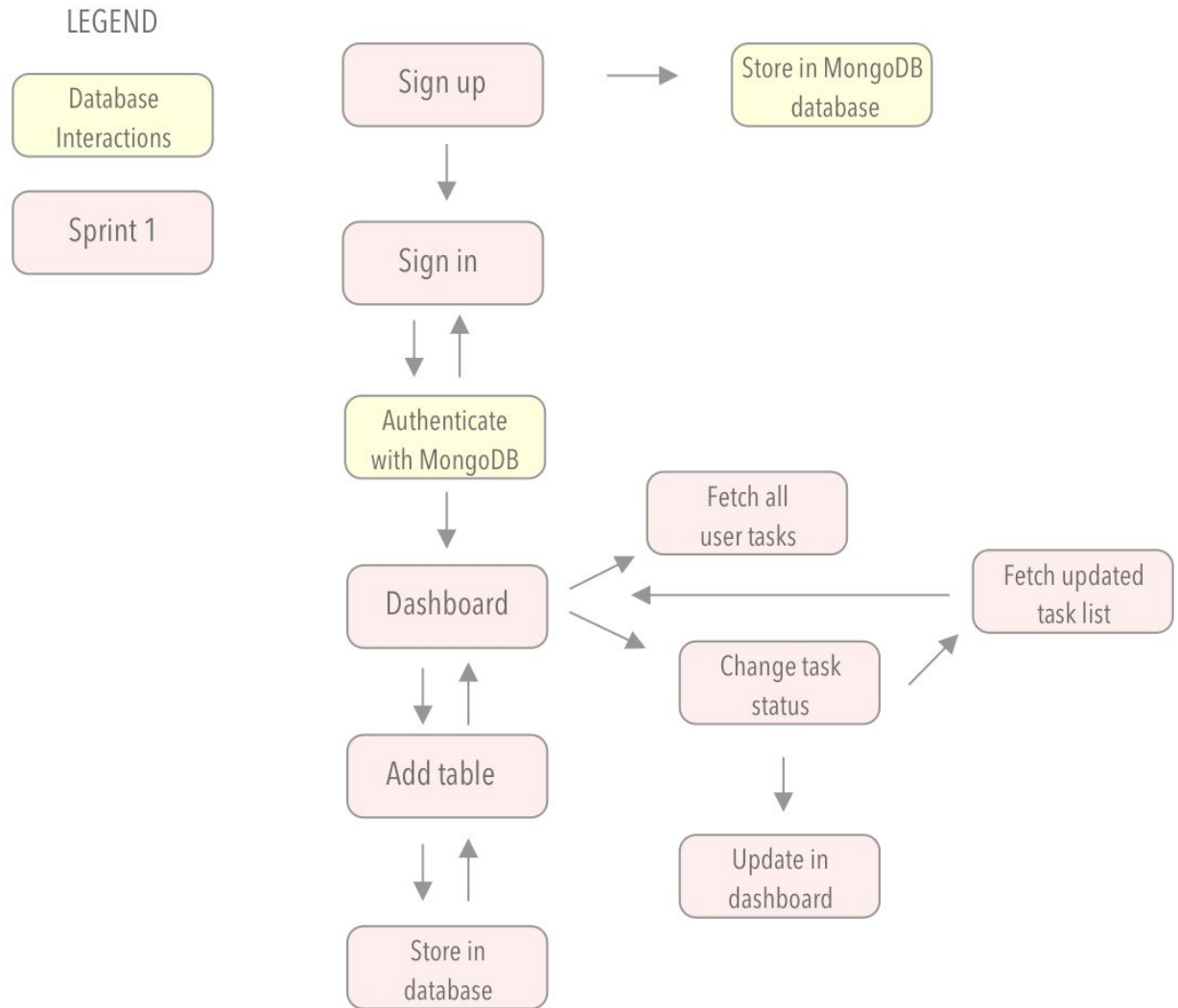


Sprint 1: Incremental Testing and Regression Testing

Arshad Alikhan, Ashwin Mahesh, Richa Tenany, Turner Dunlop, Vishva Pandya

1. Classification of Components

1.1 Define all components (diagram)



Login/Signup Module

- The Sign-up/Sign in module is responsible for creating a user account with a valid email and password
 - Sign up module prompts the user to enter a name, email and valid password
 - Sign in module validates the user's credentials to ensure an account exists
- A confirmation email is sent once the user has registered to confirm their account
- User is able to logout of the account

