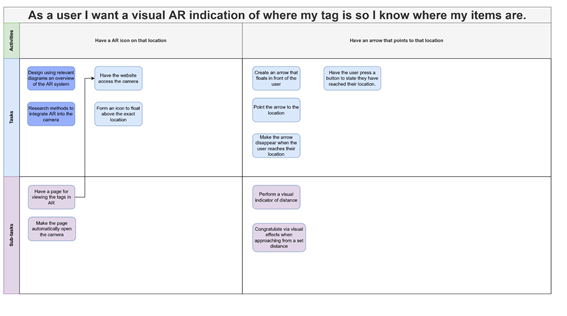
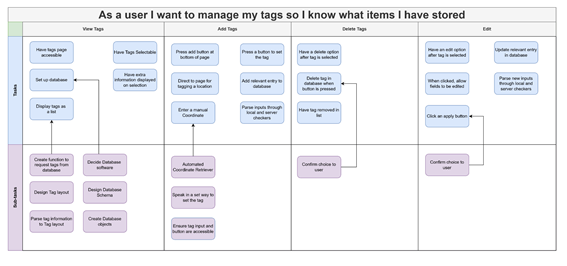
**Changing User Stories after sponsor meeting 02/11/2022**

As an initial set of user stories, we opted to have an arrow point to a location in AR:



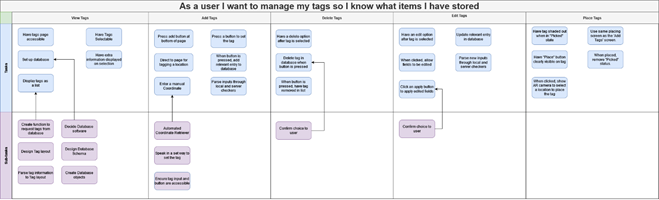
This is with having a tag at a location. In addition to this, we determined the method of managing these tags as a list of them in another page:



Where, we determined ‘edit’ as the only method to change the tags’ location. Moreso, we failed to include the idea of ‘Picking up’ and ‘Placing down’ tags. Thus, if the user were to reach their tag and pick up that object, there would be no explicit status of ‘Picked up’, and instead would be up to implementation, maybe deleting the tag outright.

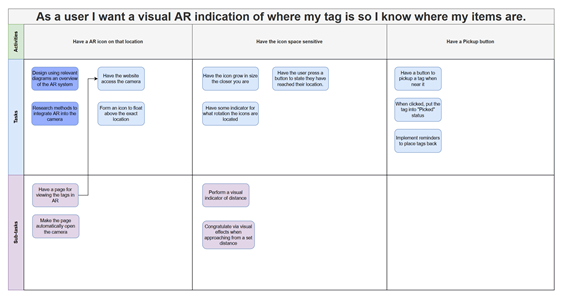
This also removes the idea of Placing down.

After the meeting with our sponsor, the idea of placing and picking was brought up, so we decided to overhaul some of our user stories to reflect this:



In the page where you view tags, we opted to have a new status for the tags – them being ‘Picked up’. In this state, the tag would be shaded out, and the only option you’d have would be to place the tag back down.

The edit tags are still here, but it will be relegated to more rare situations.

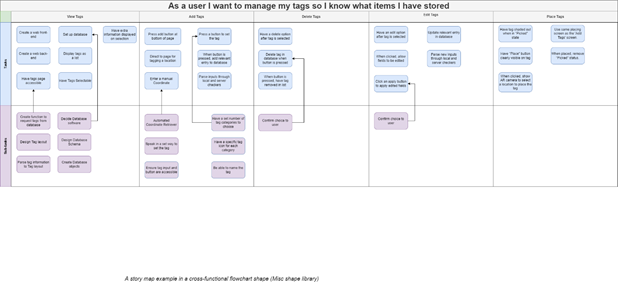


In AR mode, we have also added the user stories for having a pickup button. This would be activated when the user is close to a tag, and thus when they pick up the item, the system can reflect that with the ‘Picked up’ state.

**Changing User Stories after meeting 09/11/2022**

We had a meeting deciding what our first sprint should focus on, and we decided on a short 1 week sprint to research the most pivotal AR implementation, and also create the foundation of the website and database.

With this, we found that we had no explicit user story to create the website, so we edited the story map to reflect this change:

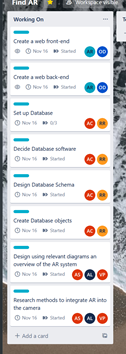


In view tags, two new user stories were added to reflect our process.

**Sprint 1 (09/11/2022 -> 16/11/2022)**

In this sprint, we decided to first create the foundation of our project. Thus, it was made to be shorter than normal.

The user stories assigned were:

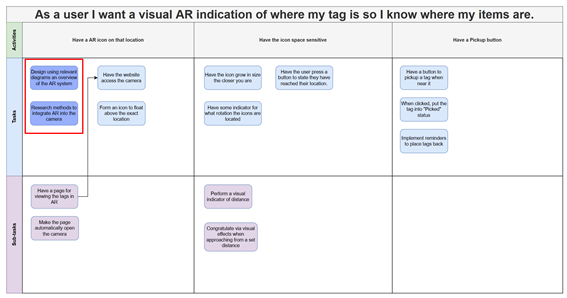


And we split the team into 3 subteams, with one working on designing/researching how the use AR in websites. Another team was set on setting up a website that everyone can develop on in the future, and the last was assigned the task of designing some extensible database that reflects our current goal of a viewable list of tags and a user login system.

Most of the user stories were in the ‘View Tags’ Epic:



And the ‘Have an AR icon on that location” Epic:



We have also set up the sub-teams to reflect the skillset of everyone. The website sub-team includes Alfie, who has moderate experience in web technologies, and Oliver, whose skillset leans more on data interpretation.

Our AR team is the largest, as no one in our team has any substantial experience in AR. This is the primary reason we have dedicated a week to just research.

The database team consists of Alan and Rodion, and both of their experience stems from first year modules. No one in our team has more extensive experience in database design, so there was no calculated choice here.

**Sub-Team Website Creation user stories:**

For this subteam, with Alfie and Olly, we were unable to progress much due to being unable to access IBM services. During this we with the team lead set up a meeting with our sponsor and got everyone to sign up to IBM services as the pre-requisite to access the services.

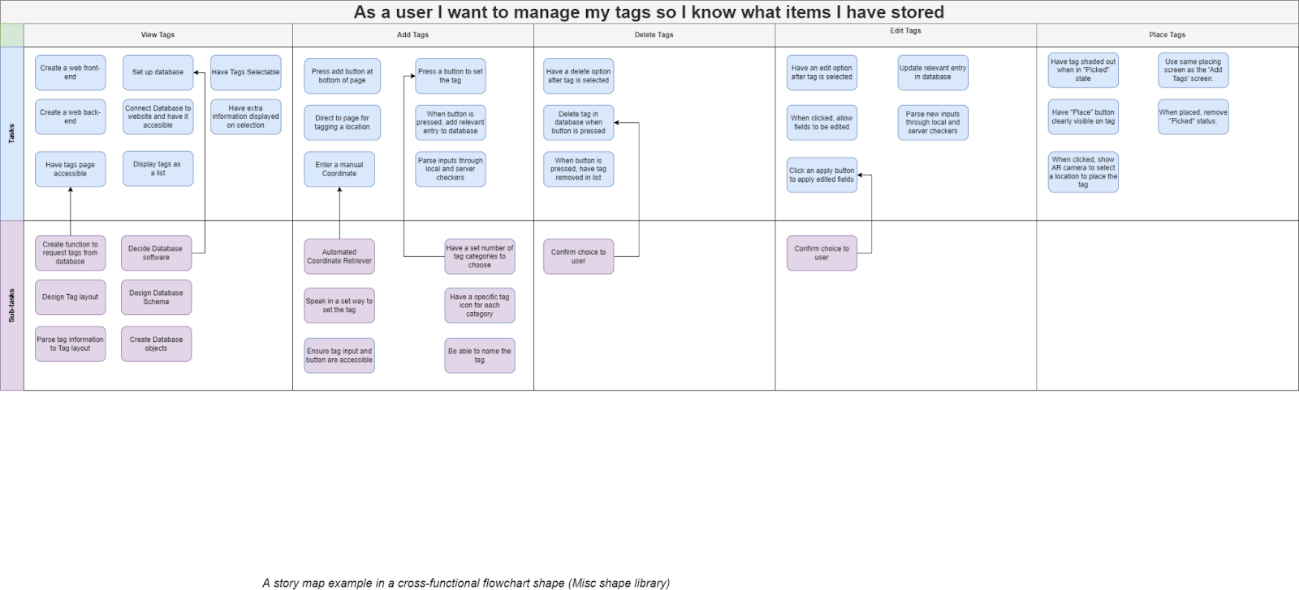
Thus, these user stories have been moved to the next sprint to do.

**Sprint 1 Retrospective/ Sprint 2 (17/11/2022 -> 1/12/2022)**

We held a retrospective on Thursday 17/11/2022 to go over what everyone achieved, what was difficult, and what we need to change going forward.

The “Set up a database” sub-team with Rodion and Aidan managed to set up and create a schema, achieving all of their user stories. They done this using IBM’s services which we were allowed to use.

However, with this, we found that we needed to change our ‘View Tags’ Epic to accommodate for the next step, integrating and connecting that database to a web-server:



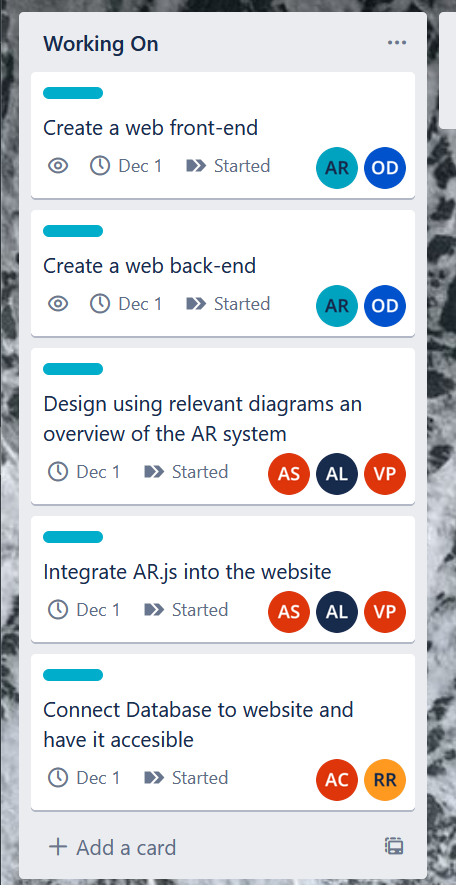
We assigned “Connect database to website and have it accessible” to the story map, as it makes more sense in the context of this retrospective.

The AR side of the project was also researched, and the sub-team of Alan, Vishal and Alex found that a lot of the work has already been done in AR.js. However, as there was no server to actively test it on, it was difficult to create concrete diagrams to outline the system, so the sub-team could only finish half the sprint user stories.

Through this we discussed to include a new user-story to ‘integrate AR.JS’ into the soon-to-be webserver, as a more specific variation of the problem:

For the website creation, none of the user stories were able to be completed, but this was due to areas outside the project, so nothing could be done to rectify it.

With this, we assigned new user-stories to be completed:



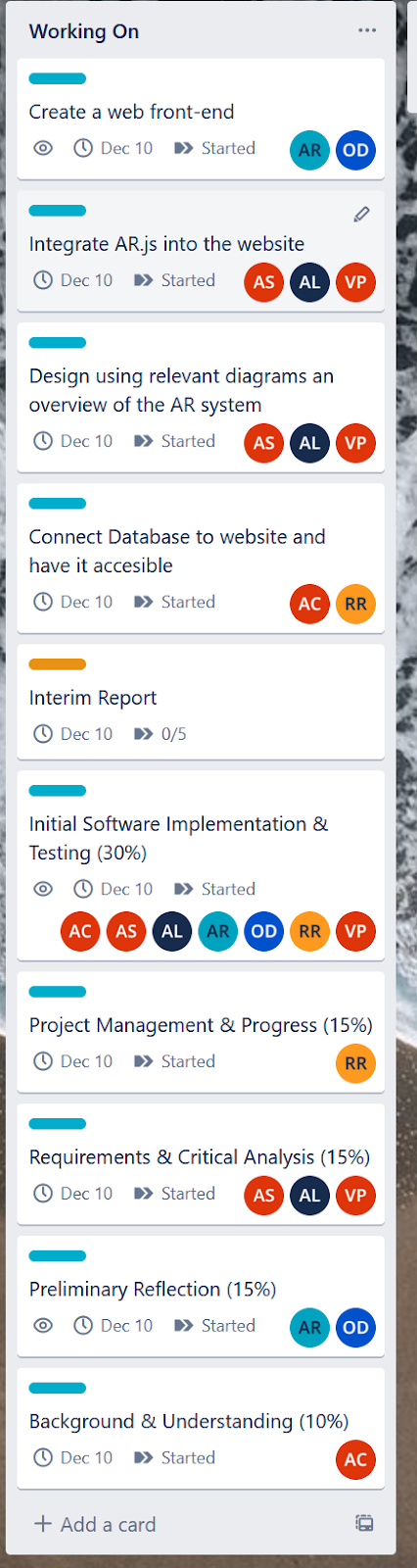
Sprint 3 (01/12/2022-10/12/2022) (Sprint 1 Retrospective)

We had a retrospective on 01/12/2022 to discuss Sprint 2.

In the Sprint 2 retrospective we discussed the difficulties in server creation. Due to delays in access to IBM’s server, half of the sprint was in a deadlock, and eventually the Website Creation SubTeam found a Serverless implementation of Node.JS in IBM Cloud that could function as a stop-gap before getting full access to IBMs resources.

However, Severless has its one challenges, the most prevalent being the requirement of local copies to properly develop, due to the CI deployment taking more than 3 minutes. In any case, this causes a week delay in being able to start, so we decided to move over the remaining user stories to a week +2 day sprint.

We also discussed the Interim Report, and thus assigned parts of it to everyone to be done in the same sprint, from which we will send the draft to our supervisor for advice:

The SubTeams will remain the same, and the only completed user story was the “Create a web back-end”, which was the Node.JS server creation using a Serverless implementation on IBM Cloud.

We also discussed difficulties for the Database Creation Sub-Team, as IBM only allowed proprietary Database solutions to be created on our limited accounts, which even after hours of attempts bore fruitless results in integrating them in Node.JS, or even connecting to them in a traditional way.

Thus, they had to move to another hosting solution for MongoDB, which is a more traditional, and open source implementation of MYSQL which doesn’t require absurd drivers to access.

So, they stated they are most likely able to implement a proof of concept connection to this MongoDB database in the Node.JS server quickly.

