ByteLog

TESTING REPORT

By Bytes Team

Friday, 24 March 2023

REPORT SUMMARY

In this report, there will be information on the approach to testing that was decided by the Bytes team during the development process. The team utilised a combination of automated testing using the Jest testing library and React Native testing library alongside manual tests done in weekly meetings. Further details regarding the automated tests and manual tests applied to the BytesLog mobile application are listed below.

TABLE OF CONTENT

APPROACH TO TESTING

Our team employed two approaches for testing:

- Automated testing
    - Jest and the React Native testing library
    - Automated testing using Github Action workflow
- Manual testing

We used unit testing for the main components of our app, however, we learned quickly that there were many important aspects that unit testing cannot cover. Hence, we made the decision to employ manual testing to complement our automated tests. Manual testing also provided us the benefit of testing the visual differences across the two platforms that we are deploying the app on, IOS and Android.

Automated testing

We experimented with a variety of testing frameworks i.e. Jest, Ava and Sinon. Our group decided that Jest was most appropriate for our app, due to its abundance of documentation available. We used a combination of Jest and the React Native testing library for backend and frontend testing.

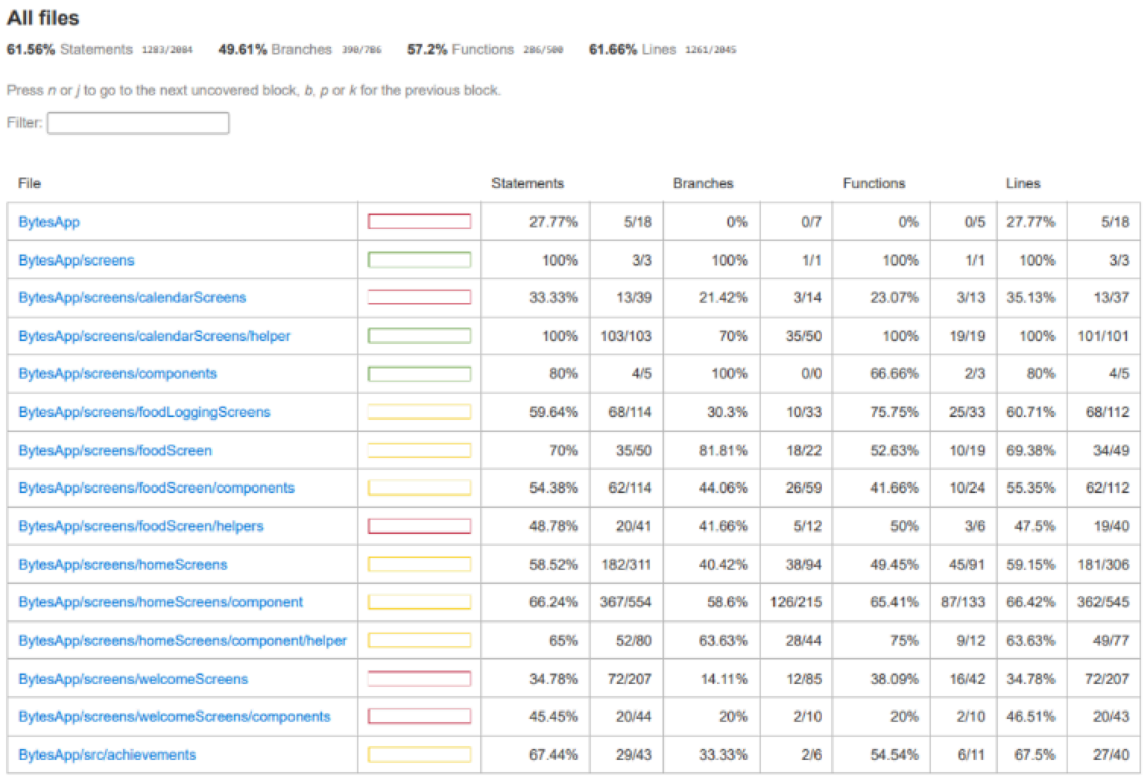
Another reason why we settled on Jest was because it provides Snapshot testing that is a useful frontend testing tool to ensure that our UI did not change unexpectedly.