Dashboard Module

- The dashboard module allows the user to see their tasks that are in each category: to-do, in progress, and done
- The user is able to move their tasks successfully from each of the categories whenever they please

Task Module

- Tasks are displayed on the dashboard in cards in each category
- Add-Task Sub Module
 - The add task module is in charge of storing user's data on their task name, task weight, category, description, and deadline
 - The user is able to cancel the current task status while they are attempting to add it to the dashboard
 - Once a user is done entering the task information, they are able to click done which will successfully store the task details in the back-end
- Remove task sub module
 - There is a trash icon located in each card on the bottom corner to remove a task
 - If the user chooses to remove a task, the task is successfully removed from the board and from the database

1.2 Which form of incremental testing did you follow

We followed the top-down integration testing because a user is required to make an account in order to login and access their account. Login and Signup are critical modules because a user cannot access any other modules until they have created an account. Thus, the lower level modules like task module and dashboard module can only be accessed when the login/sign up module has been verified by our database and implemented correctly.

2. Incremental & Regression Testing

Module: Login/Sign-Up Module

Method: Manual

Incremental Testing

Defect No.	Description	Severity	How to correct
1	User could register without entering one of the fields	Critical	Added a prompt message to ensure the user fills out all the fields
2	Email is not validated if it's a real email	Workaround	Have a link in the email sent to them to redirect them to the homepage as confirmation

3	Email was not being sent through the smtp server once a user registers	workaround	Loosened security settings on email in order for smtp server to work.
---	--	------------	---

Regression Testing

Defect No.	Description	Severity	How to correct
1	User was allowed to log in with wrong credentials	Critical	Validated username and password with the database and provided an error message when credentials didn't match
2	No error message given if passwords were not the same while registering for an account	Critical	Hashed both the passwords entered by the user and compared the hashed code to ensure they were the same
3	User was able to enter a password that was an infinite amount of characters	Critical	Set a limit of 20 for the password length

Module: Task Module

Method: Manual

Incremental Testing

Defect No.	Description	Severity	How to correct
1	If a user leaves a field empty while adding a task, nothing happens	Workaround	Add checks to prompt the user to enter values in all the fields
2	If the user changes the status of a task, the page is reloaded each time	Workaround	Use sockets to refresh the page instead of reloading the page each time

3	The user can have an overload of tasks in the done category	Workaround	Set a limit to 10 tasks for the “done” section so the dashboard is not crowded with tasks
---	---	------------	---

Regression Testing

Defect No.	Description	Severity	How to correct
1	If a user logs out, the task disappears when they log back in	Critical	Added a username field to the database to match each task with a user to ensure their information saves
2	If the user removes a task, it is just hidden from the dashboard and not actually removed	Workaround	Added a function to remove the task from the database and stopped server calls to the specific task

Module: Dashboard Module

Method: Manual

Incremental Testing

Defect No.	Description	Severity	How to correct
1	When the user clicks done after adding a task, the task would not populate the dashboard	Critical	Added a spy on the done function and checked to make sure the “done” button was being clicked and called the function accordingly to populate the dashboard
2	When the user clicks cancel while adding a task, it would do nothing	Critical	Added a spy on the cancel function and checked to make sure the “cancel” button was being clicked and called the function accordingly
3	When the user removes a task, in order to remove it from the dashboard, the	Workaround	Used sockets to emit to the parent class to remove it from the database completely

	page must be refreshed		
4	The card would not show properly on the dashboard with all the elements	Workaround	Queried the title and description of each task to ensure it matched and displayed it on each card accordingly
5	When you add a task, they may go to other sections first	Critical	When adding a task, only "to-do" section has the option to add a task and will always start in the to-do pane

Regression Testing

Defect No.	Description	Severity	How to correct
1	User was not able to see the change in status of their tasks	Critical	Reload the window so that all three dashboard sections make the server fetch call again.