro:

https://miro.com/welcomeonboard/QUZFeDRhamJheGxyRzZra0xoV3Aza0hUMGpzQUYVHI1R0pMZHHiOWRMci9lWlpVa2swOFpaNGJyaFBnaHo2aytkTy9vb2xWR2wwTDZKMjVTZ0lEY2VGT1lzWnE1NDNNbDNTZlBFRTZiZTk0bG1GbEM1YmhySFYyUlBuaHVZQWUhZQ==?share_link_id=78923288792

Trello:

<https://trello.com/invite/b/676054e5343c04ba3fd4f598/ATTI05e4da2c14f4e078bb21c7a91574244cE198AAE8/carrental-fury>

YouTube: https://youtu.be/nM_lPsqfDBU

Figures

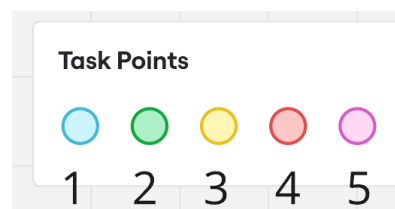


Fig. 3 Point System

As a Customer I want to be able to create an account on the system so that i can make bookings,

Fig. 4

As staff/admin member I want to be able to add a new vehicle to the database, so that customers can have access to more cars

Fig. 5

As a customer I want to be able to search for a car by it's make and model so that I know what cars are available at the pick up location

Fig. 6

As staff/admin I want to be able to book a vehicle for repairs so that the vehicles are in working order

Fig. 7

As staff/admin I want to be able to book a vehicle for maintenance so that the vehicles are in working order

Fig. 8

As staff/admin member I want to be able to view all vehicles and their status so that I can get a list of all vehicles on the system

Fig. 9

As staff/admin I want to be able to search for specific rentals so that i can see the details of that rental

Fig. 10

Minutes

Mins of Meetings – 16th December

- Decided on how to design the database
- Made a ERD diagram in pgAdmin
- Discussed the primary keys, foreign keys, relationships of the tables
- Used Miro to split the brief into stories
- Went around the room and decided what our MVP should include
- Went around the room and decided how on a points system and then assigned point to every story
- Made a Trello board for our backlog

[1] Mins of Meeting - Morning scrum 17th December

Transcript

Yeah, I'm going to take it from here. So. Yeah, yesterday we did planning. Everyone worked on planning. Today I'm going to. I want to work on the database. So I want to create a database, maybe populate it as well.

And I was thinking maybe two of us do the database, two of us do the programming and then the ones that do the database maybe do like pair programming so each of each other can like go for like 10 minutes, five minutes.

Yeah, okay. Yeah that sounds fine. I'm going to pass it right now. Yeah. So yes, they obviously need to plan in. Everyone checked in. They got the database stuff done as well while the database design done.

I'm happy to do the database with you today if you want like you say do pair programming because we can't really look at it any other way.

Yeah, but yeah, I think that's it. So over to yourselves. Yes, yesterday was pretty much planning. Since you guys are going to be working on the database, me and Malik.

Malik can start with connecting the database to the through code.

Yeah. If you start off with creating the database having a password and a username, we'll take a copy of that and then it'll start.

Yeah, yeah. Connecting it through code and then we'll go from there and I suppose like looking at the different stories and which one we comfortable doing.

Yeah, sounds good. Yeah, I think we could probably get. If we just get the big. Well saying that just to get it linked up. I can't remember. I think you just need to link to that database. The link, the username, the password. Yeah, I think so. I think if we get that

[2] Mins of meeting (Morning scrum) - 18th December.

Transcript

Okay, so this is the morning scrum. Yesterday I did the display menu for logging in registration and quitting in Java.

And then in the afternoon I faced issues with database not working.

Then Ryan on a fix it early evening. Today I'm going to be working on the ticket for as staff admin.

I want to be able to view all vehicles their status so I can get a list of all the vehicles.

Honest. Wait, no, sorry, that's the wrong one. So I'm going to be working on removing a vehicle from a list so that we have an up to date list of available cars.

And that's pretty much it from me for now. Mo do you want to. So yesterday I helped with setting up the connection to the database, creating a login function and then I used try and menu to browse cars by make model.

Today I'll be working on browsing cars by type and availability.

