

# Sprint 2

- Sprint 2 Planning
- Sprint 2 Weekly Stand-ups
  - Date - 9 Sept 2021
  - Date - 17 Sept 2021
  - Date - 1 Oct 2021
- Sprint 2 Review
- Sprint 2 Retrospective
- Sprint 2 Testing
- Pair Programming
  - Date - 15 Sept 2021

# Sprint 2 Planning

## Overview

Meeting facilitator	Edward
Attendees	Jackey, Jawad, Hans, Mia
Date & Time	8-9-2021 3:15pm - 4:15pm

## Meeting agenda

- Decide the number of user stories
- Select the most important user stories
- Decide the tasks and assign ourselves on Trello

## Sprint 2 Backlog - User Stories

As a <Role>	I want to <Do Something>	So that <Achieve some goals>	Given <context>	When <an event>	Then <outcome>	Estimated	Assigned to
User	See all my deals	Decide on what deals to work on	A registered user is logged in	Goes to his dashboard	A list of deals in the format of a Kanban board is displayed, seperated by columns	Extra large (3 weeks)	Jackey
User	create a deal	manage deals with customer/company in organized and informed manner	registered user and logged into the app and on the sales pipeline page	clicks the 'Create' button on the corresponding deal tag and fill in the details(Client/Company Name, Deal name, deal amount, additionally perhaps-website, email, phone no., social media, contract, marketing material)	The system creates the deal.	Large (2 weeks)	Hans

User	update a deal(i.e. Change the status, Add in Notes)	keep information up to date	registered user and logged into the app and on the Dashboard	clicks 'Update' button on the corresponding deal tag AND fill in the details(Client/Company Name, Deal name, deal amount, <b>additionally perhaps-website, email, phone no., social media, contract, marketing material</b> )	the system updates the deal.	Large (2 weeks)	Mia
User	delete a deal	remove unrequired/unnecessary deals.	I am a registered user and logged into the app and on the sales pipeline page	when I click the 'Delete' button on the corresponding deal tag AND fill in the details(Client/Company Name, Deal name, deal amount, <b>additionally perhaps-website, email, phone no., social media, contract, marketing material</b> )	the system deletes the deal.	Large (2 weeks)	Edward
User	Register account	I can access the app	new user of the app on the registration page	fill credentials (username, first name, last name, email address, password, retype password)	Profile created in System	Small (5 days)	Jawad
User	update a user account(mainly password)	details be up-to-date when there some changes(or change passwords)	User is logged into their account	selects user account and changes user account's detail	the system updates to reflect the new details of the user account	Medium (1 week)	

Admin	See all users	Decide on actions to taken on users	An admin is logged in	goes to his front page	a list of users would be displayed	Small (5 days)	
Admin	Delete user	Fault management	An admin is logged in	goes to his front page and clicks on button to delete user from provided list of users	the user would be deleted from the database	Medium (1 week)	
Admin	Get all deals flagged for deletion	Fault management	An admin is logged in	goes to page that holds all flagged deals	a list of deals flagged for deletion would be displayed	Small (5 days)	
Admin	Delete deals flagged for deletion	Fault management	An admin is logged in	goes to page that holds all flagged deals and clicks on button to delete corresponding deal	the deal would be removed from the database	Medium (1 week)	

# Sprint 2 Weekly Stand-ups

Date: 9-Sep-2021

## Overview

Meeting facilitator	Edward
Attendees	Hans, Jackey, Jawad, Mia

## Stand Up

	Since the last Stand-up, I have...	Before the next Stand-up, I will...	What's slowing me down is...
Edward	worked on implementing protected routes for deals	implement more user functionality (user view all deals)  implement basic admin functionality	other assignments due soon
Hans	Worked on account register and create a deal functionality	attempt to finish all the implementation and begin work on the dashboard	1. deadlines are still overlapping  2. need to learn how to implement the packages that are going to be used
Jackey	implemented protected routes and token management, made a dashboard	App will hold login states,  dashboard will be connected to db	frontend session handling is new and will need to be reimplemented
Mia	Implemented dashboard front-end functionality	Continue on the rest of the dashboard functionality	Other Assignments

Jawad	<ol style="list-style-type: none"> <li>1. Peer reviewed code for crud deals backend</li> <li>2. Tested the deals api</li> <li>3. Prepared for the Sprint 1 review and Sprint 1 retrospective meetings</li> <li>4. Prepared for Sprint 2 Planning</li> </ol>	<ol style="list-style-type: none"> <li>1. Create Customer schema</li> <li>2. Update deal schema to contain customer schema <b>BACKEND</b></li> </ol>	<ol style="list-style-type: none"> <li>1. Other commitments - Assignment, Mst</li> </ol>
-------	---	--	--

Date: 17-Sep-2021

## Overview

<b>Meeting facilitator</b>	<b>Edward</b>
<b>Attendees</b>	Hans, Jackey, Jawad, Mia