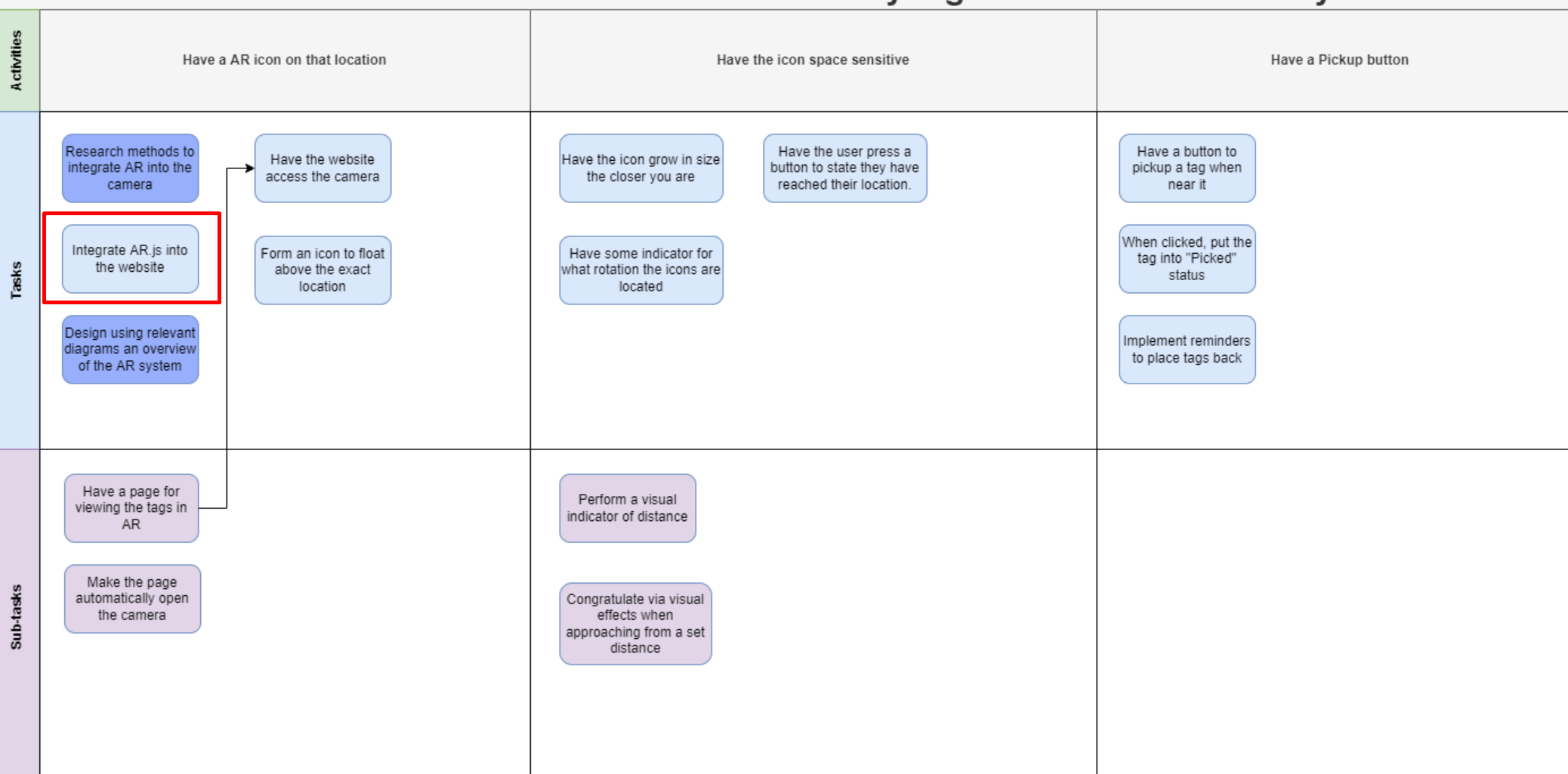
The AR Sub-Team has had more trouble, as the week delay, coinciding with the difficulty in setting up a local repo Docker container to develop in, made progress difficult. This is also ignoring the debugging nightmare of not having an easy way to test the website on your phone, because it is a local server all the way until deployment via a merge to master.

However, their task is to just integrate AR.JS into the front-end, so it should be plausible to achieve, which we all agreed on.

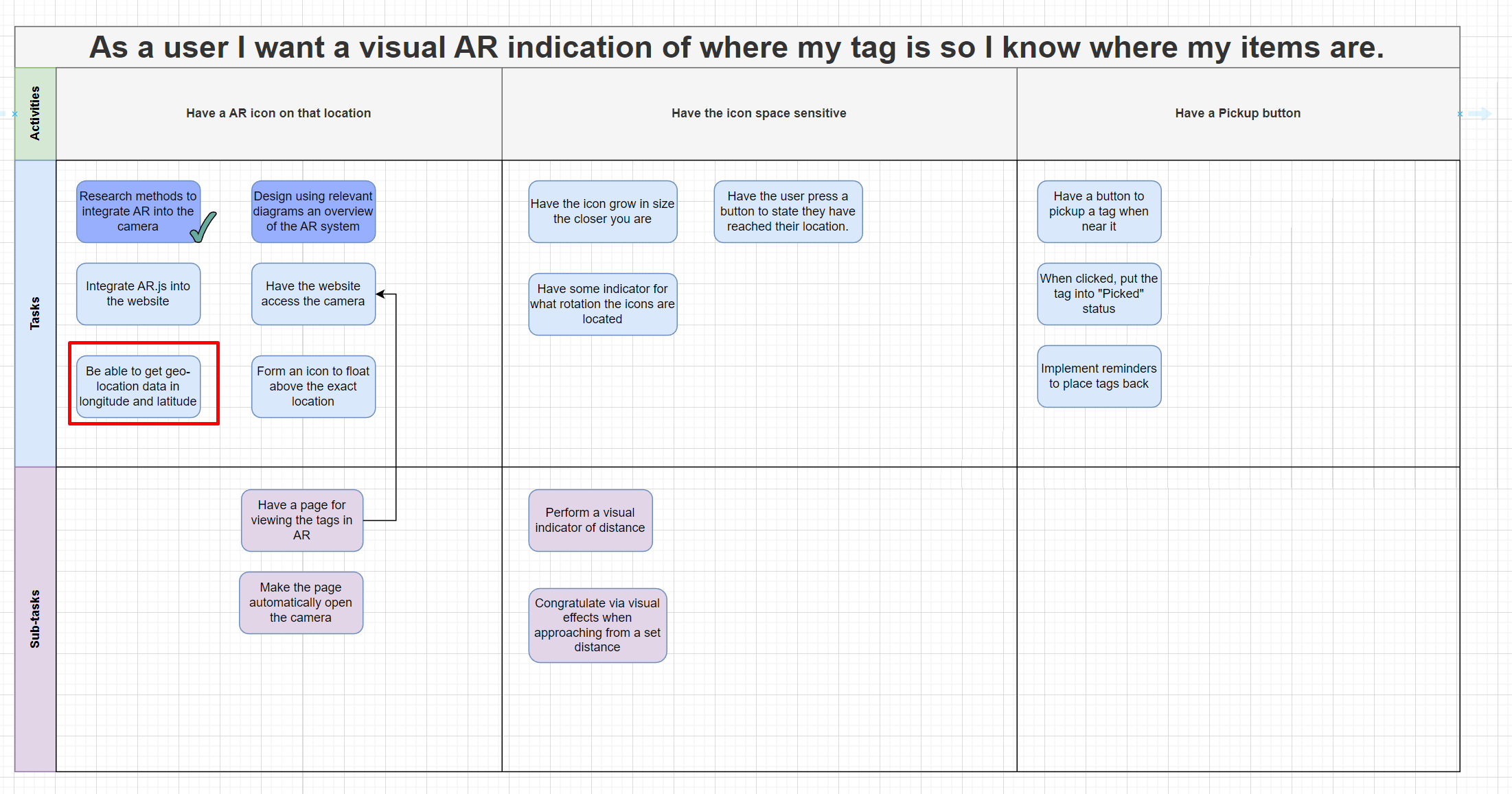
**Sprint 4 (31/01/2023-07/02/2023)**

Here we discussed the general method of development. This involved setting up everyone's local docker instance to develop on the backend and front end. After this, we discussed what user stories to create.

From this, we found that certain user stories were missing. One, was that the AR icon was missing a follow-up from integrating AR.js into the website

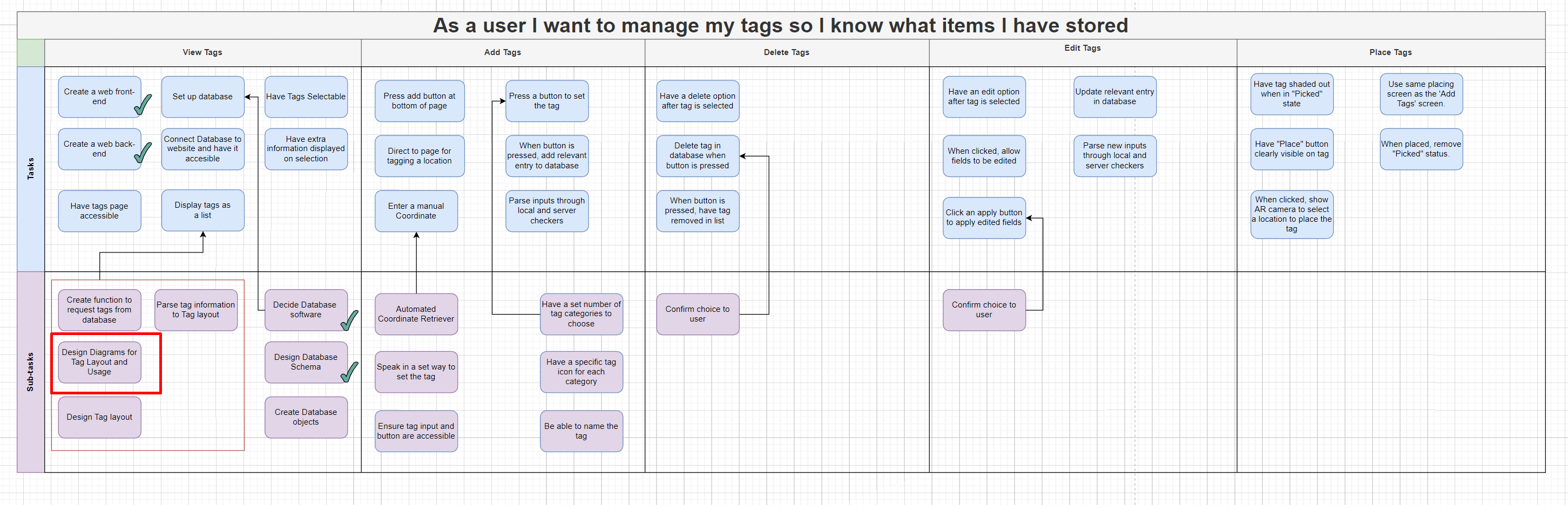


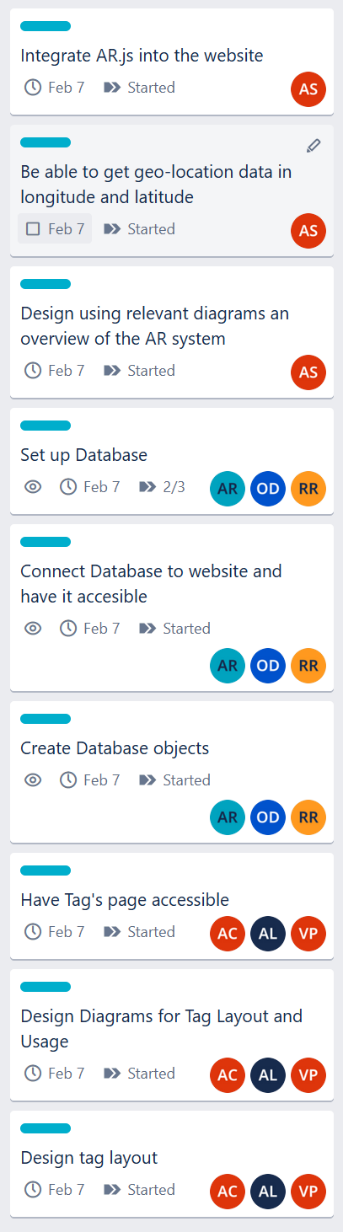
Moreso, “Integrate AR.js into the website” is a general integration, but to then use the AR for our purpose we then need to have the facility to get geo-location from the user. Otherwise, this task would be broadly involved in that single user story, and from the context of the other user stories it was thus too large.



This made it so that integrating AR.js involved just adding the AR.js template code and not also needing to add location tracking in the same user story.

After this we also decided to add an explicit design user story for the tag layout. On the tag layout page that will show the tags the user has, we have yet to properly decided how it will function, so it needs to be designed first before implementing a template on the front-end:

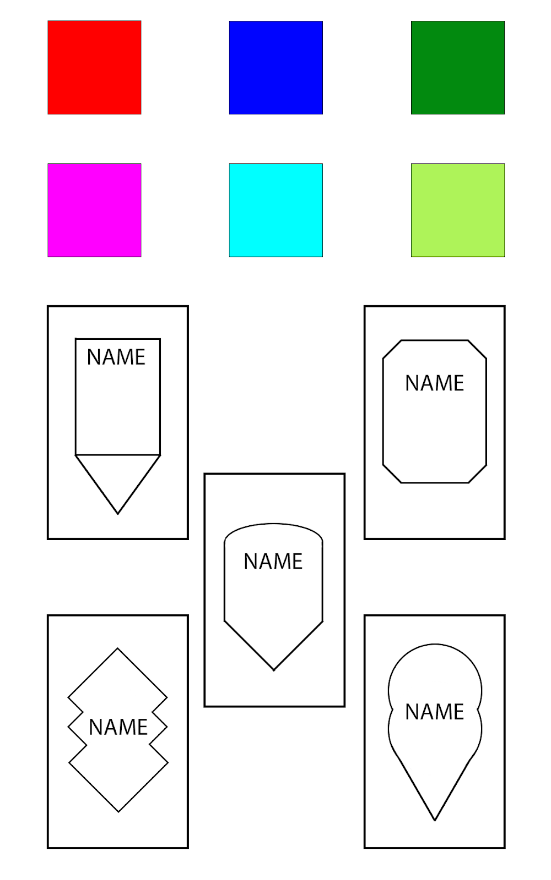
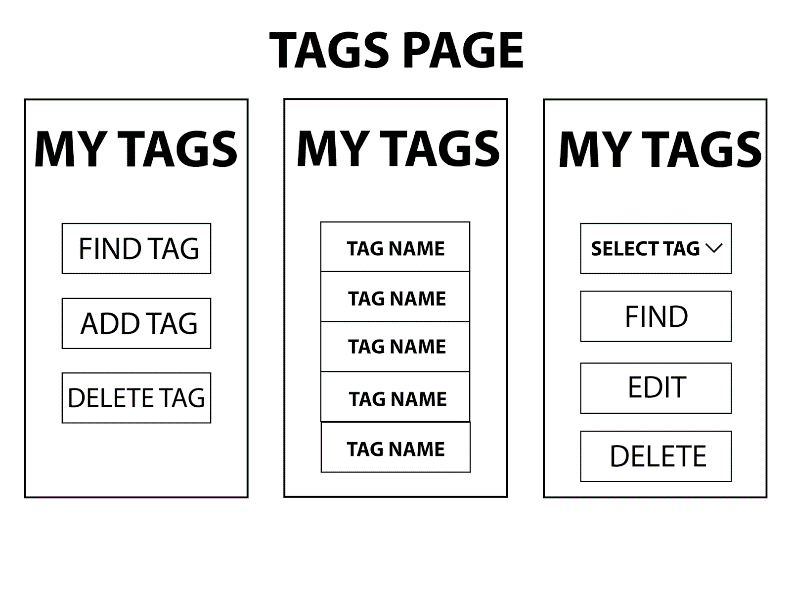


On that note, assigning user stories went into three teams. Alfie, Rodion, and Oliver were tasked with migrating the existing MySQL schema to a NoSQL implementation due to technical constraints, these being the user stories: “Connect Database to website and have it accessible”, “Create Database Objects” and “Set up database”. We assigned Alfie and Oliver as they were previously working on the server, and so haven’t had any experience with the database software, but still have backend knowledge to help with eventual int.

After this, follow the user story we needed to create the tags page which will display the users’ tags that they have saved. This tags page needs to be designed with how it will function, and how the tags will look and behave, and we also need a route to get to this page that isn’t the root directory. So, we assigned Aidan, Vishal and Alex. Aidan and Vishal are both doing Human Computer Interactions, so we felt that they would be best tasked to designing an interface. All of them need to gather experience working with Svelte. The user stories assigned are “Have Tag’s page accessible”, “Design Diagrams for Tag Layout and Usage” and “Design Tag layout”.

The last assignment was for the AR theme, and generally Alan has been working on integrating the AR.js library on and off during the holiday. He stated that it would be finished quite easily, and so didn’t need anyone else to help him and waste time. He aims to integrate the AR.js code so that you can view an item in AR that spawns on a specific location and keeps track of your location relative to it. This is all done via AR.js and so isn’t a major hurdle. The user stories assigned are, “Integrate AR.js into the website”, “Be able to get geo-location data in longitude and latitude” and “Design using relevant diagrams an overview of the AR system”.