Jest testing structure

Each file and component has a test file associated with it. The test file will contain the test suite for the component and any other helper functions associated with it. To ensure new additions to the code base are functional, we implemented the Github Actions workflow to run tests automatically whenever there is a new commit.

Automated test suite evaluation

Below is a HTML code coverage report produced by Jest.



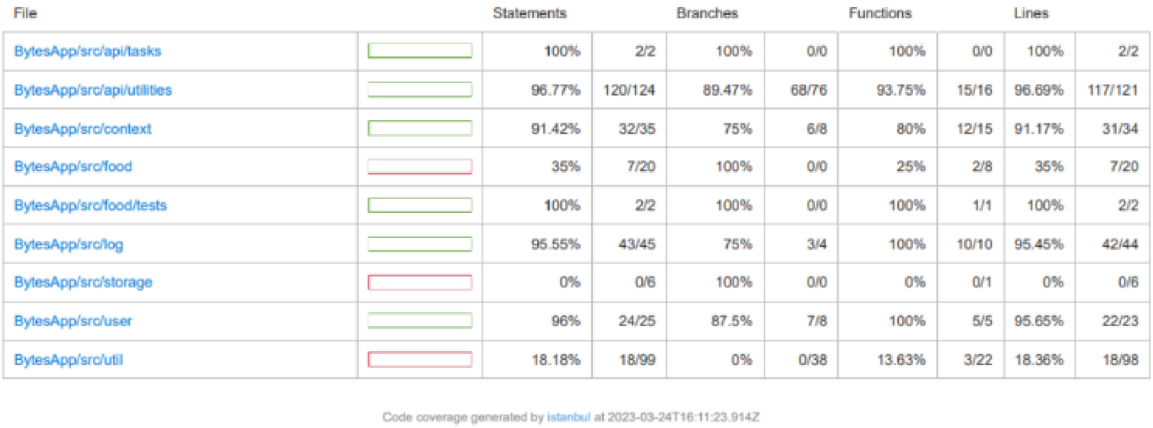
All files

61.56% Statements 1283/2084 **49.61%** Branches 390/786 **57.2%** Functions 286/500 **61.66%** Lines 1261/2045

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File		Statements		Branches		Functions		Lines	
BytesApp	🟥	27.77%	5/18	0%	0/7	0%	0/5	27.77%	5/18
BytesApp/screens	🟩	100%	3/3	100%	1/1	100%	1/1	100%	3/3
BytesApp/screens/calendarScreens	🟥	33.33%	13/39	21.42%	3/14	23.07%	3/13	35.13%	13/37
BytesApp/screens/calendarScreens/helper	🟩	100%	103/103	70%	35/50	100%	19/19	100%	101/101
BytesApp/screens/components	🟩	80%	4/5	100%	0/0	66.66%	2/3	80%	4/5
BytesApp/screens/foodLoggingScreens	🟨	59.64%	68/114	30.3%	10/33	75.75%	25/33	60.71%	68/112
BytesApp/screens/foodScreen	🟨	70%	35/50	81.81%	18/22	52.63%	10/19	69.38%	34/49
BytesApp/screens/foodScreen/components	🟨	54.38%	62/114	44.06%	26/59	41.66%	10/24	55.35%	62/112
BytesApp/screens/foodScreen/helpers	🟥	48.78%	20/41	41.66%	5/12	50%	3/6	47.5%	19/40
BytesApp/screens/homeScreens	🟨	58.52%	182/311	40.42%	38/94	49.45%	45/91	59.15%	181/306
BytesApp/screens/homeScreens/component	🟨	66.24%	367/554	58.6%	126/215	65.41%	87/133	66.42%	362/545
BytesApp/screens/homeScreens/component/helper	🟨	65%	52/80	63.63%	28/44	75%	9/12	63.63%	49/77
BytesApp/screens/welcomeScreens	🟥	34.78%	72/207	14.11%	12/85	38.09%	16/42	34.78%	72/207
BytesApp/screens/welcomeScreens/components	🟥	45.45%	20/44	20%	2/10	20%	2/10	46.51%	20/43
BytesApp/src/achievements	🟨	67.44%	29/43	33.33%	2/6	54.54%	6/11	67.5%	27/40