## Stand Up

	<b>Since the last Stand-up, I have...</b>	<b>Before the next Stand-up, I will...</b>	<b>What's slowing me down is...</b>
Edward	<ol style="list-style-type: none"><li>1. Created Admin Schema</li><li>2. Created basic functionality for Admin; login, register, view all users, delete user</li></ol>	<ol style="list-style-type: none"><li>1. Create function for users to flag deals for deletion</li><li>2. Finish up admin functionality</li></ol>	-
Hans	Created login/register page elements that connects to the db	finish dashboard and implement all functionalities	<ol style="list-style-type: none"><li>1. feverish for a few days after vaccine</li><li>2. have to learn redux for states</li></ol>
Jackey	tested different methods to hold login state without success	implement a centralized store for credentials, perhaps use React-redux	Impeding due dates causing existential dread.
Mia	<ol style="list-style-type: none"><li>1. Working to make the dashboard draggable</li><li>2. Overlay design for deal creation function</li></ol>	<ol style="list-style-type: none"><li>1. Complete the draggable and deal creation functionality</li><li>2. Additional front-end functionality as required.</li></ol>	<ol style="list-style-type: none"><li>1. Few Assignments</li><li>2. Draggable functionality of the dashboard is difficult to implement</li></ol>

Jawad	<ol style="list-style-type: none"><li>1. Create Customer schema</li><li>2. Update deal schema to contain customer schema BACKEND</li></ol>	<ol style="list-style-type: none"><li>1. Test the api for deals and customer</li><li>2. Update the User details (Backend)</li><li>3. View all deals BACKEND (User)</li></ol>	Hopefully nothing this week.
-------	--	--	------------------------------



Date: 01-Oct-2021

## Overview

Meeting facilitator	Edward
Attendees	Hans, Jackey, Jawad, Mia

## Stand Up

	Since the last Stand-up, I have...	Before the next Stand-up, I will...	What's slowing me down is...
Edward	<ol style="list-style-type: none"><li>1. Implemented working authentication for Admin</li><li>2. Finished up Admin functionality, now does everything client requested</li></ol>	Plan for Sprint 3	Assignments from other subjects
Hans	implemented dashboard and its functionalities	<ol style="list-style-type: none"><li>1. finish styling on the dashboard</li><li>2. maybe add admin UI components</li></ol>	<ol style="list-style-type: none"><li>1. steep learning curve of redux</li><li>2. assignments</li></ol>
Jackey	<ol style="list-style-type: none"><li>1. made a new ui</li><li>2. implemented user friendly navigation and menus</li><li>3. implemented password changing functionality</li><li>4. unified authentication functions and reduced redundant code</li></ol>	plan for sprint 3 explore data visualization components	"Turing Machines" Lacking best practice guidelines.

Mia	<ol style="list-style-type: none"> <li>1. Added elements to the previously rendered UI.</li> <li>2. Improved the Drag and Drop functionality with smoother user interaction</li> <li>3. Styling</li> </ol>	<ol style="list-style-type: none"> <li>1. Improve styling</li> <li>2. Merge the dashboard UI with crud modal functionalities.</li> </ol>	<ol style="list-style-type: none"> <li>1. Other subject responsibilities</li> <li>2. Inexperience in front-end coding.</li> </ol>
Jawad	<ol style="list-style-type: none"> <li>1. View deals backend</li> <li>2. Update user password backend</li> <li>3. Test the register page frontend, (confluence)</li> <li>4. Add estimate (Time) to each user stories</li> <li>5. Code review(pull requests)</li> <li>6. troubleshooting</li> </ol>	<ol style="list-style-type: none"> <li>1. Will decide after sprint 3 planning next week</li> </ol>	Approaching the end of semester, so assignments and lecture reviews/ catchups

# Sprint 2 Review

## Overview

<b>Date &amp; Time</b>	<b>4-Oct-2021 3:15 AEST</b>
<b>Meeting facilitator</b>	Mia
<b>Attendees</b>	Jawad, Jackey, Hans, Edward, Wei Li (Client)

## Meeting agenda

- Start – Sprint Review Meeting starts
- Welcome the Stakeholders – Product Owner welcomes the stakeholders to attend the review
- Present Review Agenda – Product Owner presents the agenda for the Sprint Review
  - Product demo
  - Client feedback
  - Comments on next Final sprint
- Present Product Increments – Development Team present the product demo that have been implemented in the Sprint
- Get Feedback – Product Owner asks the stakeholders for feedback regarding the product that have been delivered
- Update- Progress rate in the next sprint

## Minutes/ Notes

1. Warned us of multiple logins in different browser, but was non issue as we used JWT
2. Wanted for admin to be able to update customer details (including email)
3. Looking forward to our admin front end
4. Overall positive reception from client
5. Client Question:
  1. Deployed? Yes link supplied
  2. State synchronized? Not an issue as jwt is used
6. Positives
  1. Drag and drop work good
  2. Admin backend
7. Feedback: work on resolving errors reported on console
8. Maybe later work on single browser tab store- redux
9. Need to fix error and finish admin features

## Resources:

<http://visual-paradigm.com/scrum/what-is-sprint-review/>

# Sprint 2 Retrospective

## Overview

Sprint 2 retrospective - <https://www.atlassian.com/team-playbook/plays/4-ls-retrospective-technique>

<b>Meeting facilitator</b>	<b>Edward</b>
<b>Attendees</b>	Jawad, Hans, Mia, Jackey
<b>Date &amp; Time</b>	1st of October 2021, 1640 AEST