**Design Front End Team**Aidan tackled the user stories focused on the design of the tag page and the tags itself. He created mockups that can be used as a basis for the design of the front end as well as a focal point for design discussions. These designs are not final and are meant to be used to explore which direction we want to take with the UI.



\*Note the 3 designs of the tags page are separate concepts but they may be interwoven to produce a final product. For example design 1 could pop up over design 2 after the user has selected a tag from the list.

These designs were built upon research into what makes a good UI and specifically what makes a good UI for the elderly/disabled. The tldr from this research was to keep things simple as possible as to not overload the user. The main keyword is comfortable, we want our user to feel relaxed while achieving their end goal. Taking this into account, the design uses simple shapes with their actions/information clearly indicated.

The full report of this research can be found in the linked document.  
[What Makes A Good UI](https://docs.google.com/document/d/1eMTlwMNlD9AuidL6vLBELJxXlo2CLSeF7vB6LEnorOc/edit?usp=sharing)

Moving forward we hope to refine these designs and then implement them into the webpage. Furthermore the designs are open to improvement including the introduction of icons depicting when each button does or what item each tag refers to.

**AR Front-End Team**

Due the workload for the AR aspect of the program being significantly smaller than expected the AR Team made the decision to split, leaving Alan to do the rest of the AR and Vishal and Alex to focus on the front end design with the other teams.

Most of this sprint was spent actually implementing AR into code, unfortunately due to the lack of a server we were unable to deploy our working prototype which was necessary as we needed to be able to access our application from a phone in order to use the camera and gps facilities that it would provide.

Alan found a workaround to this issue by creating his own private repo on github and using the github pages features, which allows easy deployment of the website meaning we could test our website on your phones.

Using the code from ar.js studio and a placeholder asset Alan deployed the website as a purely html website. Initially it seemed to work with the camera appearing on screen and an asset appearing overlaid on the camera. But there was one glaring bug which was that the asset would “stick” to the camera only displaying the direction of the asset but not the distance from the user.

Attempts at debugging were made but they were all unsuccessful but after some more info it turned out that ar.js is not heavily used anymore and therefore had many bugs, so the code found ar.js studio was replaced with one from ar.js wiki which when deployed to pages provided a working implementation.

After the working implementation was made, more testing was done another problem was found, in which we found the accuracy of the marker was lacking, further testing needs to be done to make sure that it is usable, the main problems seems to be the with the accuracy of GPS data whilst indoors, more research on this problem will be done next sprint. Additionally there is no way for the marker to indicate elevation, further research will again be done to try solve this issue.

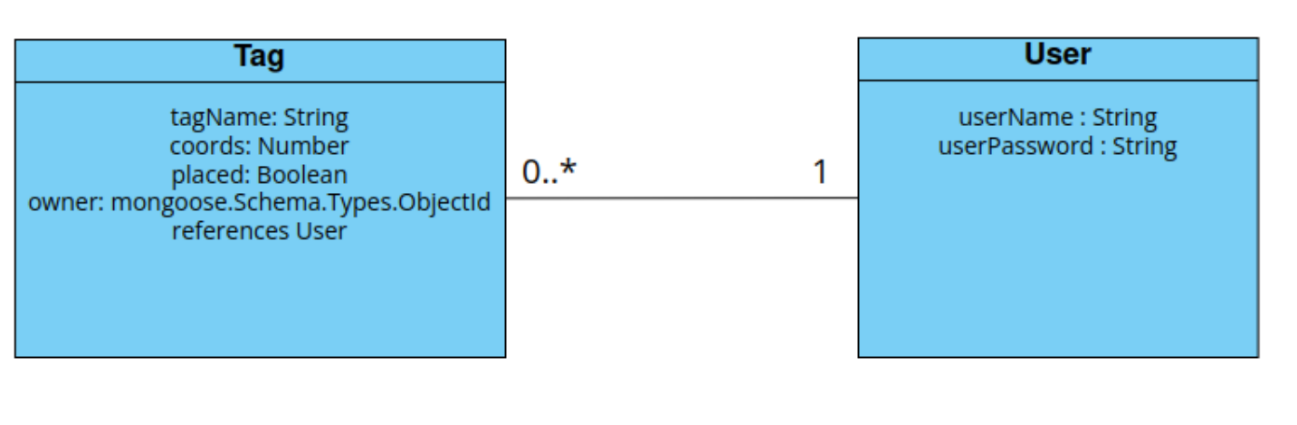
Goals for next sprint:

* Embed the code into the actual server
* Fix elevation and accuracy problems
* Draw diagrams to explain how the process of creating an AR instance should work.
* Do Documentation on the AR system

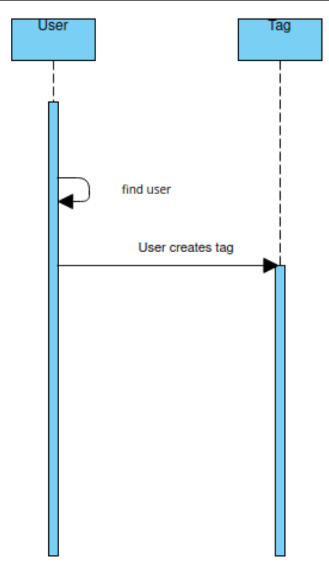
**Database Back-end team**

The goal of this team was to convert the MySQL schema and code into MongoDB, and then implement a connection in the serverless website. We thus assigned Oliver with the task of converting the code into MongoDB, Alfie with implementing this code into the website’s backend, and Rodion with documenting the schema and a useful scenario.

To outline the new schema, Alfie implemented this into node.js to connect to the MongoDB free server so that we could implement these schemas. Rodion then outlined these two objects:



And as a usage:



Therefore, we were able to complete all three user stories:“Connect Database to website and have it accessible”, “Create Database Objects” and “Set up database”. Though the design is a common creation in every software user story, so it is not explicitly stated.

**Sprint 5 (07/02/2023-14/02/2023)** (Sprint 4 Retrospective)

For the retrospective, we went over everyone's contributions.

For AR.js, we found that Alan had to do user stories outside the scope of his assignment. Moreso, all the user stories he had assigned were not completed, because he had to do:

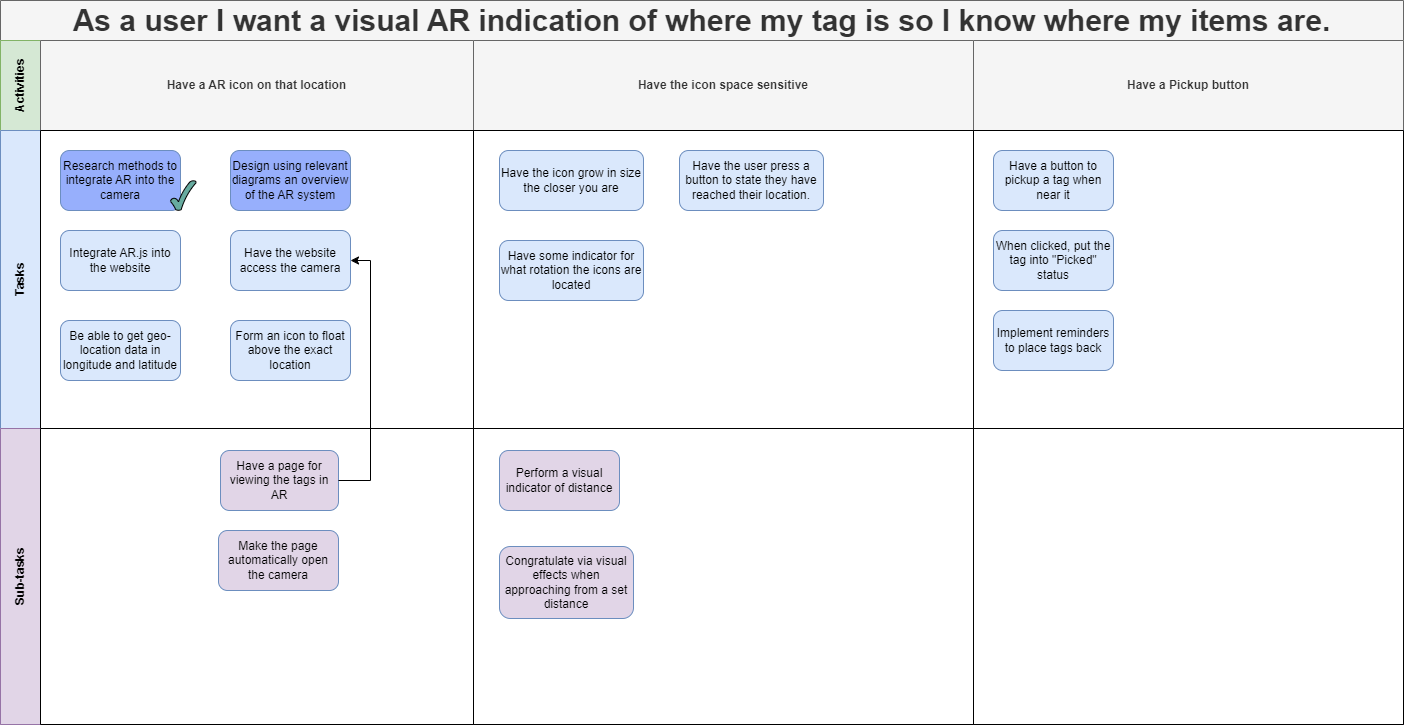
“Have the page access the camera”

“Form an icon to float above the exact location”

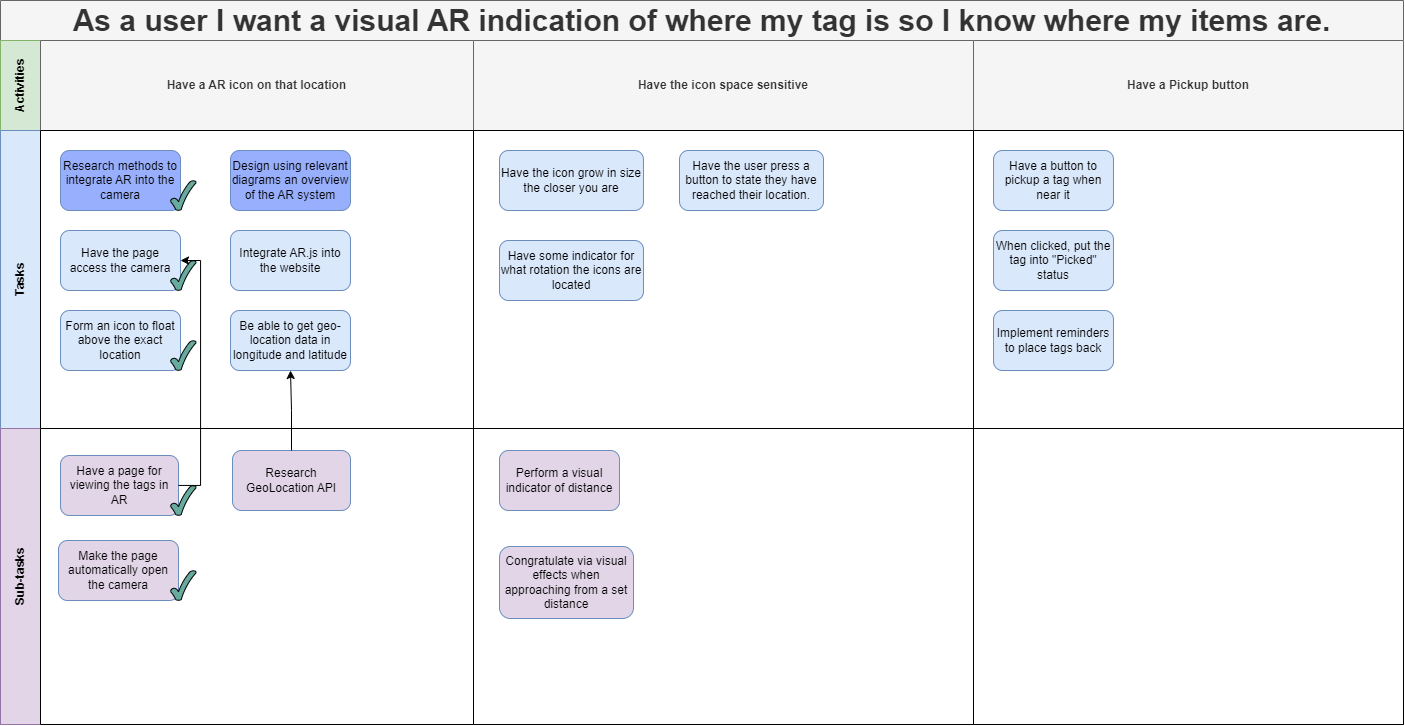
“Have a page for viewing the tags in AR”

“Make the page automatically open the camera”

All of these user stories had to be completed before he was able to integrate AR.js into the website, and so Alfie rehashed the story map from:

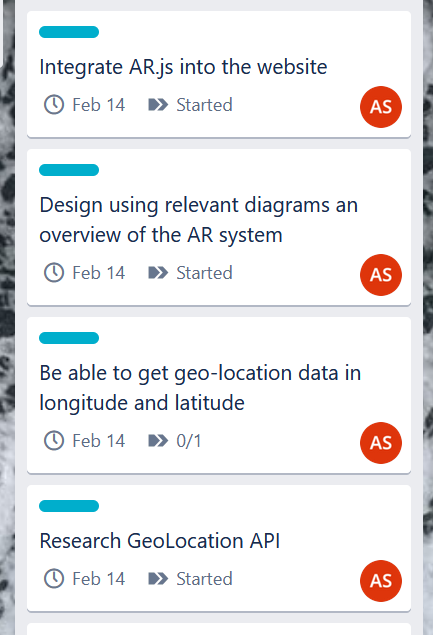


To:



Note here that all those user stories have been placed before Integrate AR.js into the website. We also added a new user story “Research GeoLocation API” for Alan to find a library that get people's locations for the AR.js placing system.