File		Statements		Branches		Functions		Lines	
BytesApp/src/api/tasks	🟩	100%	2/2	100%	0/0	100%	0/0	100%	2/2
BytesApp/src/api/utilities	🟩	96.77%	120/124	89.47%	68/76	93.75%	15/16	96.69%	117/121
BytesApp/src/context	🟩	91.42%	32/35	75%	6/8	80%	12/15	91.17%	31/34
BytesApp/src/food	🟥	35%	7/20	100%	0/0	25%	2/8	35%	7/20
BytesApp/src/food/tests	🟩	100%	2/2	100%	0/0	100%	1/1	100%	2/2
BytesApp/src/log	🟩	95.55%	43/45	75%	3/4	100%	10/10	95.45%	42/44
BytesApp/src/storage	🟥	0%	0/6	100%	0/0	0%	0/1	0%	0/6
BytesApp/src/user	🟩	96%	24/25	87.5%	7/8	100%	5/5	95.65%	22/23
BytesApp/src/util	🟥	18.18%	18/99	0%	0/38	13.63%	3/22	18.36%	18/98

Code coverage generated by istanbul at 2023-03-24T16:11:23.914Z

Manual testing

In most weeks of the development process, new features were being added so automated testing was not as efficient and some tests would become redundant as our code changed. At the start of each meeting, we would manually test the app as a team to ensure we had implemented functionality as intended. To provide structure to manual testing, we made a checklist of all the problems that we encountered in manual testing our app. We prioritised fixing these issues in the next sprint each time.

APPENDIX

Testing on	Input	Expected output
Quiz Pages (before sign in)		
Selecting a number of options greater than the max allowed for a quiz question is not allowed	Try to select multiple options for a question that is not allowed (e.g. 2 options for gender) and click the ‘next’ button	A warning message should display telling the user how many selections are allowed, and does not let the user proceed to the next question. The user is only allowed to go to the next question if the question has been answered correctly.
Quiz must be filled out completely for user to proceed	Try leaving a field empty in Quiz (e.g. text field for height) and click the ‘next’ button	A warning message should display telling the user to input an answer for the current question before proceeding.
Ensure next and back buttons do not go outside the range	Pressing the back button on the first page and next on the page with the last question.	The back button for the first page should be disabled and the next button on the last question should navigate to the Quiz Results screen only if the quiz is completed.
Calorie goal appears in Quiz result screen	Fill out all Quiz questions completely.	The calculated calorie goal should display on screen.
Signup and Login pages		
Keyboard scroll works properly	Click a text input box (e.g. email) from the sign up or log in form.	On both Android and IOS, the keyboard should pop up and users should still be allowed to scroll the screen, switch the text input box. Keyboard should disappear when clicking anywhere on the screen.
Home screen		
Time period buttons in Stats box shows statistics for specific time period selected	Select a time period button day, week, month, year	Should show stats for day, week, month, year accordingly.
Current progress has the correct days with logged data	Look at previously logged food and add the nutrient percentages	Nutrient percentages should be consistent.
Page lets you scroll to the bottom/end of the page’s content	Scroll as far down as possible	The scroll should not stop midway.
Update profile	Navigate to the profile	Successfully display the newly

picture	screen and change profile picture	uploaded profile picture visually in section one of the home screen. Check Firestore Storage to see that the photo file is uploaded and stored.
Achievement box shows top three of the user's achievements with most progress	Look at the achievement section at the bottom of the home page	Display the top three achievements (description and progress indication) that the user has most progress on. If no progress is made, show the first achievements on top of the list.
Calendar screen		
New addition of meal indicates on Calendar	Adding a meal from the log food screen	Green dot should appear on Calendar under today's date to indicate a log.
Front and back arrows on Calendar	Click the back and front buttons on Calendar	For both Android and IOS, Calendar view should successfully render previous or next month.
Month dropdown renders Calendar correctly based on month selected	Select a month on month dropdown	Calendar changes accordingly to display the page of the selected month.
Year dropdown renders Calendar correctly based on year selected	Select a year on year dropdown	Calendar changes accordingly to display the page of the selected year.
Time period buttons in Stats box shows statistics for specific time period selected	Select a time period button day, week, month, year	Should show stats for day, week, month, year accordingly.
New addition of meal changes macronutrient details on Stat box	Adding a meal from the log food screen	Macronutrient percentages should update on the stats box with the new intake additions.
Daily log screen		
Shows a break down of each macronutrient amount from all of today's log	Click anywhere in the Analytics Stats box to be redirected to the Daily log screen	The list of each macronutrient amount and intake for today is displayed. In the Firestore database, check the specific document in the DailyLog for a list of meals, and for each meal item, sum nutrients from the documents in the MealLog collection to check that correct amounts are shown.
Back button	Click the 'back' button on	The user should be redirected back