No hurdles so far. I'll pass it to the other. All right, so yesterday I work with Ryan to create a database to populate the database with mock data.

Then after we finish the database around probably somewhere midday, I took the ticket to create an account on the system so that I can make bookings.

So register now works. I also changed some something with the connection so it's pre it up into more functions and makes it easier to read into main.

I worked on the stuff login and I also worked on searching for a car by its make and model.

So now that when a user wants to search if a car is on the database they can just input the make and model of that car and they will get a list of all the cars that are on that Megan order.

Currently I'm doing adding a new vehicle to the database so for that I needed the staff login and after I finish that I'm going to do other stuff.

I want to be able to book a vehicle for repairs so that the vehicles are in working order.

Don't think there were any problems for me yesterday.

I'm gonna pass it on to Rhino. So yesterday started the day off with setting up the database with Tudor, populating it with some mock data that took us up to about midday.

After that I helped Malik in trying to resolve his database, his G admin issues which have most of the afternoon and then finally picked up the tasks to create the rental menu and then started with the select pickup locations.

That's complete. And today I'm gonna do the select drop off first and then link up the select vehicle to the browsing for vehicles that Sal has made and then work on the confirmation so it can actually pull all that data together and put it into the bookings menu.

The only hurdles. I need to put a couple of variables into the stuff you guys have made just to know that it's a confirmation or that it's a booking being made rather than just a pring from the get go.

That should be easy enough to come over. And that's about it for me. Perfect. Thank you everyone.

[3] Mins of meeting (19th December)

Transcript

So yesterday I worked on first of all doing the drop off location in the confirm rental section so you can choose where the vehicle will be dropped off.

Then I moved on to doing the vehicle selection with going through the pickup location.

You then get put into the vehicle selection where it only if you can ask for a list of vehicles and it'll only show you a list of vehicles from that specific location that are available.

And then once you select that, it'll take you back to the car rental screen.

And then I worked on creating a confirmation of the booking so that it displays all of the customer details, the locations the vehicle selected and then you can confirm that and that creates the booking and sends it through to the database.

Also worked on the exit, like exiting application validation so that it takes you into a menu, it'll tell you, ask you are you sure you want to quit it?

Yes or no. And then if you hit yes, like exit the application, if you hit no, it'll take you back to the menu you exited from.

So I think it only currently works on the login and the customer menu, but we can build it well, we can build it into any menu that is in we've set up yesterday, but obviously I'm not sure which ones you set up yesterday and today I think I'm going to polish up some of the code and then start doing testing.

I might do testing and then once we finish that, polish up some of the code and then.

Yeah, I think that's it for me. I'll pass it to Tudor. Yesterday I worked on the staff side of the project.

So now the staff is able to book a vehicle for maintenance in that I needed to list all the vehicles and link it up to the location they were at.

The staff is also able to same with the maintenance similar to that, able to book a vehicle for repairs again using the same function I created to list all of the vehicles available and their location also.

Yeah, I worked on making that list and joining it to the location table.

So now the staff can view a list of all the cars and their corresponding location without making any changes to it to the status of the vehicles.

I also worked on the staff being able to search for a booking.

So now staff basically can look at the whole booking details, see who booked it, the vehicle details, the location was booked at the drop off point.

I'm not sure that I did that pickup point. Might have to do that later. And I think this is everything I did yesterday. Might be missing something, don't remember. Oh, I did the list that Ryan was using for the vehicle.

So I changed it so that everything that uses a similar list uses the list I implemented so that we have somewhat of a cohesive menu and response to the user.

Think there were any problems for me yesterday. Today I'm gonna look into setting up testing and yeah, do some tests and yeah, I'm gonna pass it on to Sal now.

So yesterday I was working on the customer side of things.

I was working on displaying and browsing all cars by availability and then I moved into fetching all existing bookings for all customers on the customer side of things.

I'm a bit stuck on that at the minute. Having issues with joining tables and selecting the required fields.

Today I'll be working Graphicstar and then I'll move into some unit testing, polishing up the code.

I'll pass it over to Malik. So yesterday I worked on the staff admin so being able to remove a vehicle from the list so we can have like an up to date list of available cars.

Also worked on presenting the present rentals for bookings and today I'm hoping to do some testing and you know, adding some comments to the code and yeah, that's pretty much it.