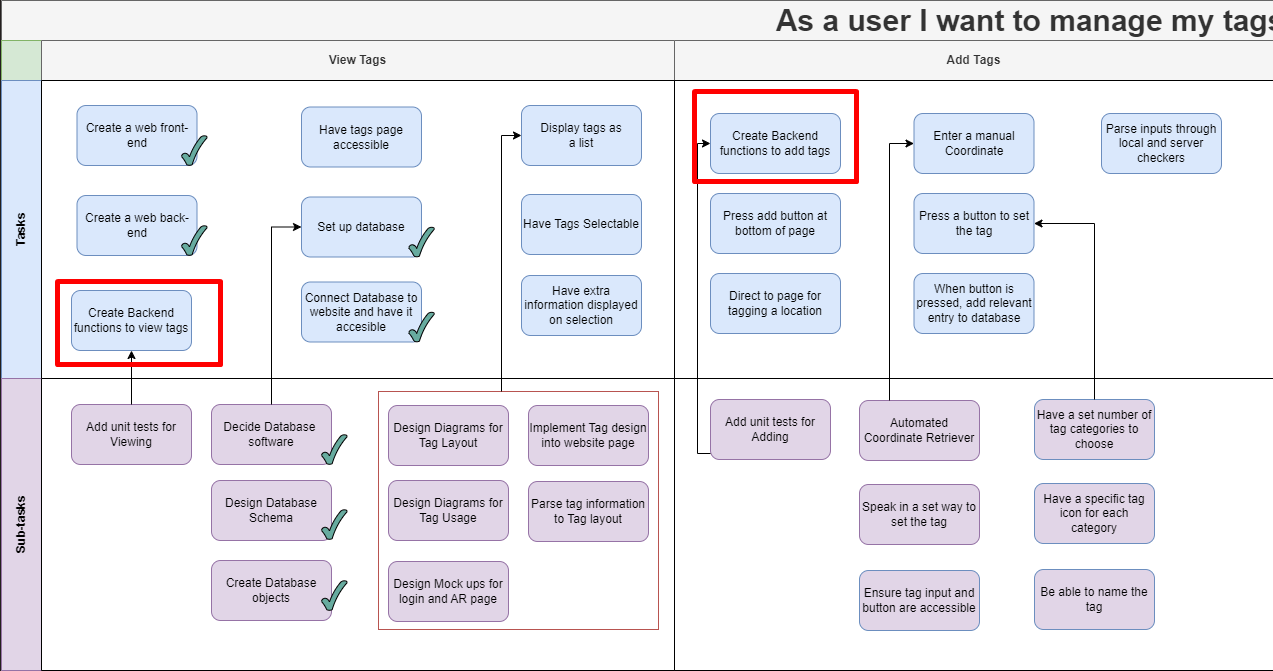
From this, we assigned Alan the following user stories:

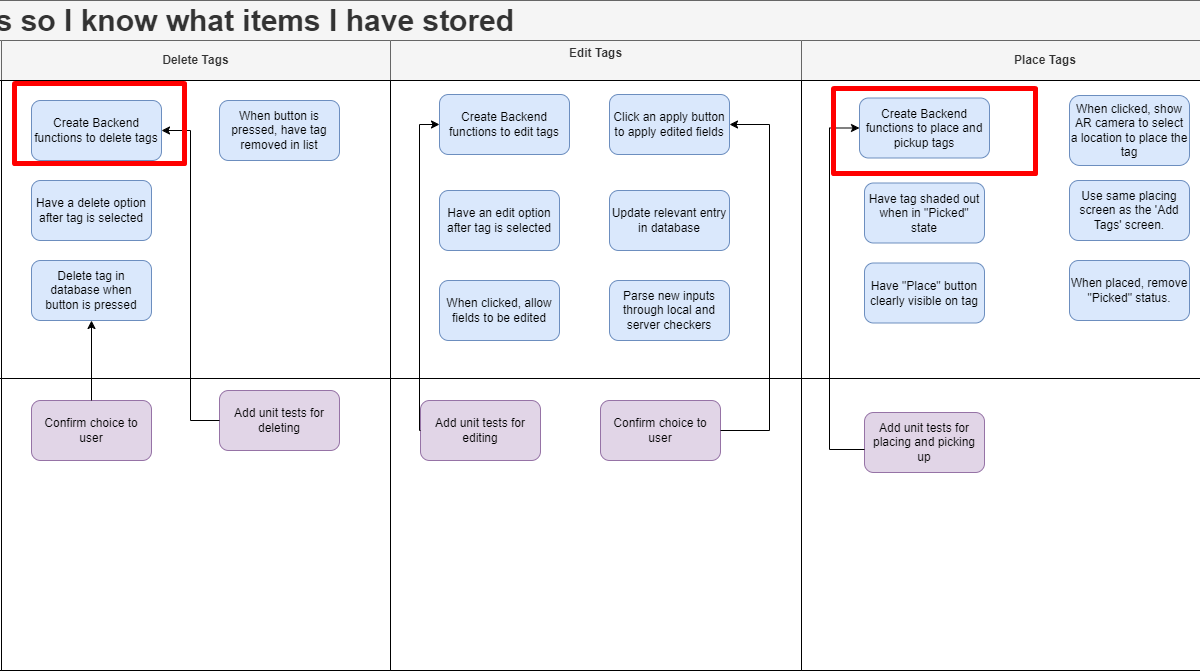


Two of them are from the previous sprint, and one is a sub story of the geolocation story.

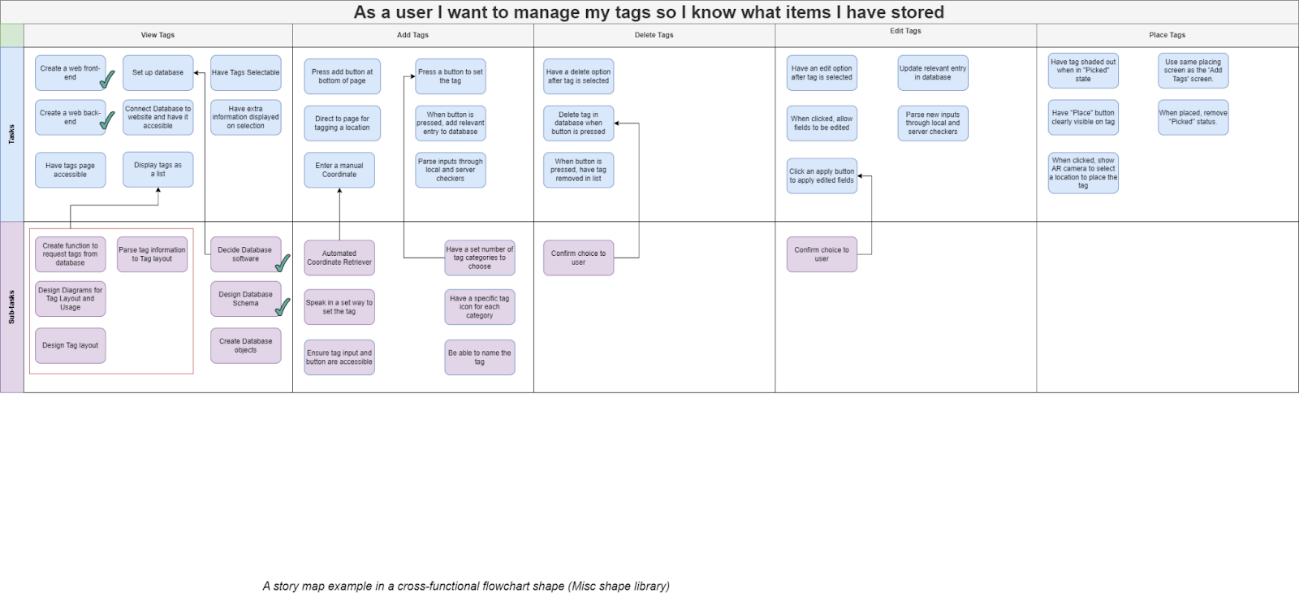
We discussed a focus on the diagrams of how AR.JS will be used when going to the AR page in the website, and integrating AR.js into the website is a trivial exercise of moving the code from github to the svelt front-end.

For the Backend Database team, all of their user stories were completed. However, after this we found a severe lack of explicit user stories to design parts of the backend, such as viewing tags, deleting them, etc. So, Alfie redesigned the story map and conjured these additions:

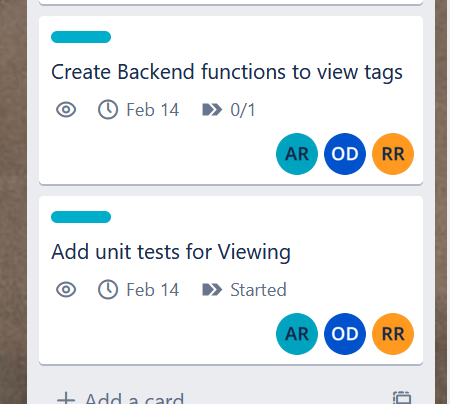




This is from the previous story map:



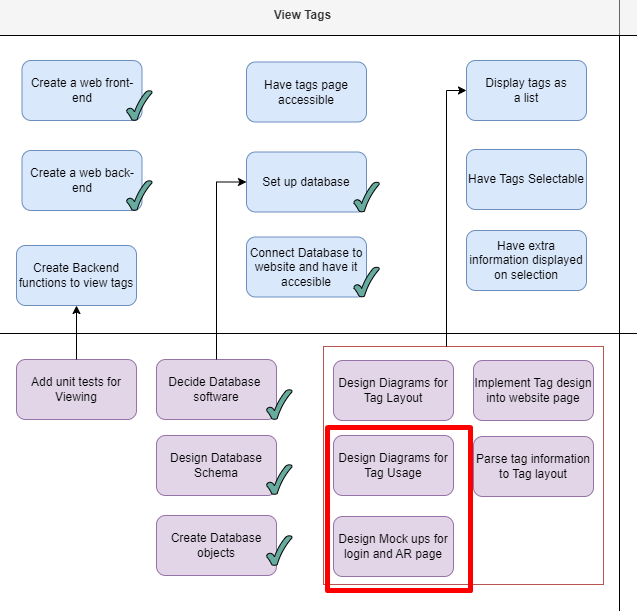
These user stories allow us to explicitly assign tasks to the backend functionality, and of course are required for any functionality needing the database. So, we decided to assign:



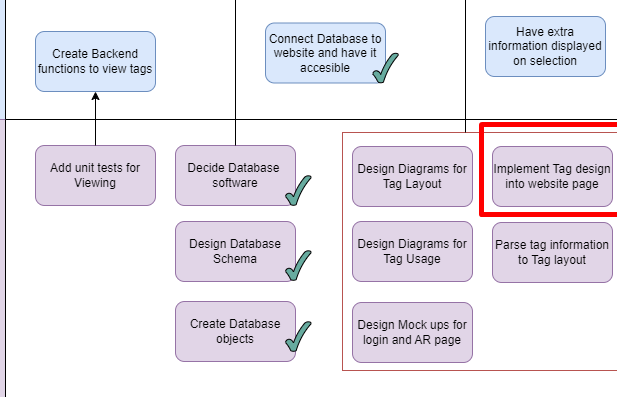
This will implement the functionality of getting tag data, and unit tests for this. This also implicitly includes diagrams for the expected usage of viewing tags. If this is done quickly we will assign more of these backend functions, because they can be done in parallel.

For the Designing and creation of tags, the team of Aidan, Alex and Vishal were able to come up with designs of how the tag will look in AR, and some prototypes of the tag page. Due to it being a new framework, the team needed more time to properly implement these designs into Svelte.

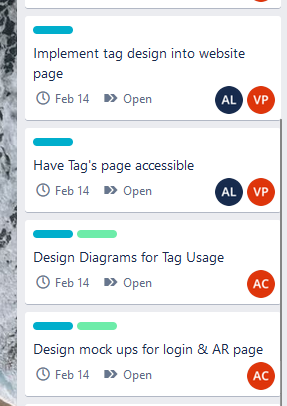
They also decided to split the team into two, with Aidan focusing on the design of the pages due to his experience with the HCI module, and we added a new user story to make explicit the design of the tag usage, and the mock ups for the login and AR page and how they will flow:



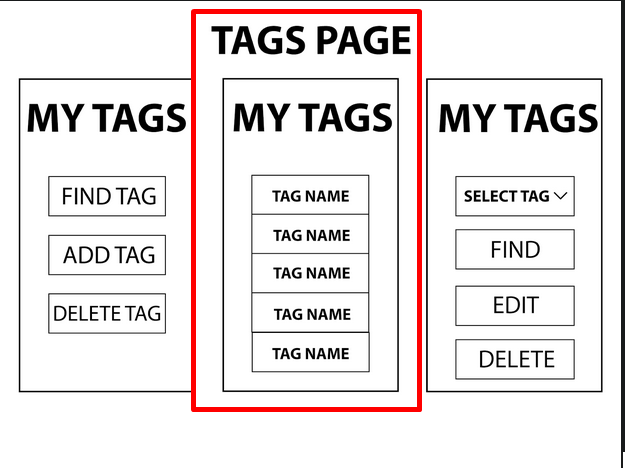
Alex and Vishal wished to focus on the Svelte page, as this can be done in more parallel work. We added a new user story to make it explicit for implementing the tag design we chose into the website:



Thus the assignments were:



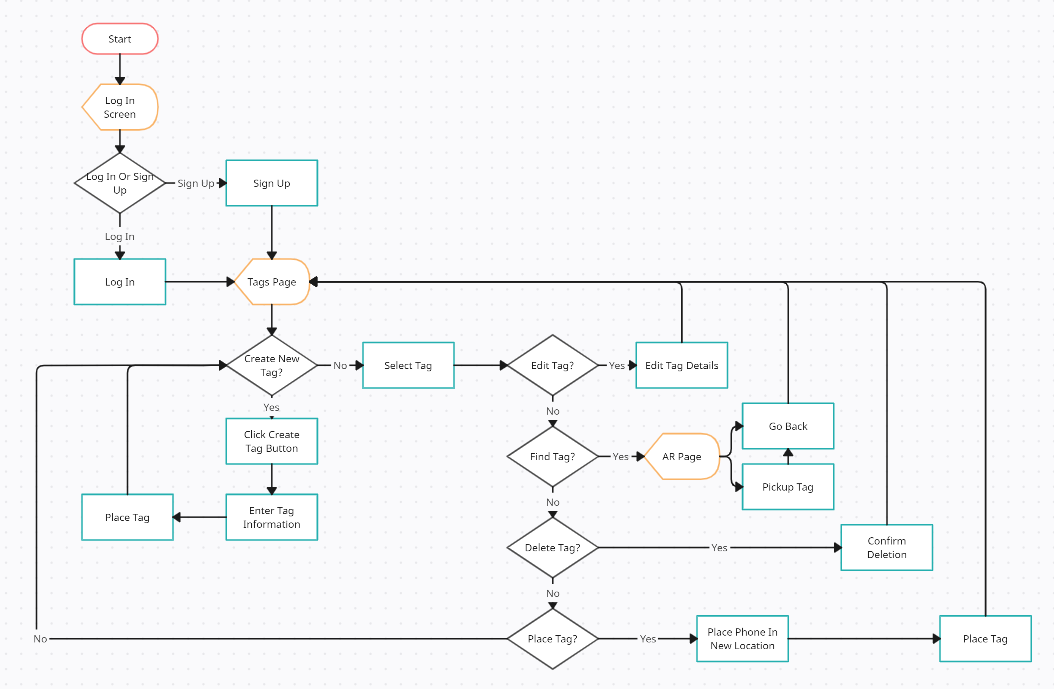
And, for the page design we chose, it was the list one:



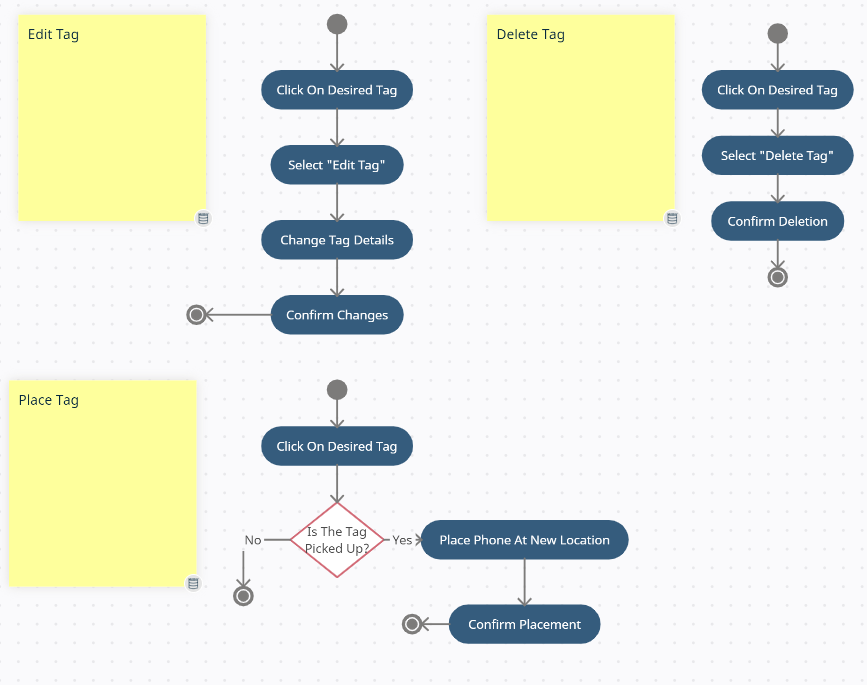
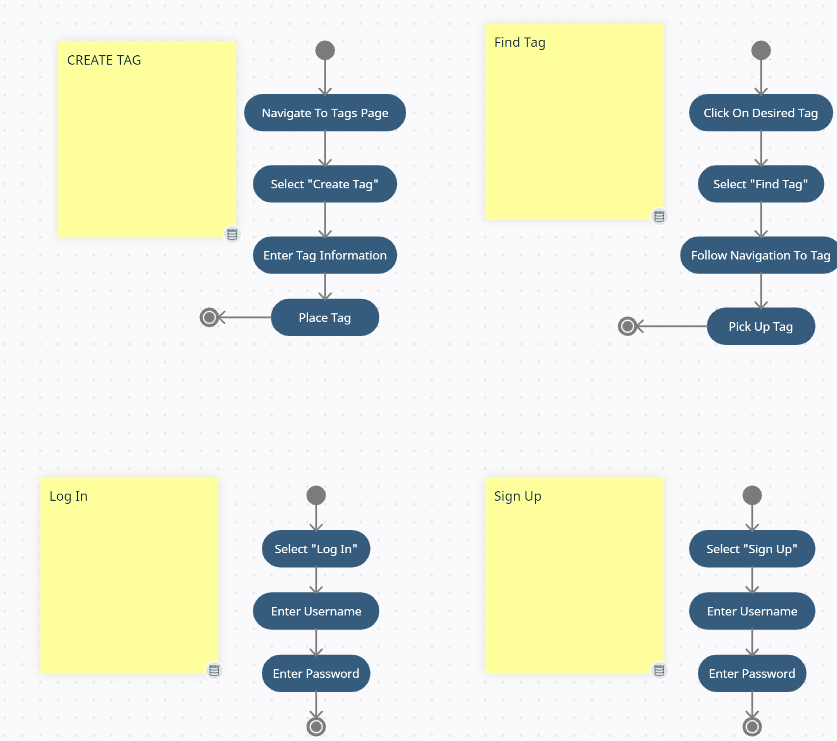
So, this team will work on the design, and the implementation of the tag design, which is a long form tag in a list. The implementation has some liberties, as specifics in how the tag will look is up to Alex and Vishal. However, Aidan will also communicate on how he expects the workflow of this page to look like, so that they are on the same page with the implementation.