## Minutes/ Notes

	<b>LOVED: what you loved about your work over the time period.</b>	<b>LONGED FOR: what you wish you'd had.</b>	<b>LOATHED: what made life worse back then. What do you hope will never happen again?</b>	<b>LEARNED: what you learned from your successes and your mistakes.</b>
Edward	Seeing how the different parts we built pieced together into a working web app	Better styling sense to provide input when designing the website.  More initiative	Figuring out authentication to handle both users and admin	Learned to work out authentication for both users and admin  Learned to be more vocal when working with team
Hans	Learning user states and syncing states with backend	more prior information and skills about redux	401 errors and other errors	learned about user states and how to sync what is showing on screen to the backend
Jackey	Building functionality is easier without having to worry about data synchronization	knowledge about workflow of a project in a production environment	"Error too many re-renders"	learned to call api with aync functions, setting up and using a global store, using middlewares with React, better decision making using local state and global states.

Jawad	working collaboratively, efficient use of git and GitHub.	contribute to the frontend	error with cors, assignments	<ol style="list-style-type: none"> <li>1. Learnt conceptually how redux works</li> <li>2. not to endeavor into stuff with half knowledge ; to end up wasting time and nothing constructive.</li> </ol>
Mia	Rendering colorful creative design with styling	Gradient Styling, polished looking and smoother working UI	Draggability code complications and bootstrapping	Front-end styling, working with data structures and other coding elements React and CSS

# Sprint 2 Testing

- Create Deals

## Frontend

Test: create the deal ✓

Precondition: Successfully authorized, login in and have a valid jwt token

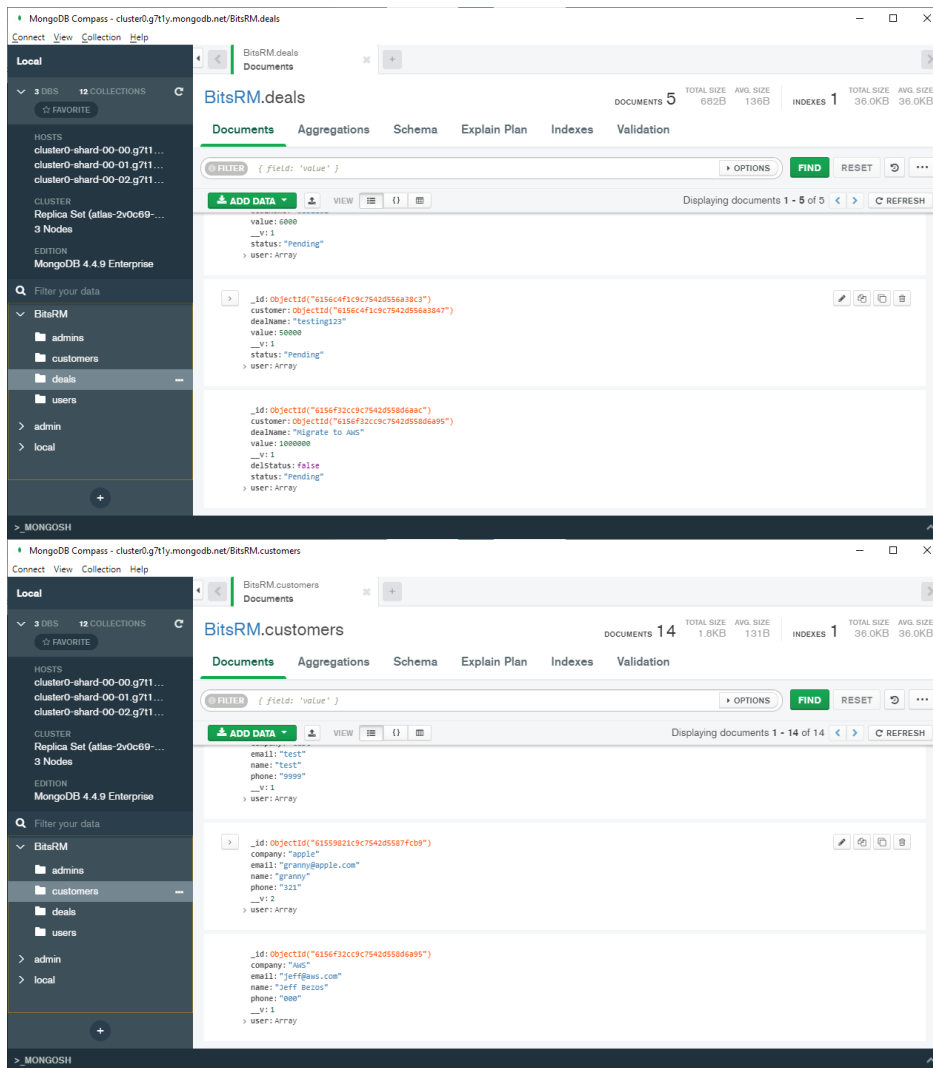
The image shows two screenshots of the BitsRM web application. The top screenshot displays a modal form for creating a new deal. The form fields are as follows:

Field	Value
Deal Name *	Migrate to AWS
Customer Name *	Jeff Bezos
Customer Company Name *	AWS
Customer Email *	jeff@aws.com
Customer Phone *	000
Deal Value *	1000000

Buttons: CANCEL, SUBMIT

The bottom screenshot shows the main dashboard with a table of deals categorized by status: Pending, Accepted, Declined, and Done. The 'Pending' category shows the deal 'Migrate to AWS' with a value of '\$1.00M' and a green checkmark, indicating successful creation. The other categories show existing test deals.