functionality works	the Daily log screen	to the Calendar screen.
Screen is scrollable	Try scrolling on the Daily log screen for a long list of logs	Scroll should work properly for both Android and IOS.
Search food/recipe page		
Add own food	Click ‘Add your own’ button and after entering food name and nutrient details, click ‘Create’	On page refresh, a newly added food item is added to the food list with a fire and person symbol indication. Check the FoodItem collection in the Firestore database for a new entry recording all added details of the food item.
Add own recipe	Click ‘Add your own’ button and after entering recipe name, nutrient details and ingredients, click ‘Create’	On page refresh, a newly added recipe is added to the recipe list with a fire and person symbol indication. Check the Recipe collection in the Firestore database for a new entry recording all added details of the recipe.
Log food screen		
Add a new meal for the day for a given meal category	Click the ‘Add meal’ button, choose a meal category, add as many food/recipes as you want to the meal then click the ‘Save meal’ button	The new meal category that you have selected for the day would be added to the ‘Today’s meals’ list, and clicking into it would show the food/recipe added to that meal.
Add a new meal for the day with a custom category	Click the ‘Add meal’ button, create a new meal category through ‘Add category’ and then choose the newly defined meal category. Add as many food/recipes as you want to the meal then click the ‘Save meal’ button	The newly defined meal category would be added to the ‘Today’s meals’ list, and clicking into it would show the food/recipe added to that meal.
Add an food/recipe to existing meal	Click the ‘Add a food or recipe button’ on an existing created meal from ‘Today’s meals’ list. After selecting food/recipe to add, click ‘Save’	When clicking into the edited meal from ‘Today’s meals’ list, the new food/recipe should appear in the list.
Delete an existing meal from today’s log	Click the ‘Delete’ button on an existing meal in the ‘Today’s meals’ list. An confirmation alert should	The deleted meal should disappear from the list of Today’s meals. Check firestore database MealLog collection that the respective meal entry is

	display and click ‘OK’	deleted. Check the current user’s log today in the DailyLog collection that the meal is deleted from the array of meals.
Profile screen		
Set profile picture	Click the profile picture icon and select a picture to set	Successfully display profile picture visually in profile
Set achievement badge to show in profile	Under the badges section, click a badge out of the possible options	Selecting a badge would automatically update the badge of the profile. Check that badge change is saved by logging out and logging back in to see that badge selected is set in profile.
Number of badges show up correctly	Navigate to the profile screen	The correct number of badges should be displayed.
Edit profile saves changes	Go to Profile, and select the ‘settings’ icon at the top right. Enter in profile details to update and then click ‘Save’	Click the ‘settings’ icon again and see that changes are updated by the fields initialised with the new values set. Check the UserProfile collection in the Firestore database to see that changes are reflected for the specific user document.
User Food & Recipes section in Profile Screen		
Add new bookmarked food or recipe	Navigate to an unbookmarked item screen of a food or recipe and press the bookmark icon	The bookmark icon changed to a saved icon and the list of food/recipes is updated in real time when refreshing the page and navigating back to the bookmarked item.
Remove bookmarked food or recipe	Navigate to the item screen of a food or recipe and press the bookmark icon.	The bookmark icon changed to a not saved icon and the list of food or recipes is updated in real time when navigated to.
Achievement section in Profile screen		
Page lets you scroll to the bottom/end of the Achievement list	Click the Achievements header in Profile Screen. Scroll as far down as possible.	The scroll should not stop midway.
There is a progress indication for each achievement	Click the Achievements header in Profile Screen.	Check that each achievement box has a green bar that correctly visually indicates the progress for an achievement.