**Design Front End Team**

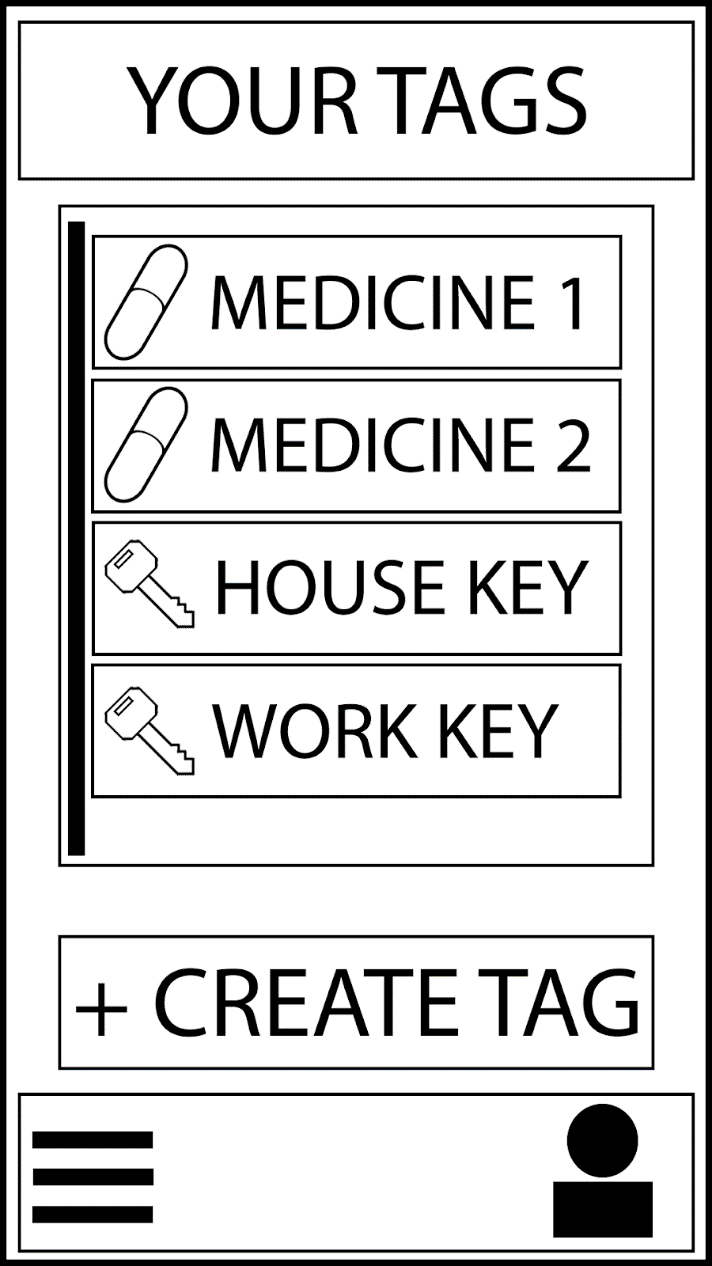
After Aidan presented the mockups to the team a discussion was had about the overall flow of the web app. Based on these discussions Aidan drew up an overall flow diagram to depict how the user will use our product.



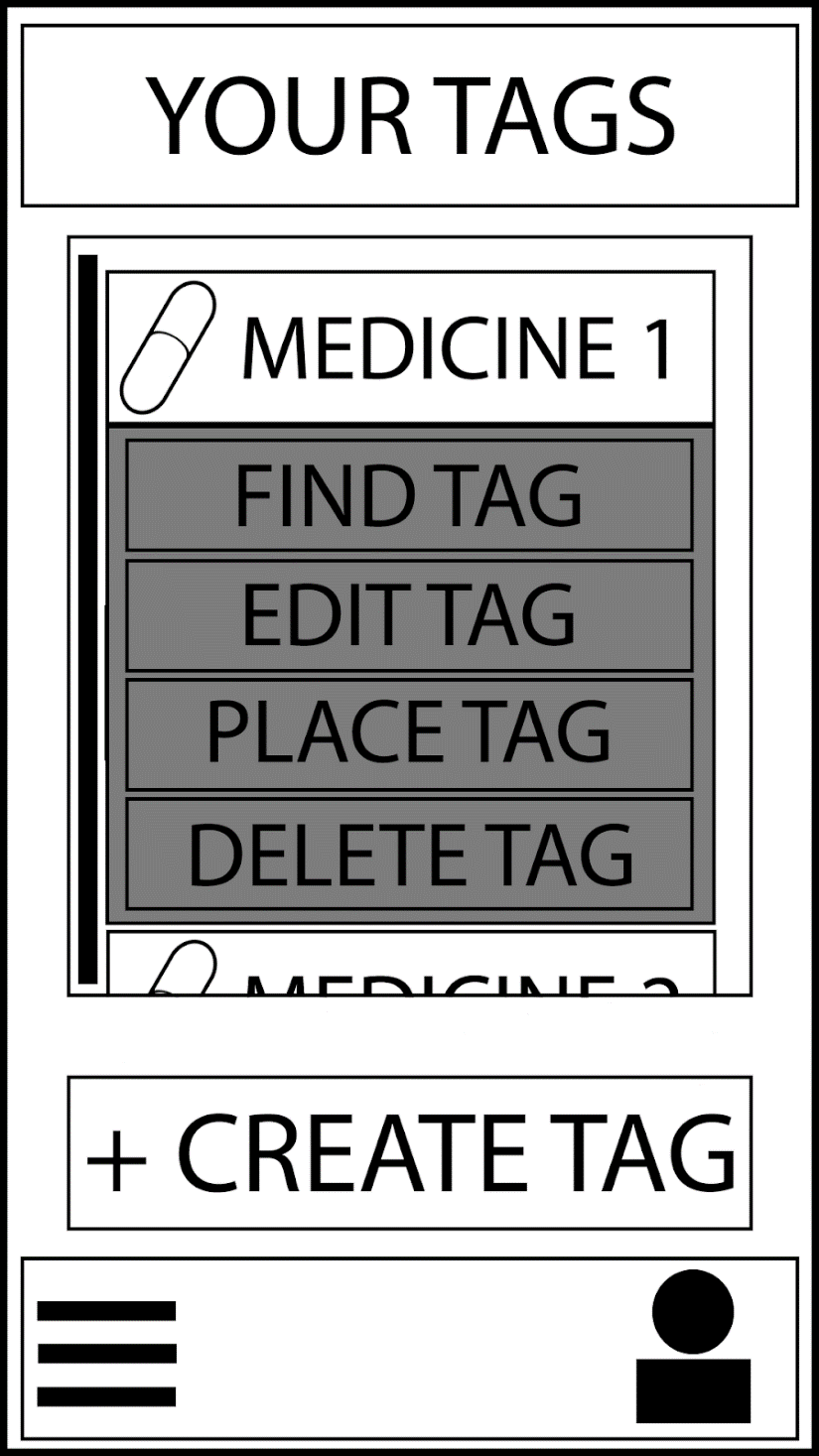
Building upon this diagram Aidan also developed a series of activity diagrams showing the flow of specific actions the user will carry out. They involve simple steps which will aid the user in completing their end goal.



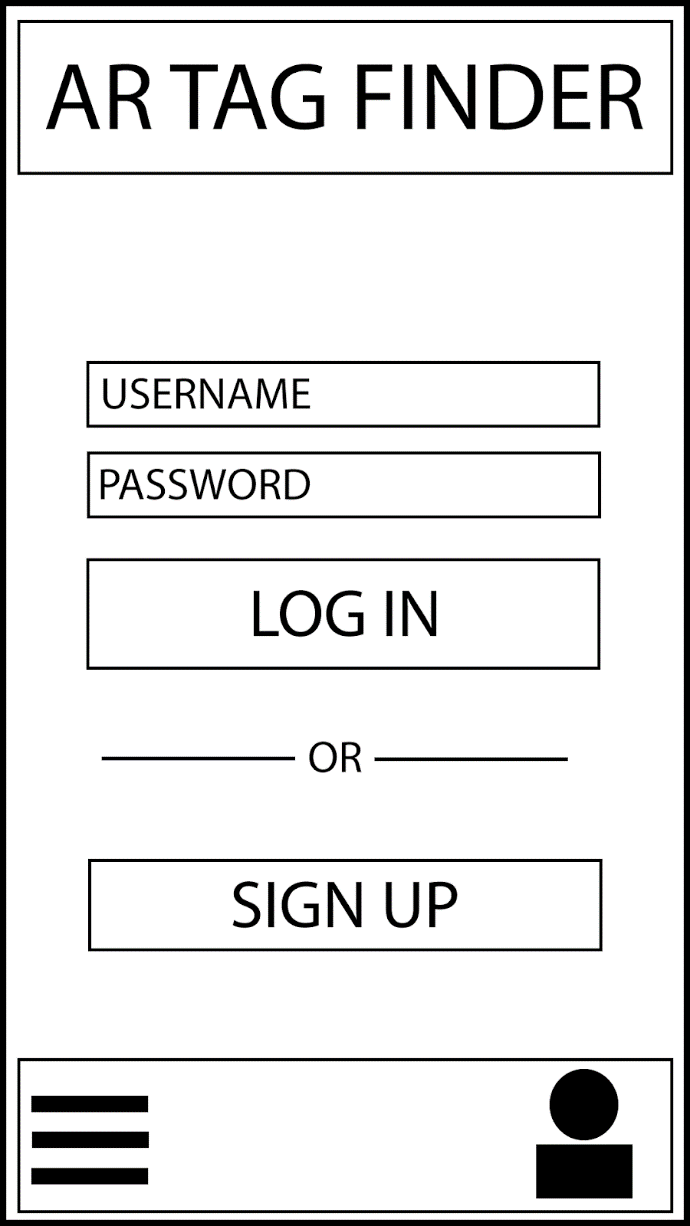
During this sprint Aidan also took the feedback about the initial mock ups and refined the design to more reflect what we want our end product to look like.

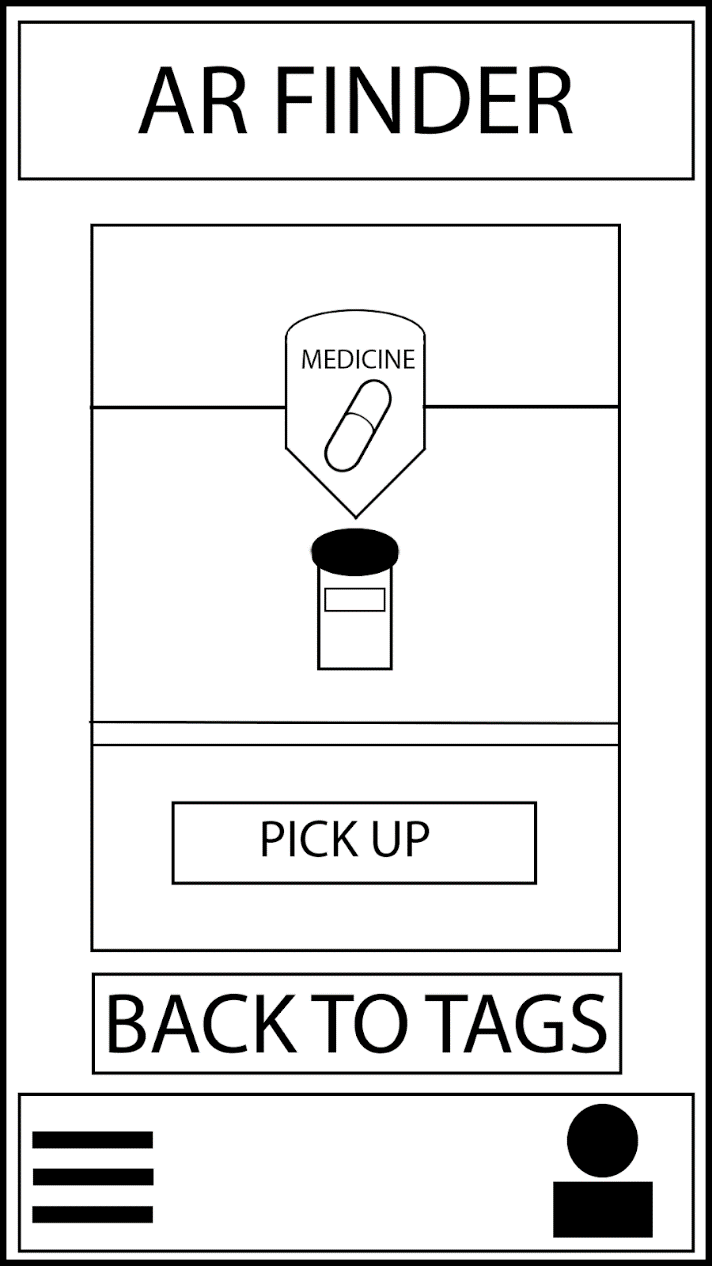
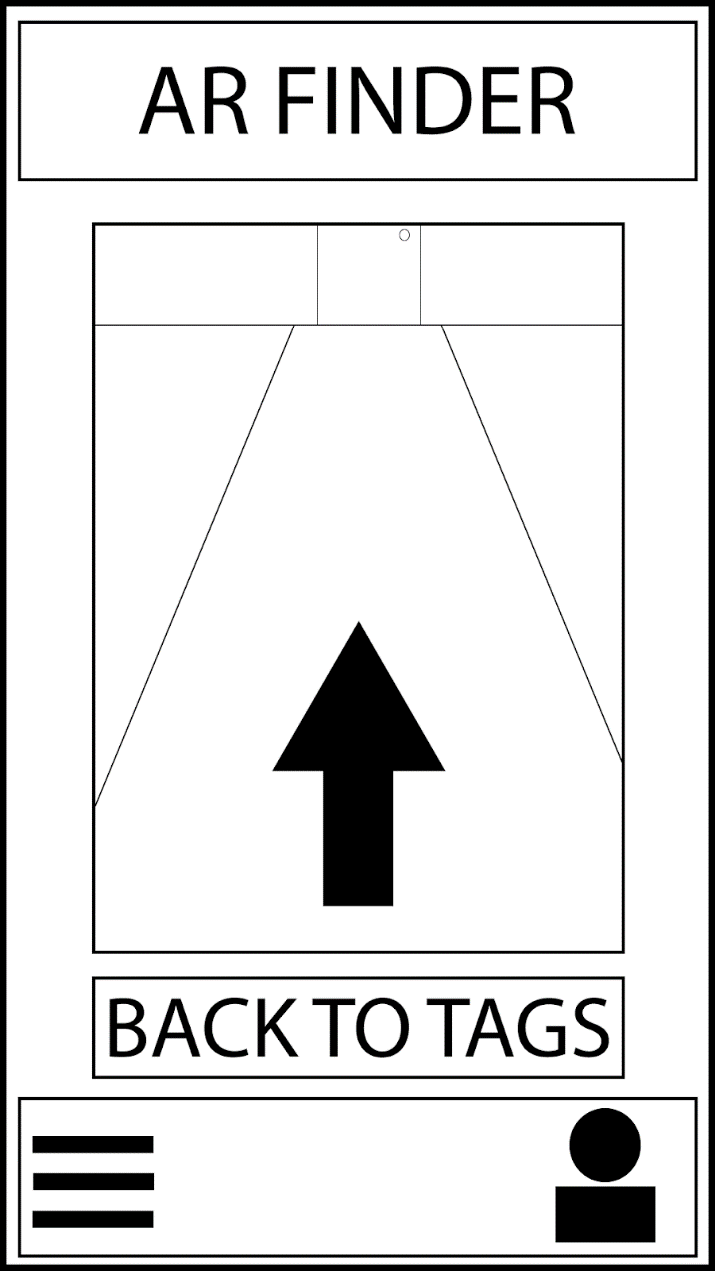


The new design takes the core element from the old design (the list of tags) and builds upon that to increase usability. This includes the clear  “create tag” button, the navigation bar on the bottom and example tags listed in the scroll list.  
In this design when you press a tag it will expand to show the operations you can do on the tag, such as find, edit, place or delete.  
This is done to make the UI as simple as possible while also retaining all actions on one screen to help the retention of the user.