Status	Deal Name	Value	Actions
Pending	Migrate to AWS	\$1.00M	EDIT, DELETE
Accepted	test5	\$100K	EDIT, DELETE
Accepted	test deal 2	\$25.0K	EDIT, DELETE
Declined	test deal 5	\$30.0K	EDIT, DELETE
Done	test2	\$3.333K	EDIT, DELETE



## Backend

Test: API call to create the deal ✓

Precondition: Successfully authorized, login in and have a valid jwt token

POST <https://bits-please-api.herokuapp.com/user/deal>

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "dealName": "support",
3   "customer": {
4     "company": "fb",
5     "name": "john",
6     "email": "john@fb.com",
7     "phone": "042",
8     "value": 6000
9   }
}
```

Body Cookies Headers (9) Test Results Status: 201 Created

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "msg": "Deal created!",
4   "deal": {
5     "user": [
6       "6129ea39a1d1400b6c326c99"
7     ]
8   }
9 }
```

MongoDB Compass - cluster0.g7ty.mongodb.net/BitsRM.deals

Local

3 DBS 19 COLLECTIONS

cluster0-shard-00-00.g7ty...  
cluster0-shard-00-01.g7ty...  
cluster0-shard-00-02.g7ty...

CLUSTER  
Replica Set (status: OK)

3 Nodes

VERSION  
MongoDB 4.4.0 Enterprise

Filter your data

BitsRM

admin  
customers  
deals  
users

admin  
local

Documents

BitsRM.deals

documents 1 TOTAL SIZE 1,300B AVG SIZE 1,300B INDEXES 1 TOTAL SIZE 36,0Kb AVG SIZE 36,0Kb

Documents Aggregations Schema Explain Plan Indexes Validation

EDITED { fields: "value" }

+ OPTIONS FIND RESET

+ ADD DATA + VIEW

Displaying documents 1 - 1 of 1

```
{ "_id": "6129ea39a1d1400b6c326c99",  
  "customer": {  
    "company": "fb",  
    "name": "john",  
    "email": "john@fb.com",  
    "phone": "042",  
    "value": 6000  
  },  
  "status": "pending",  
  "user": "6129ea39a1d1400b6c326c99" }
```

- **Update Deals**

## Backend

Test: API call to update the deal ✓

Precondition: Successfully authorized, login in (have a valid jwt token) and deal ID to update



The image shows a REST client interface with a PUT request to `https://bits-please-api.herokuapp.com/user/deal/{{idDeal}}`. The request body is a JSON object:

```
1 {
2   "dealName": "Callcenter",
3   "customer": 1,
4   "company": "Fb",
5   "name": "Johson",
6   "email": "John@fb.com",
7   "phone": "042",
8 }
```

The response status is 200 OK. The response body is a JSON object:

```
1 {
2   "success": true,
3   "msg": "Deal updated!",
4   "deal": {
5     "user": [
6       "6129ea39a1d1400b6c326c99"
7     ],
8     "status": "Pending",
9   }
10 }
```

Below the REST client, the MongoDB Compass interface is shown. It displays the `BitsRM.deals` collection with one document:

```
{ "_id": ObjectId("614598ed1589c3b640b86f"), "customer": ObjectId("61459a491589c3b640b841"), "dealName": "Callcenter", "value": 7000, "v": 1, "status": "Pending", "user": Array }
```

- **Delete Deals**

## Back End

Test: API call to delete the deal ✓

Precondition: Successfully authorized, login in (have a valid jwt token) and deal ID to delete

DELETE

https://bits-please-api.herokuapp.com/user/deal/{{idDeal}}/

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (9)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

"success": true,

"msg": "Deal deleted!"

MongoDB Compass - cluster0.g71y.mongodb.net/BitRM.deals

Connect View Collection Help

Local

1 DBS 12 COLLECTIONS

cluster0-shard-00-00.g711...  
cluster0-shard-00-01.g711...  
cluster0-shard-00-02.g711...

CLUSTER  
Replica Set (atlas-2v0c69-...)  
3 Nodes

EDITION  
MongoDB 4.4.0 Enterprise

Filter your data

BitRM

admins  
customers  
deals  
users

admin  
local

BitRM.deals

Documents

DOCUMENTS 1 TOTAL SIZE 135B AVG. SIZE 135B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW 0 REFRESH