And also finishing my ticket. Just do some more testing on that. But that's pretty much it from me.

[4] Retrospective - 20th December

Transcript

Ready? We will go around the room, discuss what went well, what didn't go so well and things we could do differently.

I'll start with what went well. I think the use of git GitHub was pretty good for the whole team.

Yeah, I think it was really good. Especially admin review to approve each pull request.

We had a really good working environment, very supportive and we ran into issues.

What else? Yeah, I think overall as a team we did well. We were able to communicate if we were struggling with issues and help each other get through them quickly, which helps speed up progress.

I think our use of trial or agile in general was fairly good.

We planned out all the stories pretty much straight away.

I think we have to add a few in here and there. But overall having that plan gave us a really good, you know, this is what we're getting done.

And then. Yeah, so we managed to do that really well. Passing them out. Yeah, I agree with what everyone said. Thank. Using the whiteboard to draw like visually what you think how a design should look like, like the menus and then going on to like the different menus and what things should do and stuff like that.

I think that was helpful. Yeah, that's pretty much it. I'll pass it to Tudor. What didn't go so well for Tim was having the issue we had with Maliek's database.

We basically lost all day of working for Malik. Ryan was a few hours and helped him. So we are short like two people for a few hours. I feel like touch with his made us kind of bug for that day at least.

We didn't finish as much as we wanted. Testing was also a problem as well. We didn't plan for testing. We wanted to do it all at the end. So that was an issue that we didn't account for to begin with.

But I'll pass it on to some. Yeah, I think, yeah, testing was. And also we had to do a lot of work in a short period of time.

Maybe if this project was done two weeks long ago could have made it a lot better.

Yeah, I still try. Yeah, I think obviously testing was the biggest thing.

I think everyone's kind of inexperienced with. It didn't help. I, you know, I've never done testing really before.

Never used Mockito. So doing it yesterday and I feel like what you said to me and Malik lost a good few hours on Tuesday and that then kind of set us back yesterday when we got to testing because we couldn't start it quite as quickly as soon as we wanted to.

So we had to fix a few issues and code and everything else.

But yeah, I think, unfortunately, I don't know if there's anything we really could have done differently in that just because of obviously the time scale we're on.

But obviously, you know, we know that going forward and I think planning out the structure of our code and how, you know, we implement it earlier on, could it be beneficial?

So, yeah, I think mainly the database issues, the testing and just the short timescale with the big factors.

Yeah. So like Tudor and Ryan mentioned, database slowed me down.

I couldn't really work on as much data tickets as I wanted to most.

Some of the. I think the MVP got done by the minimum. Got done by like Wednesday, Wednesday morning. So I couldn't really get on to do much tickets though.

The tickets that I wanted to do, I couldn't do because they were already done and they were done during the time of the database issue that I was having.

So. Yeah, and then again with testing, didn't really know how to use Mockito.

Not really done testing before. So yeah, kind of slowed us down in that area and that's pretty much it.

So things I would do differently would be the planning for better planning, the code for testing, for example, having arguments and return statements in our functions.

We didn't know we needed them to begin with. So going forward, I think that would really help us with the testing we could do.

We won't get stuck on using Mockito. We could use the normal JUnit testing that we were shown in class.

Anything else? I don't know. Yeah, just like Theodora said, maybe planning how we implemented our program could have helped us testing going forward.

Yeah, I'll definitely run it so that it can be testable.

Yeah. So planning out, like you say, the code, make sure that we're actually implementing in a way that can be tested.

I think some of the tasks we. The user stories we made could have probably been broken down further, but I think overall we still did well with that, at least for me.

I think I needed to commit more code rather than doing a giant chunk then committing it at the end.

So I think I could do a little bit better with git.

But yeah, overall I think that's everything for me.

So I think for like, as what's been mentioned previously, I think for testing we could have maybe on Monday had a look at like what tests we would have been using or like how, like what functions, like if it would be like.

Because we didn't really know, like for static voids, you.

You can't test those, and you would need, like, a return.

So I think something like that would have helped, because obviously you can't really test with Void functions, so.

So there's really not much to test there other than the ones that can return something for the functions, but, yeah, that's pretty much it.

Anything else to add? No. No. Okay, we'll mark that as the end of the retrospective, then.