Since the tags page is not the sole page of our web app, Aidan also designed mock ups for the other pages (Find, Login, Signup)

*The Login & Sign Up Page*



*The Finder Pager  
Left represents looking for the tag using an arrow to guide the user  
Right depicts locating for the tag and picking it up*

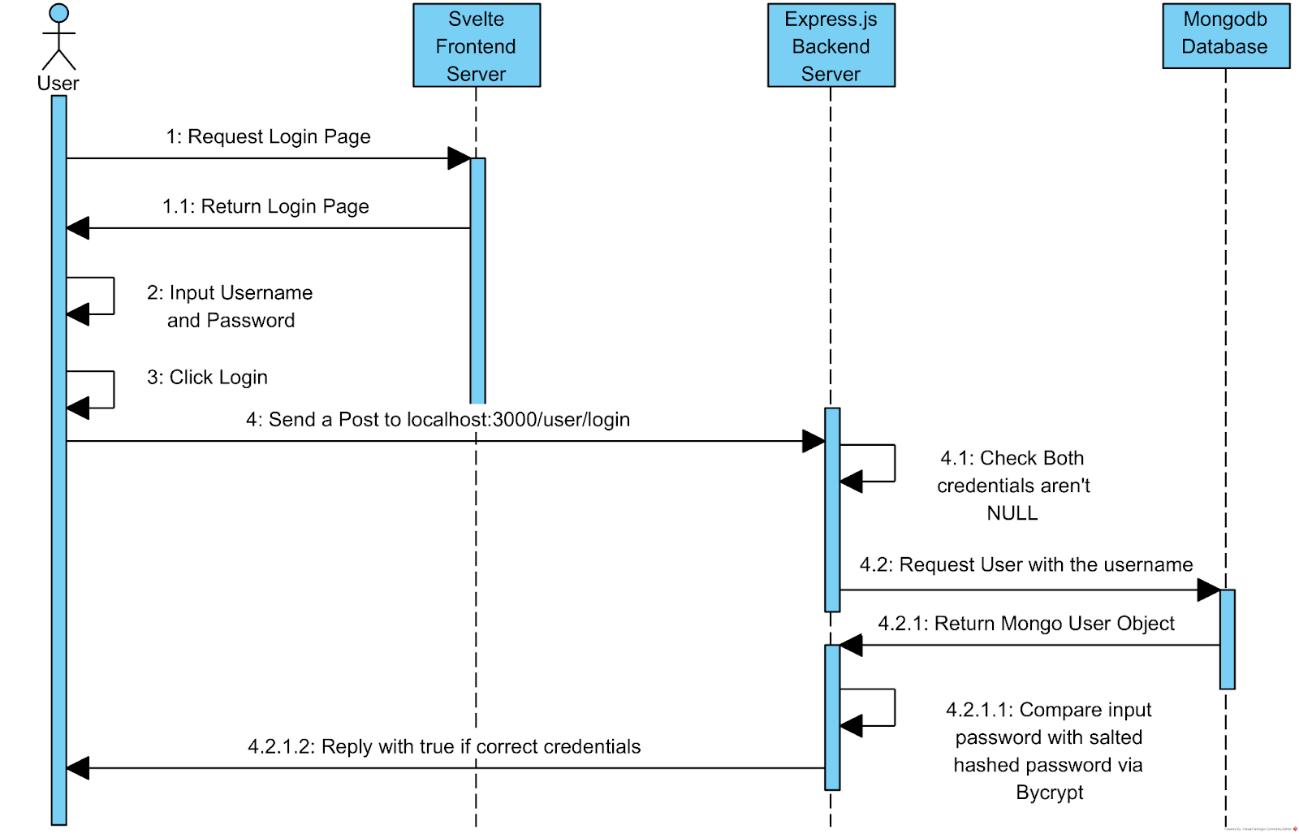
**User Back End Team**

From the user stories, we quickly found that it would be better to first setup the User backend to a rudimentary degree before starting the tags. Through this, we were able to setup the express.js server to respond to Adding Users, Deleting Users, Updating Users, and a function that tells you if you can login.

None of these functions have any proper authentication, but the general set has been made. We have also ensured that every password saved is hashed and salted with bycrpt from Mongodb’s best practices such that no plain text password can ever be saved.

Alfie Created the User backend functions to create, ‘login’, update, and delete. Rodion created the framework for encrypting and checking passwords, which Alfie then implemented into the MongoDB framework.

From this, Alfie created a diagram to summarize a scenario using the backend:



**Implementing Tag Design Team:**

# Sprint 6 (14/02/2023-21/02/2023) (Sprint 5 Retrospective)

For this retrospective we made good progress on a number of elements. Alex and Vishal were able to conjure a tags page that had working expandable tags with non-functioning buttons to modify when the backend is complete.

Alfie and Rodion were able to complete a rudimentary User Backend to add users and login, though there is a need to add authorisation for high-privilege calls such as deleting and updating. Oliver has been tasked with creating some unit tests to test the a typical usage of creating, editing, login, and deleting a user.

Aidan was able to create a multitude of diagrams outlining the structure of the website, which will be used for the implementation in Svelte in the Sprint 6.

In the specifics, the User Backend Team (Alfie, Rodion and Oliver) completed the extra user stories:

“Encrypt User data”

“Apply a hash function”

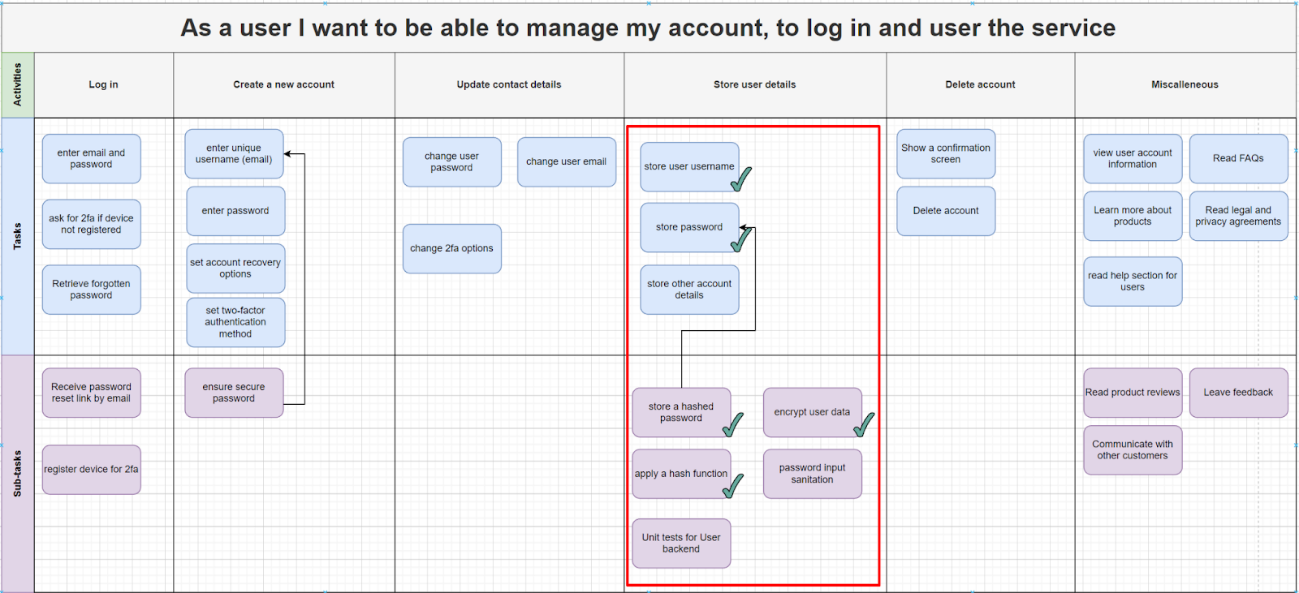
“Store a hashed password”

“Store user password”

This entails that the story map for managing accounts from:



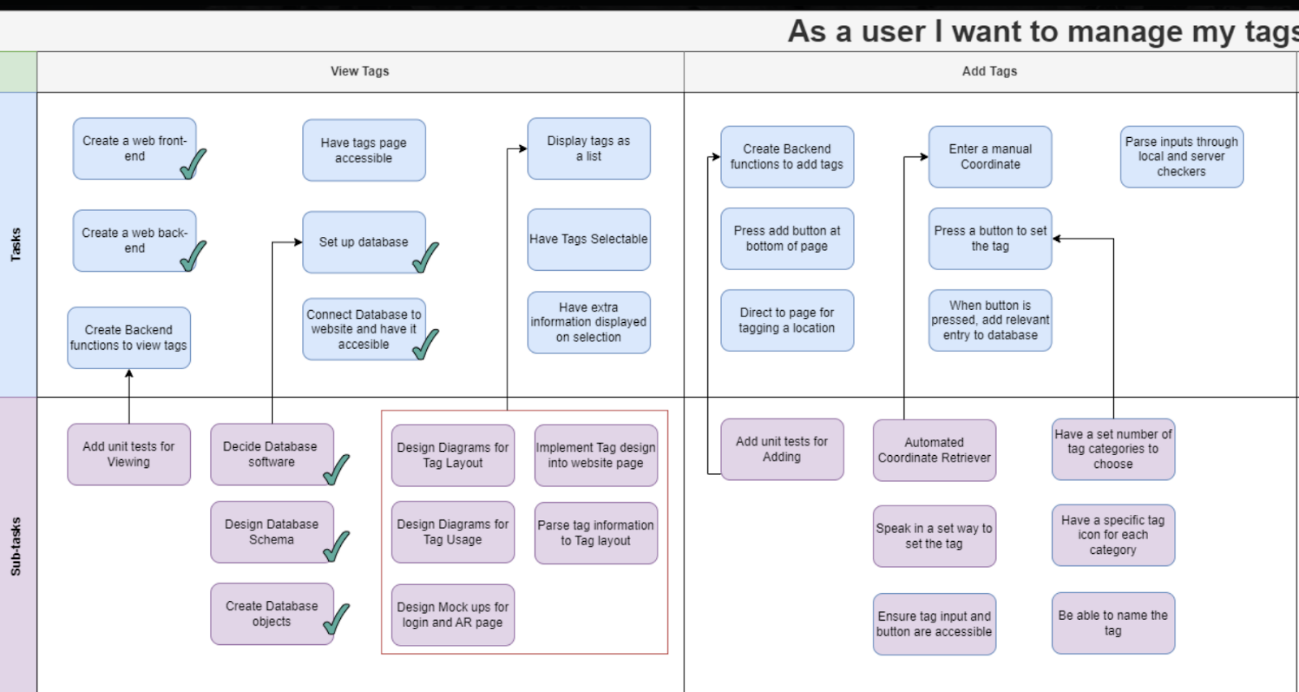
Has become:



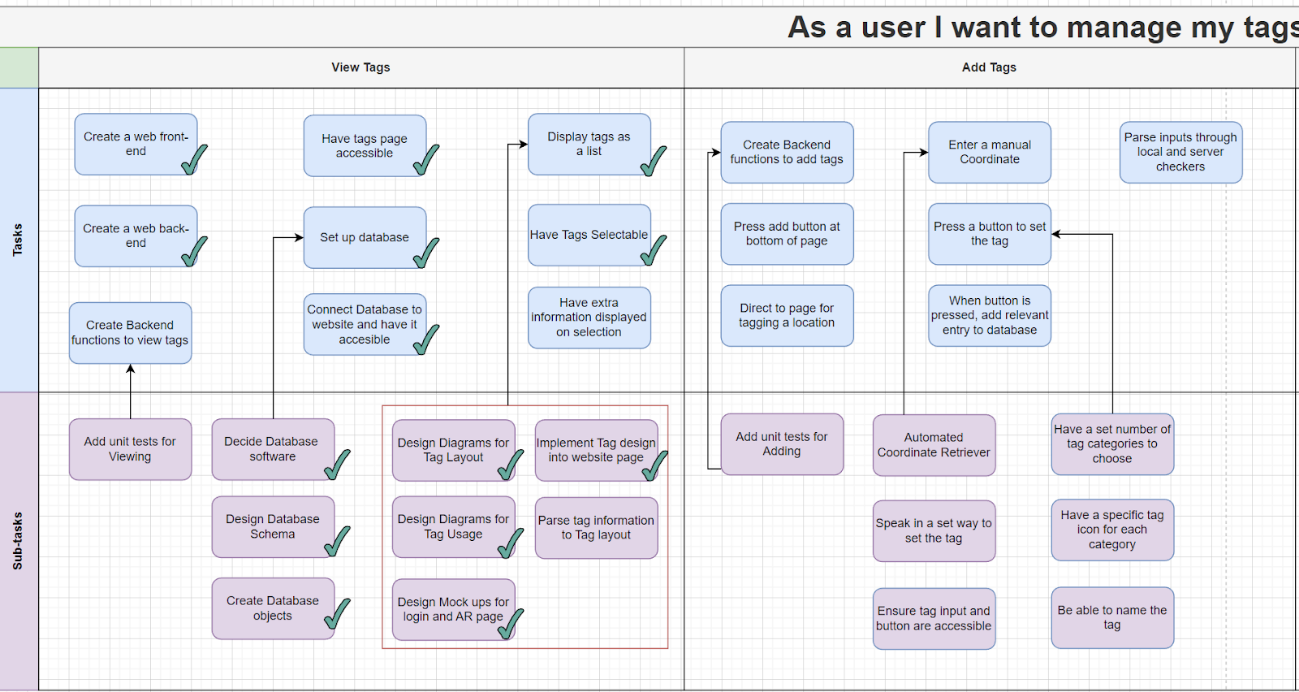
From this, we decided to assign specific people to each of the remaining user stories from the previous sprint. Rodion will work on creating the Backend functions for viewing tags, Alfie will work on adding unit tests for these, sanitizing passwords in the database. Oliver has been tasked to add unit tests for the User backend once Rodion is done.

For Aidan’s work, he has completed all the user stories assigned to him, designing all the mockups for the login and tags page, and how we expect it to function with activity diagrams. These will help a lot with implementing the front end in svelte. Alex and Vishal were also able to create a basic animated svelte implementation of a Tags page that has expanding tags in a list.