Displaying documents 0 - 0 of N/A

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file

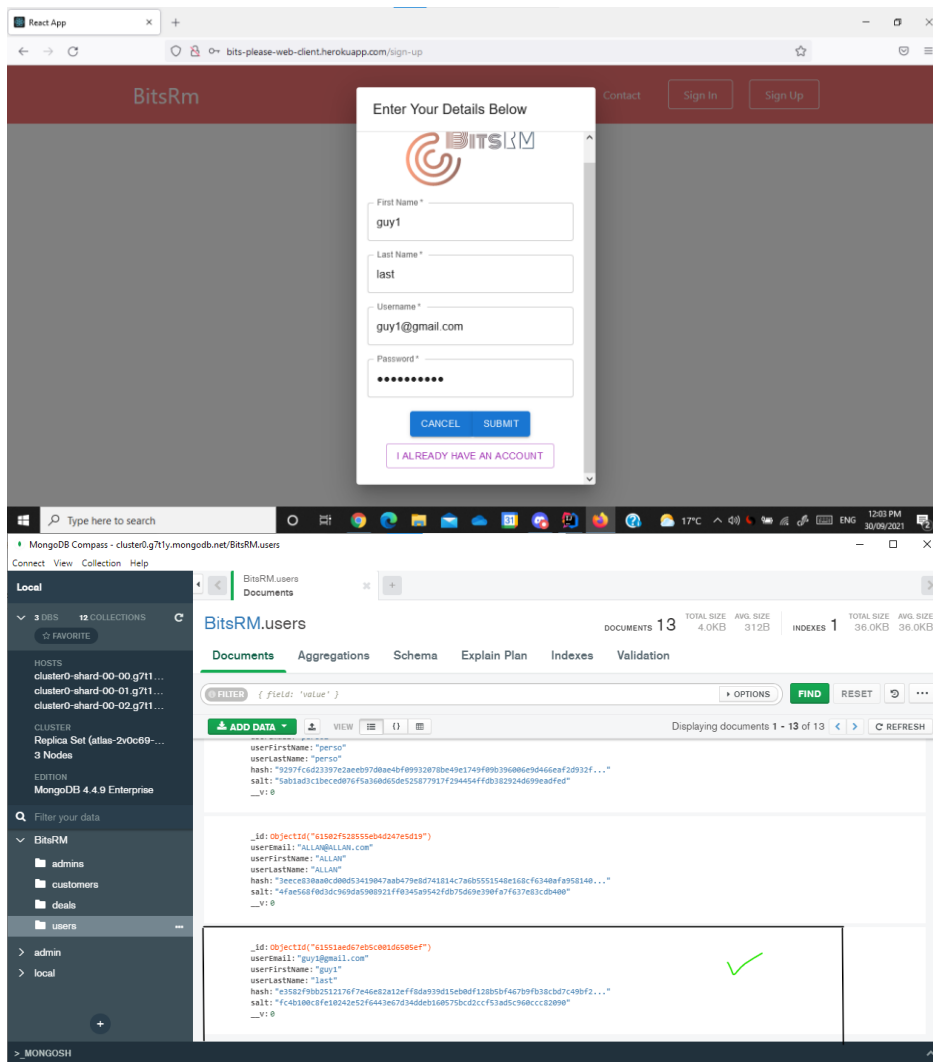
Import Data

> MONGODB

- Register User

## Front End

Test: User registration form works and user is successfully registered in the database ✓



## Back End

Completed in the sprint-1

- **Update password:**

## Back End

Test: API call to update password, check the hash before passwords update and hash after password update and compare if different, also try to login with new password ✓

Precondition: Successfully authorized, login in (have a valid jwt token) and deal ID to delete

MongoDB Compass - cluster0.g71ty.mongodb.net/BitRMUsers

Local

3 DBS 12 COLLECTIONS

HOSTS

- cluster0-shard-00-00.g71ty...
- cluster0-shard-00-01.g71ty...
- cluster0-shard-00-02.g71ty...

CLUSTER

Replica Set (atlas-2v0c68-...)

3 Nodes

EDITION

MongoDB 4.4.9 Enterprise

Filter your data

BitRM

- admins
- customers
- deals
- users

admin

local

BitRM.users

Documents

Documents 14 TOTAL SIZE 4.3KB AVG. SIZE 313B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

Filter { field: "value" }

ADD DATA VIEW

Displaying documents 1 - 14 of 14

Refresh

Document 1:

```
{
  "userFirstName": "person",
  "userLastName": "person",
  "hash": "d0f37e0a748979425a15850f6c5cdf9e8c3585c3a80de4d9af06377405bdeafe...",
  "salt": "f1b0ce4dbd4f62f94c4c403d1d846f2e0b68d39f90f18f7b7405b1621a",
  "_id": 0
}
```

Document 2:

```
{
  "_id": ObjectId("6129ea39a1d4006c326c99"),
  "userEmail": "guy@gmail.com",
  "userFirstName": "john",
  "userLastName": "Smith",
  "hash": "ae0edec619f6a7708922e3ac854ee3ea7c0a2d74a5b832da043d70627e346070b71...",
  "salt": "d302691358948a3c06f94c5d7c8548a09182417c72fe70e131f340587950",
  "_id": 0
}
```

Document 3:

```
{
  "_id": ObjectId("613545e190f53001d6ca33d"),
  "userEmail": "mugug@gmail.com",
  "hash": "4e32094d087949551d5cccd8c1e39aca3e643779740b333f2166f773e53701585...",
  "salt": "c1caf93ba2bdc8f8f94c9f919705bdc6820dbf7c34805fcl1c95567932d5",
  "_id": 0
}
```

Document 4:

```
{
  "_id": ObjectId("613566e190f53001d6ca340"),
  "userEmail": "mugug@gmail.com",
  "hash": "4e32094d087949551d5cccd8c1e39aca3e643779740b333f2166f773e53701585...",
  "salt": "c1caf93ba2bdc8f8f94c9f919705bdc6820dbf7c34805fcl1c95567932d5",
  "_id": 0
}
```

MONGOSH

MongoDB Compass - cluster0.g71ty.mongodb.net/BitRMUsers

Local

3 DBS 12 COLLECTIONS

HOSTS

- cluster0-shard-00-00.g71ty...
- cluster0-shard-00-01.g71ty...
- cluster0-shard-00-02.g71ty...