Moreso, they have also completed all their user stories. This has caused the story map for the user tag theme to go from:



To:



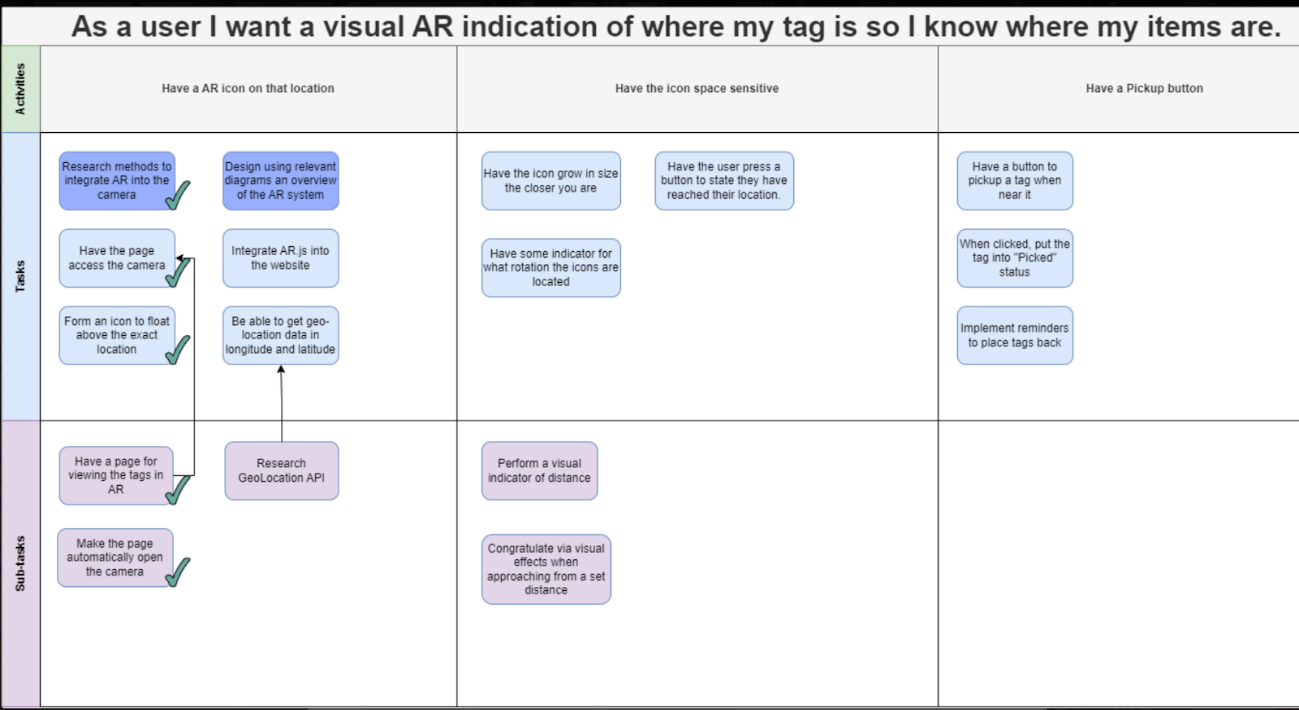
We have nearly completed an Epic.

From this, we assigned Alex to Parsing the Tag information in the backend (once Rodion has finished) into the frontend and displaying them. We then found that we could assign Aidan and Vishal to create the login page from the user story:

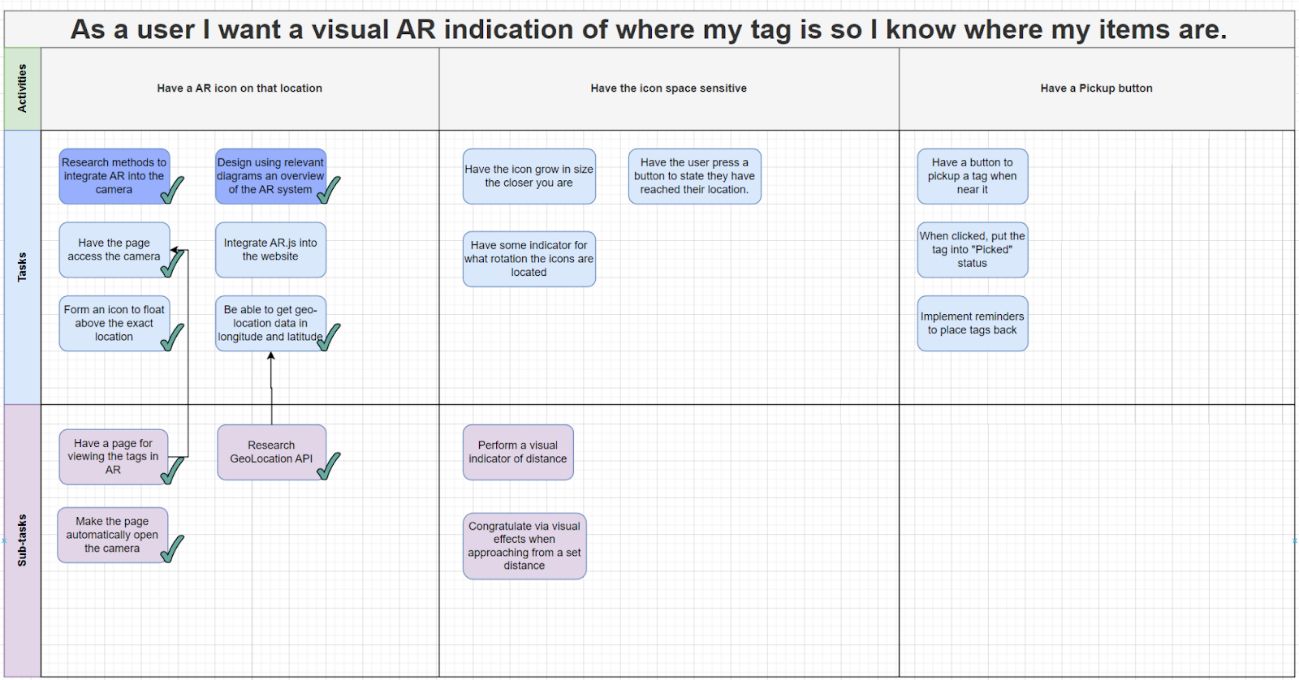
“Enter Email and Password” for the Login Epic.

Alan was not able to integrate AR.js into the svelte website. This is half due to there being no easy way to use it on your phone, as IBM has yet to properly fix the permissions to create a Kubernetes server and host the docker container we are developing on.

However, he was able to create a diagram for the AR system, Research how to get the Geolocation from a library, and has stated it works for HTTPs connections. His only job for this sprint is to see if he get the front end page to show it can access the camera on Svelte, not just a github page.

From this the AR story map has changed from:

To:



**AR Team**

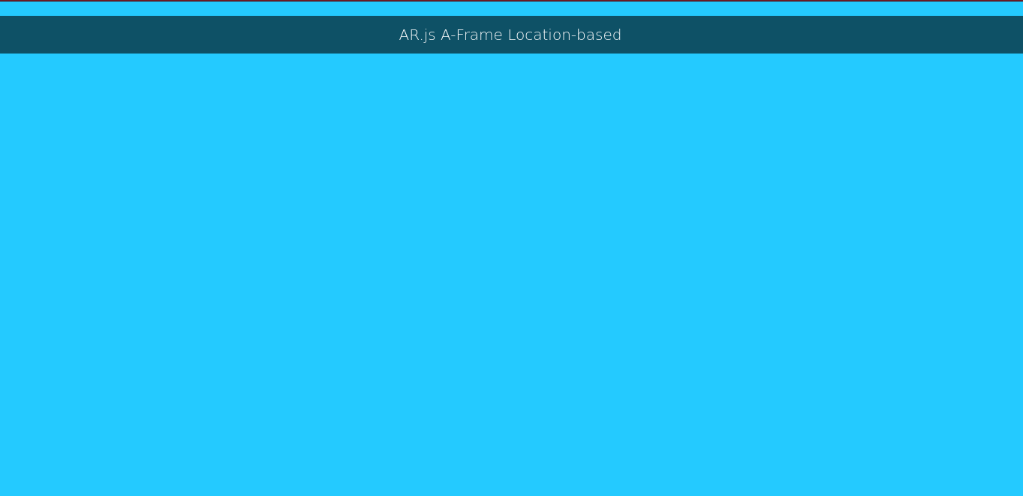
This sprint was focused on debugging the AR.svelte page, initial debugging research didn’t yield much as the use of AR.js in svelte specifically was undocumented. Fortunately Alan was able to find a git repo containing an example of AR.js used within a svelte project after examining this repo further it was found out the reason ar.js did not work was due to the fact the imported scripts below had to be put between a svelte:head tag instead of regular html head tag.



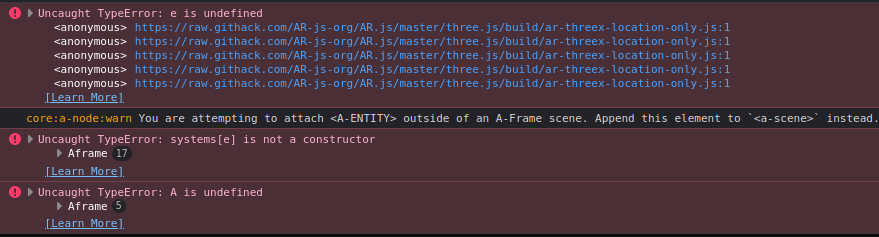
The git repo in question : <https://github.com/sectorxusa/svelte-aframe-arjs>

After examining source/routes/index.svelte we found the scripts tag were being placed in svelte:head, further inspiration was also gained from this repo about the possible need to **resize ar.js camera according to page dimensions.**

After implementing the fix, selecting the ar page lead to this appearing:

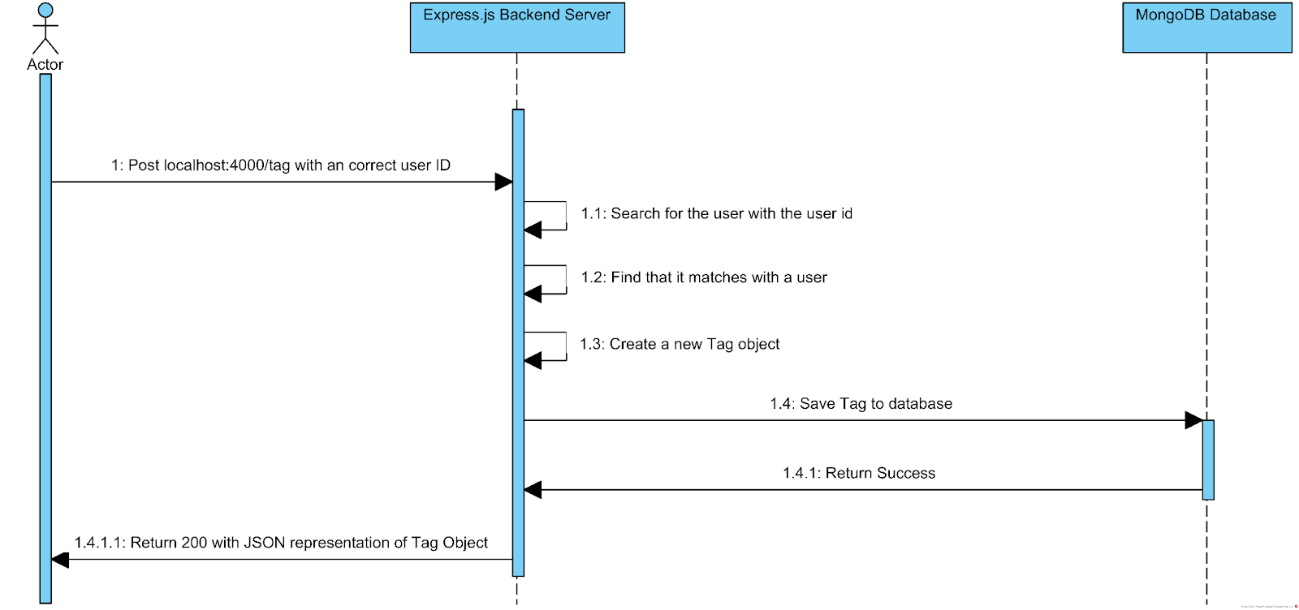


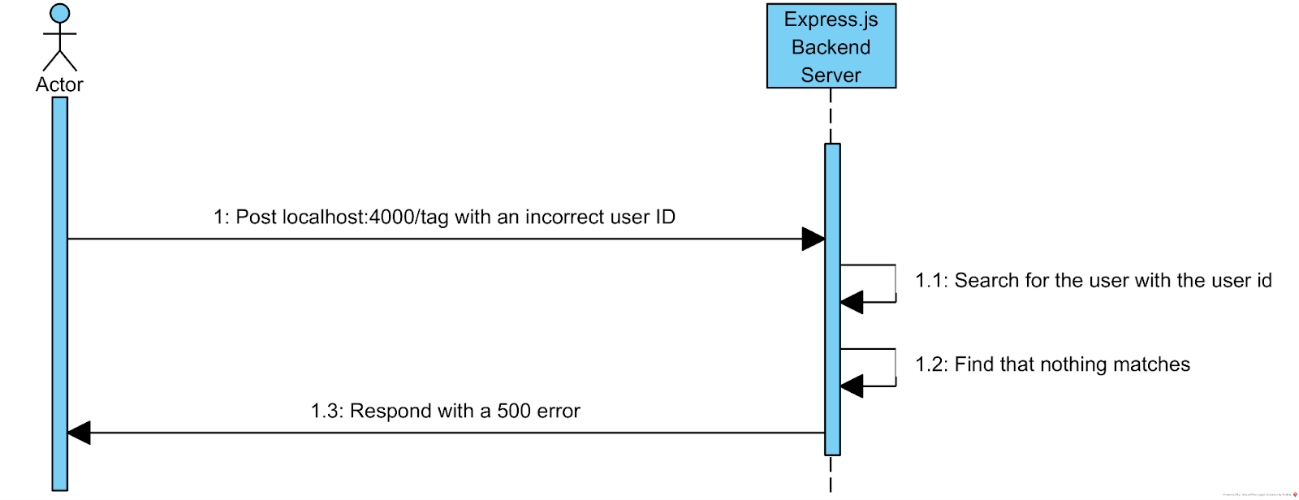
The reason the background is blue is because Alan was testing on his laptop which did not have access to a camera. Further testing will be done this sprint on a phone to see if it fully works. Additionally there are a significant amount of errors to handle, this is cause for concern for the next sprint.



Backend Team:

Rodion was able to create the Tag Backend through modifying the User backend from the previous sprint.

From this, Alfie wasable to formulate a number of unit tests for the Tags backend, and also create password sanitation for the User backend. Due to IBM architecture it took some time to make the mocha test framework.



These two diagrams outline some of the unit tests that were done for the Tag backend. Password sanitation involved regex.

# Sprint 7 (21/02/2023-28/02/2023) (Sprint 6 Retrospective)

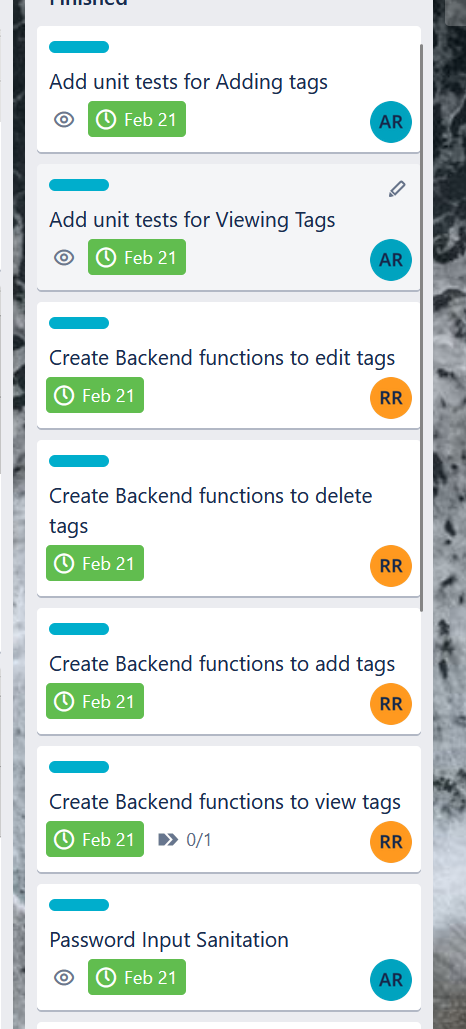
In this sprint retrospective we went over how everyone did. For Alan with AR.js, he was able to get the svelte page to display a tag in a specified location, however when it came to testing we found that some phones did not work at all for some reason. Through this Alan decided to look further into the issue to make sure it is a stable implementation of AR.js, and hope that it is not just a framework issue but something we can fix.

For Rodion, Olly and Alfie, on the backend team, Rodion was able to finish the Tag backend, or a basic version for basic implementation, Alfie was able to then create some Unit Tests for the tags, but as there was a complication with fixing the Test framework with ibmcli, Olly was not able to finish his work.

Aidan and Vishal was not able to complete the login page. Vishal has had issues setting up docker on his windows install, and we are attempting to circumvent this issue. Aidan decided he also wanted to make the homepage for the website, as it currently displays a blank screen.

Alex was set to work on integrating the Tag backend onto the website. As he had to wait for Rodion to finish, he needed some more time to learn how js get requests are handled, but he said it would only take a few days at most to properly implement.

From this, most of the user stories will have to be carried over. The finished user stories from this sprint were:

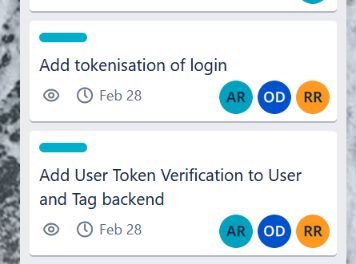


As Rodion did all of the backend functions, except a specific placing one (which can be done with updating), a few more than normal were able to be done.

From this, a serious part of the backend will now be set to be implemented, which is Cookie Login Tokenization. This allows the ability to login and save a cookie of some token that lets the website stay logged in. This is vital to create a coherent backend. Currently, every function is able to be called by anyone, which needs to change as Users should only be able to add tags for their own self, not for other people. Delete calls can delete any user, etc.

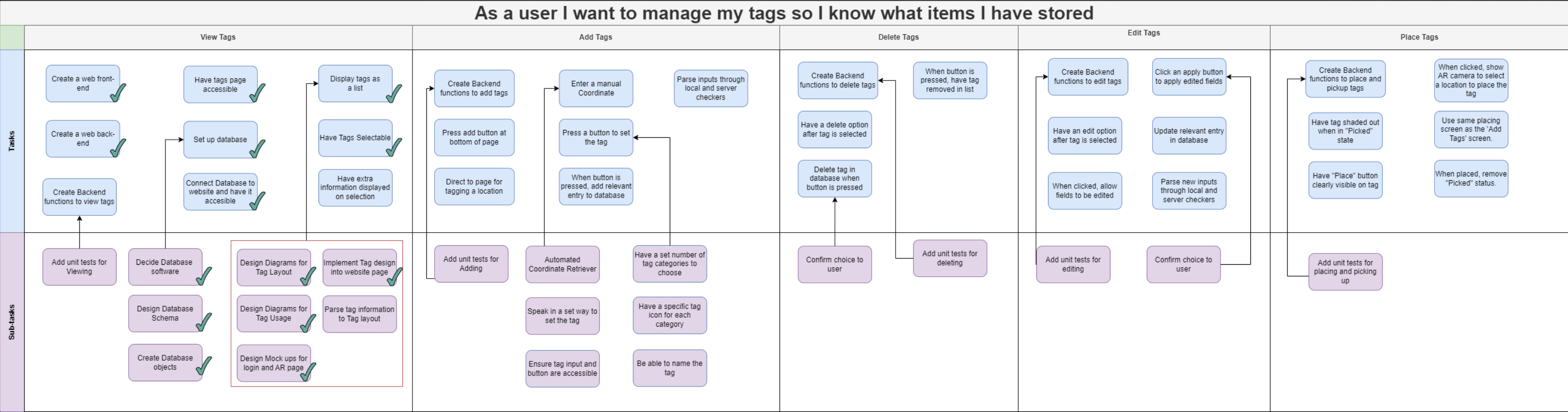
This token method will allow the frontend to send requests holding said token and thus allow the backend functions to authenticate. This is much more secure compared to just storing a plaintext password in the browser, thus our reason to choose this.

Thus, the new user stories assigned are:

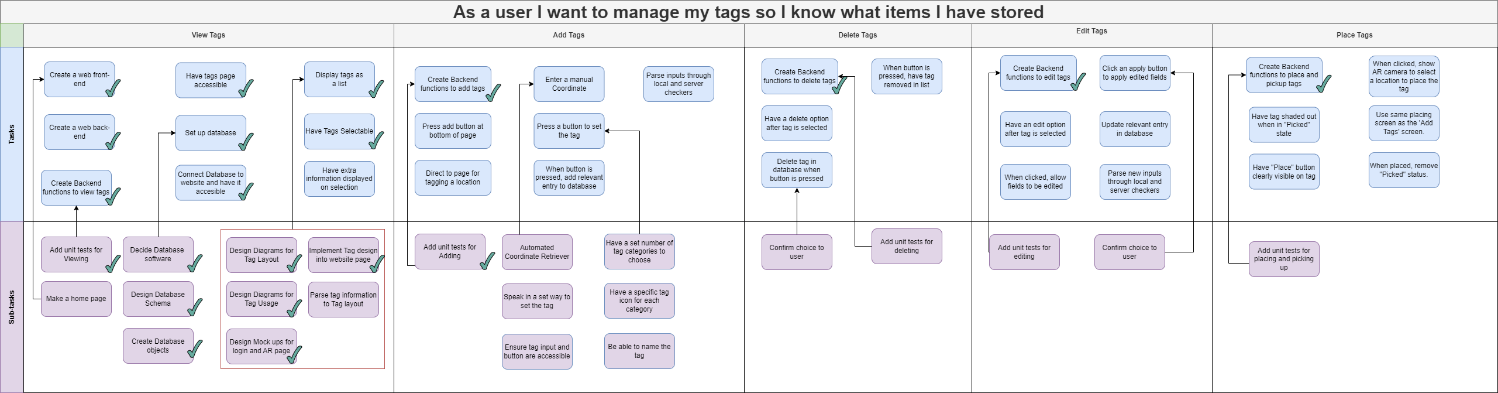


As it’s a large set of user stories we will have all of the backend team working on it. Because everyone else needs to finish the previous user stories, we are unable to mix the team up this sprint.

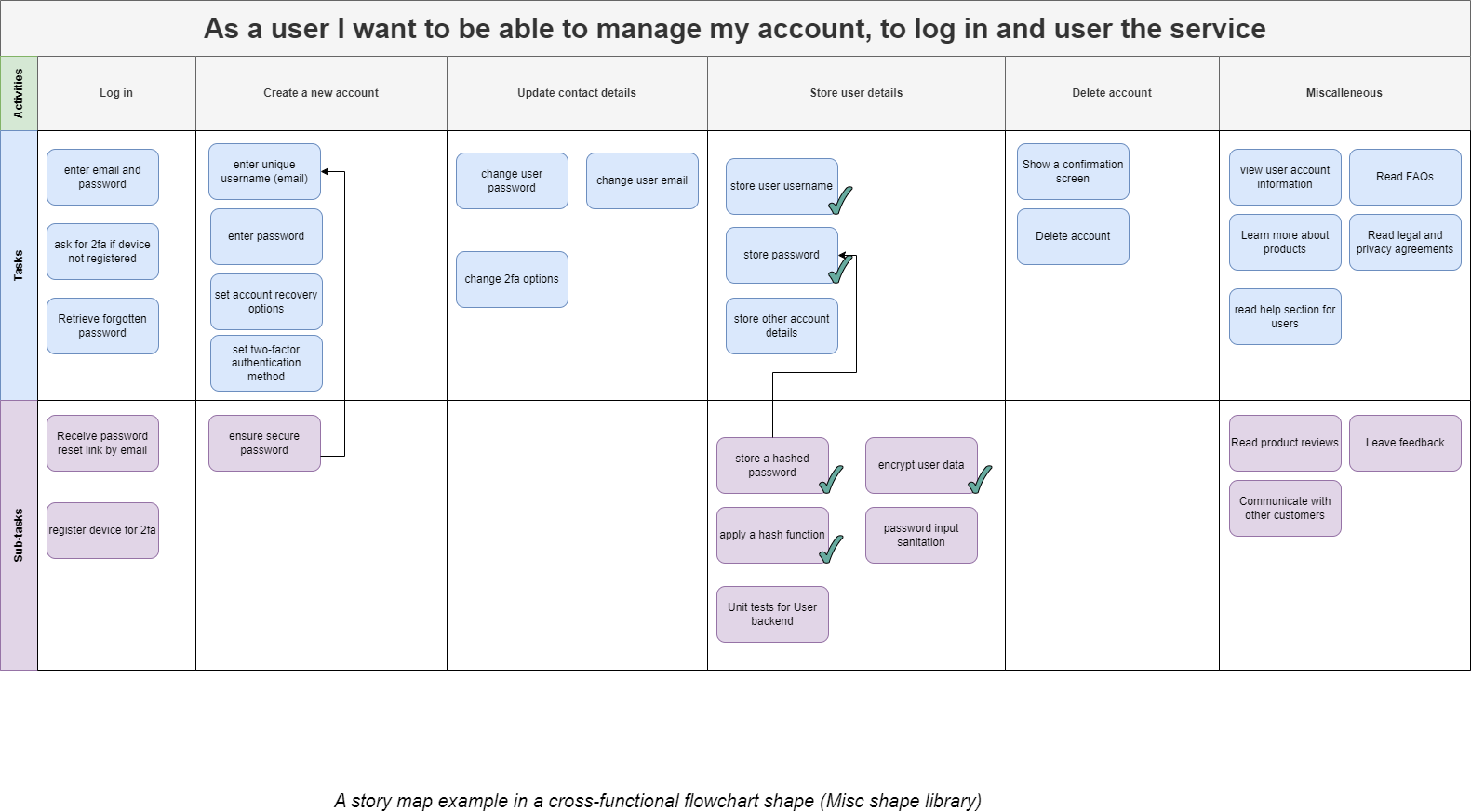
So, to summarize, our story maps have gone from:



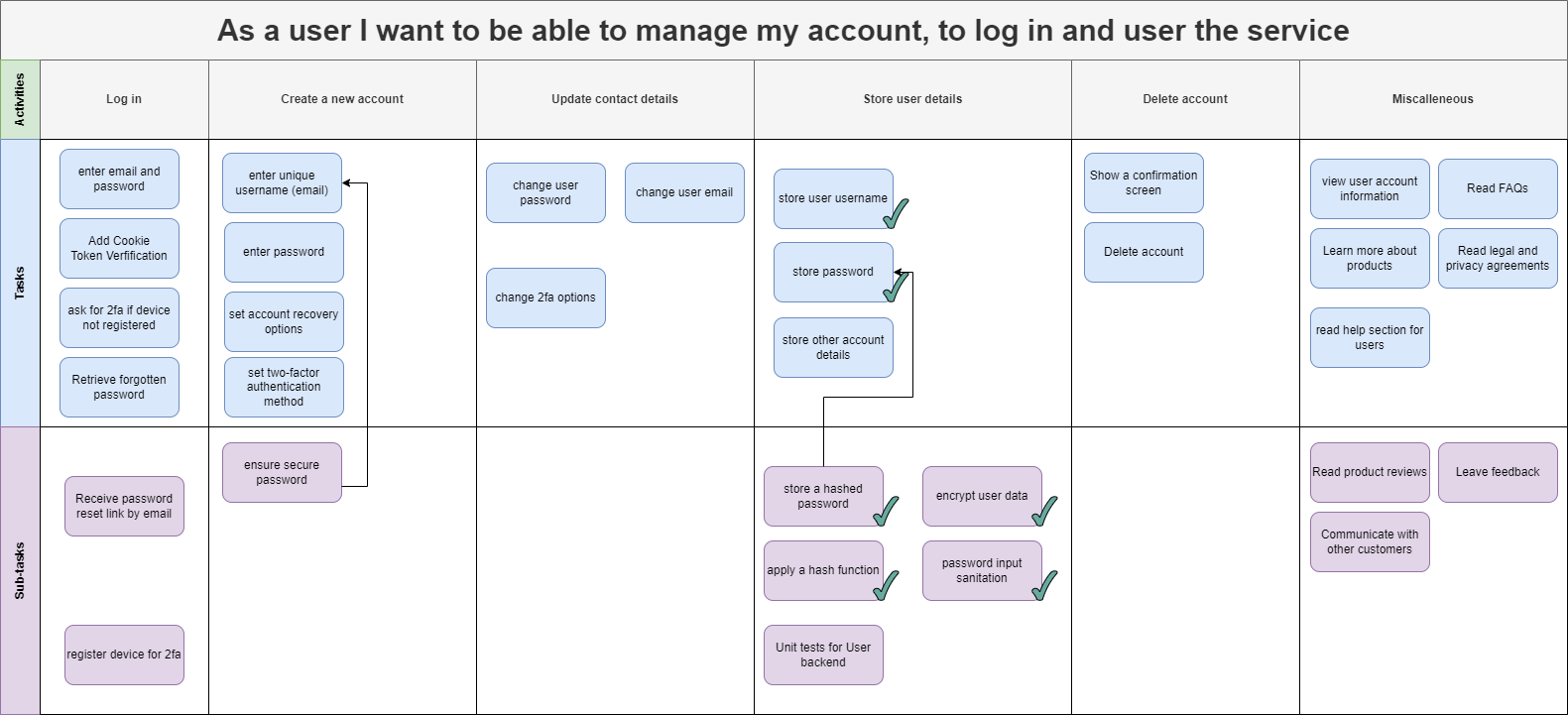
To:



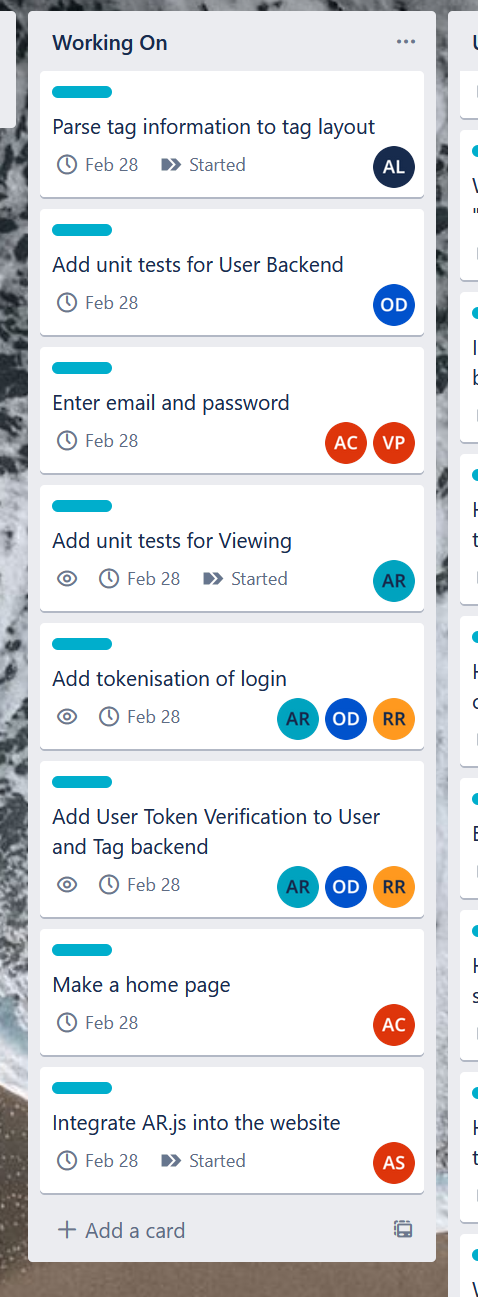
And with the Account theme:



To:



For this sprint, the assigned stories are:



The Unit tests will remain for the backend, as they are partially done and the difficult setup of mocha and Chia have been completed.

Everything else is mostly identical to the previous sprint. We deemed it necessary to speed up.