CLUSTER

Replica Set (atlas-2v0c68-...)

3 Nodes

EDITION

MongoDB 4.4.9 Enterprise

Filter your data

BitRM

- admins
- customers
- deals
- users

admin

local

BitRM.users

Documents

Documents 14 TOTAL SIZE 4.3KB AVG. SIZE 313B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

Filter { field: "value" }

ADD DATA VIEW

Displaying documents 1 - 14 of 14

Refresh

Document 1:

```
{
  "userFirstName": "person",
  "userLastName": "person",
  "hash": "d0f37e0a748979425a15850f6c5cdf9e8c3585c3a80de4d9af06377405bdeafe...",
  "salt": "f1b0ce4dbd4f62f94c4c403d1d846f2e0b68d39f90f18f7b7405b1621a",
  "_id": 0
}
```

Document 2:

```
{
  "_id": ObjectId("6129ea39a1d4006c326c99"),
  "userEmail": "guy@gmail.com",
  "userFirstName": "john",
  "userLastName": "Smith",
  "hash": "ae0edec619f6a7708922e3ac854ee3ea7c0a2d74a5b832da043d70627e346070b71...",
  "salt": "d302691358948a3c06f94c5d7c8548a09182417c72fe70e131f340587950",
  "_id": 0
}
```

Document 3:

```
{
  "_id": ObjectId("613545e190f53001d6ca33d"),
  "userEmail": "mugug@gmail.com",
  "hash": "4e32094d087949551d5cccd8c1e39aca3e643779740b333f2166f773e53701585...",
  "salt": "c1caf93ba2bdc8f8f94c9f919705bdc6820dbf7c34805fcl1c95567932d5",
  "_id": 0
}
```

Document 4:

```
{
  "_id": ObjectId("613566e190f53001d6ca340"),
  "userEmail": "mugug@gmail.com",
  "hash": "4e32094d087949551d5cccd8c1e39aca3e643779740b333f2166f773e53701585...",
  "salt": "c1caf93ba2bdc8f8f94c9f919705bdc6820dbf7c34805fcl1c95567932d5",
  "_id": 0
}
```

MONGOSH

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import

POST L B. PUT u. GET S. POST L B. PUT u. GET S.

BitRM-Production / User / update password

PUT https://bts-please-api.herokuapp.com/user/password

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Body

```
{
  "oldPassword": "abcde12345",
  "newPassword": "Abcde12345"
}
```

Body Cookies Headers (9) Test Results

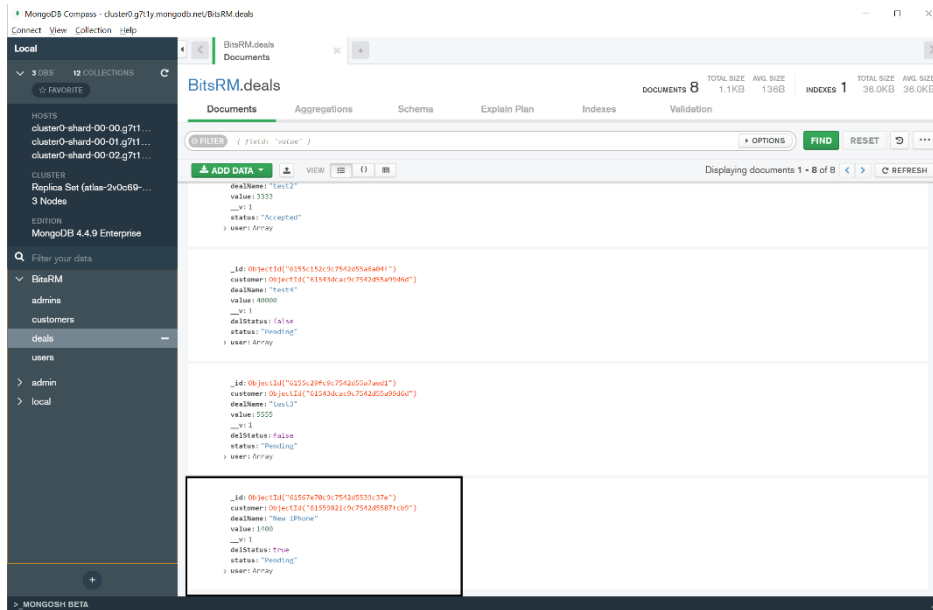
Status: 200 OK Time: 319 ms Size: 115 KB Save Response

Body

```
{
  "expires": "1d"
}
```



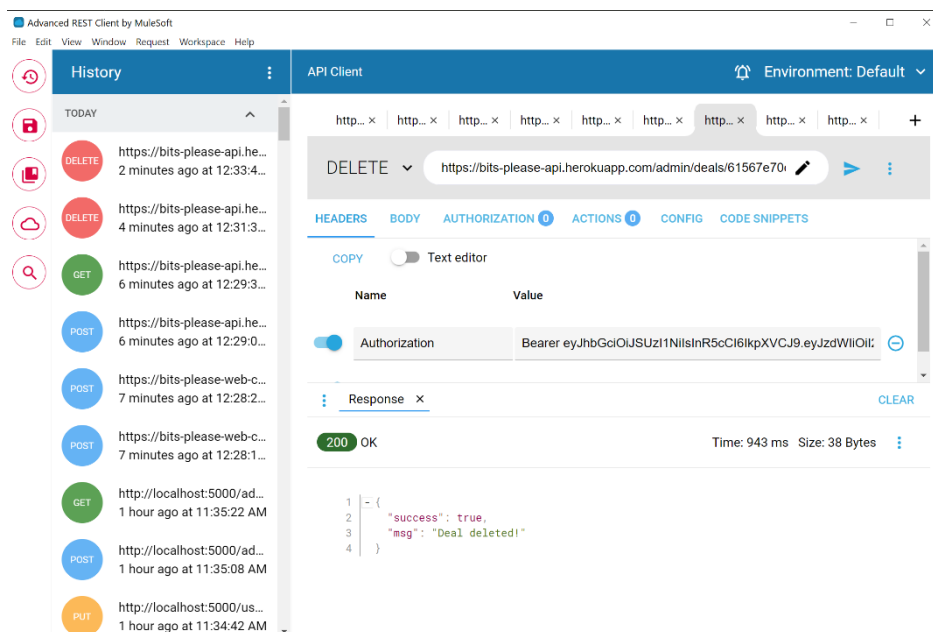
Precondition: Successfully authorized, login in (have a valid jwt token)

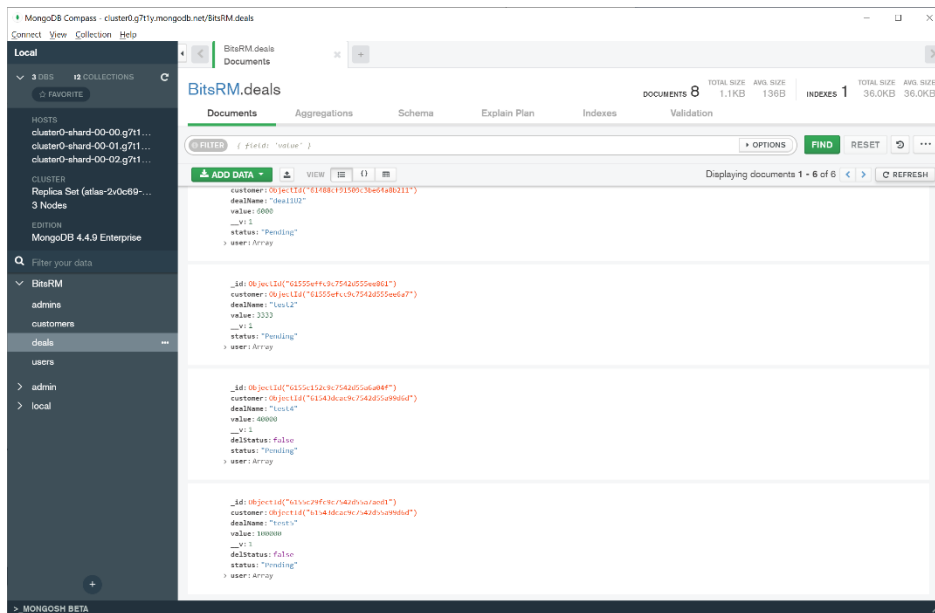


- Admin delete deals flagged for deletion

## Back End

Precondition: Successfully authorized, login in (have a valid jwt token) and deal ID to delete

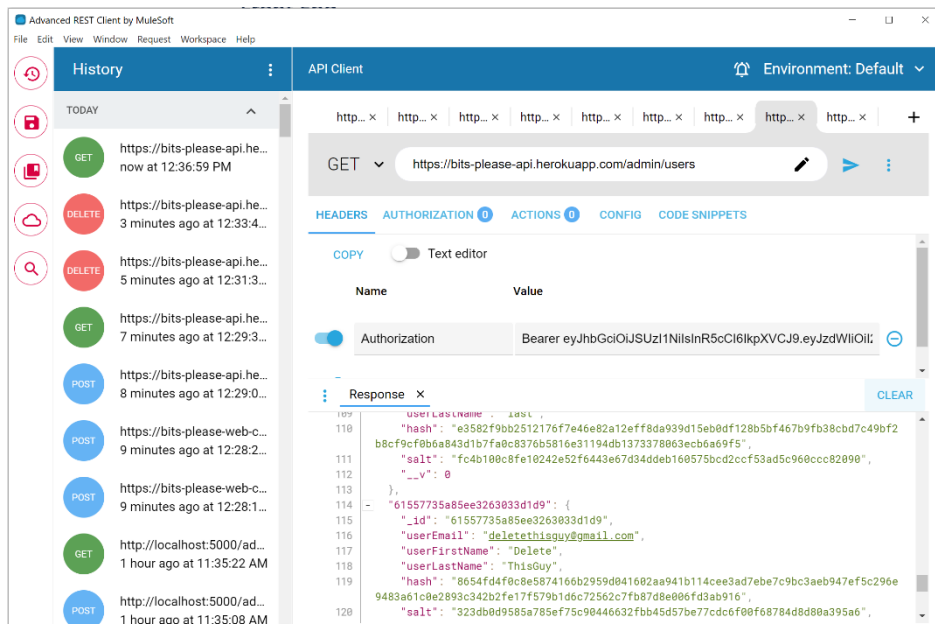


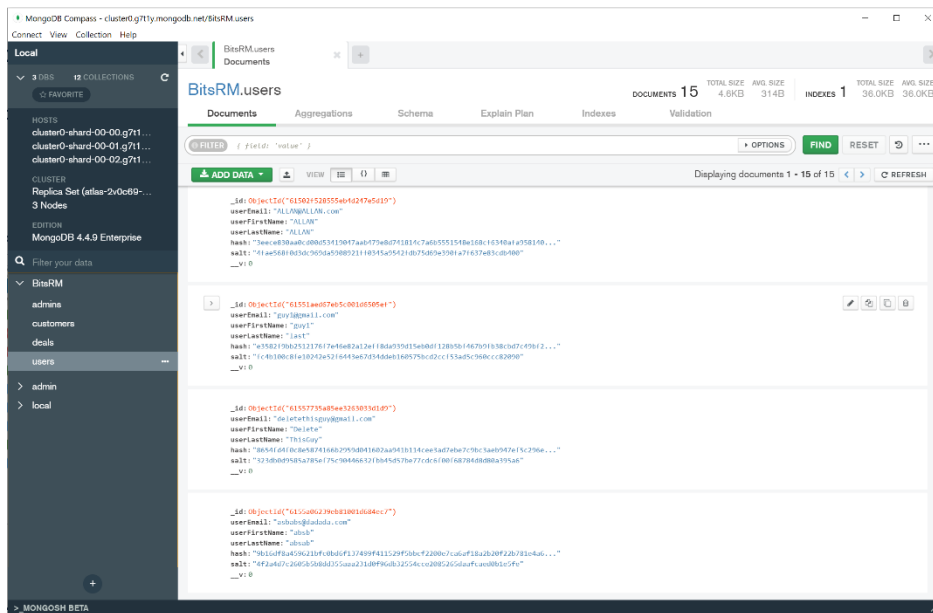


- Admin get list of user accounts

## Back End

Precondition: Successfully authorized, login in (have a valid jwt token)

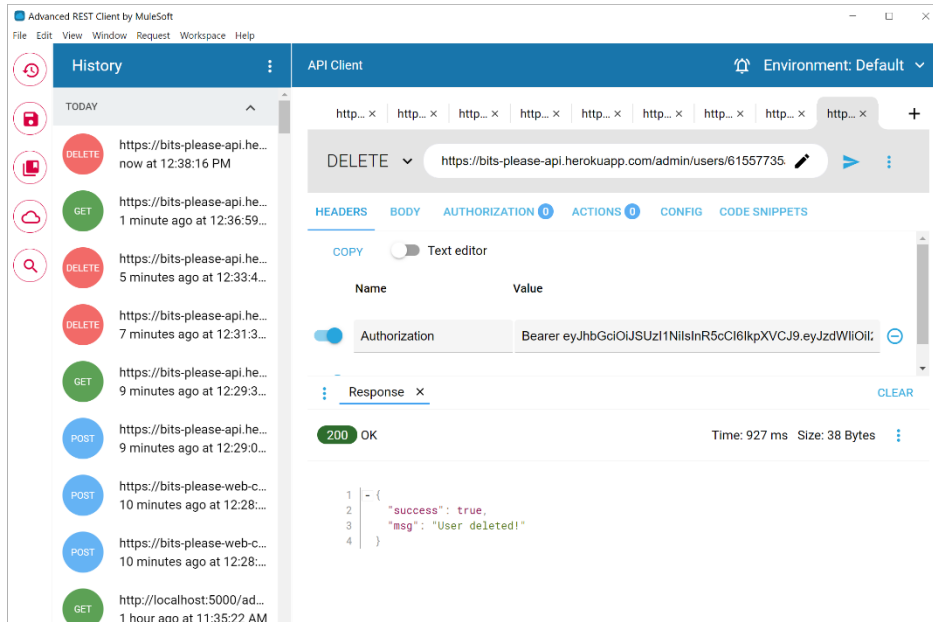




- Admin delete user account

## Back End

Precondition: Successfully authorized, login in (have a valid jwt token), and a user ID to delete







# Pair Programming Meeting

Date 15-9-2021

## Overview

<b>Attendees</b>	Edward, Hans, Jackey, Jawad, Mia
<b>Time</b>	3:30 pm - 4:30 pm

## Meeting agenda

- Review backend code for deals and customer api
- hotfix to solve the network error login
- discuss potential solution to the state in frontend
- Documentation of API on postman

## Discussions

1. Backend code for deals and customer looks good now
2. Solved the network error with deployed login page
3. Use Redux to maintain state on frontend
4. Further troubleshooting