

Gyroscopic Project Final Design Documentation

COIS 4000Y

2022/04/25

Team 2

Steven Nash

Jared Corman

Tharmarajah Saranyan

Mohammed Ayaan Shirazi

In this document we will take a look at how the Design of the project has progressed and/or changed throughout the development process of the project.

When comparing the previous design document to this newly updated one, we can see some changes in the following sections:

- Business events and their inputs/outputs
- User Stories
- Mock-ups
- Open Project issues
- Break down of Completed vs incomplete tasks

1) Outline of the purpose of the project. List of key stakeholders

Gyroscopic internal collaboration tool is a remote retrospective tool. It simplifies the collection of feedback from team members and facilitates collaboration and communication. Gyroscopic internal collaboration makes it easy to maintain an asynchronous workflow. It allows users to assign tasks to other team members, add team members to projects, and monitor deadlines. It is very useful as a to-do list or calendar for strategic planning.

- Gyroscopic's co-founders (Waun Broderrick, Chris Beetham, Aiden Potts)
- Sprint Team Leaders
- Sprint Team Members / Developers

2) Key constraints, usually about running environment, time etc.

Solution Constraints

The below solution constraints have been tackled successfully:

- Boards as well as comments must update in real time: We were able to make sure that all changes made in our project such as editing a board, adding a comment, adding a team member and so on are all done in real-time
- Board members must be able to join via with a web link (website): We were successful in allowing our users to be able to send out an invite link to allow other users to be able to view their boards without necessarily having to be logged in

Time Constraints

The below time constraints were dealt with successfully:

We were able to meet the following deadlines and submit the required deliverables on time to the concerned receivers (clients + Sri):

- October 24 - Complete requirements document. **(Completed)**
- November 14 - Initial project design document. **(Completed)**
- November 26 - First Presentation: each team will deliver a 20-minute presentation. The presentation will include descriptions of your client, your team members, your project, and your project alpha. Alpha release is an early prototype that is not feature complete. **(Completed)**

- December 3- Client satisfaction assessment. Your client will grade you based on their satisfaction with your project. Usually, the client's grade reflects the quality of your meetings with the client, and the quality of your work as assessed by the client. **(Completed)**
- February 6 - complete project design. **(Completed)**
- March 4 – Beta release due (second prototype) – the beta release should be mostly feature complete but lacking polishing and will likely have many bugs. It may also be lacking complete content or data. **(Completed)**
- March - Testing: Each team will present the testing report to the class. **(Completed)**
- April (end of the semester) – Second client satisfaction assessment. **(In progress)**
- April (after classes end) - Final Project Presentation: **(Completed)**

Logistical Constraints

The Logical Constraints were dealt with successfully:

We must ensure the highest standard security and privacy measures. Integrity and privacy of user's information should be protected.

- We were able to tie in a strong authentication system into our code to make sure that all our sensitive fields were encrypted and not made easy to access.

3) Terminology or relevant facts which might not be obvious that the graders

Client: Gyroscopic

User: all users are “*board members*” users are not always registered, some users are board members in addition to another or multiple other position (board creator, team creator, team member)

Board: a board is a named group of columns, additionally each column has its own name. where users can add/edit/delete/move anonymous comments throughout the set of columns. Some boards are accessible only through team links, others are accessible via any link and the board creator has advanced ‘features’ within the scope of the board.

Logged in User: a logged in user must have registered their account and is actively logged in.

Board creator: Users may create a board when someone creates a board they become that board's creator with access to additional features within the scope of the board.

Team creator: A user may create a team and create links for other Users to join the team as team members.

Team members: Team members are logged in users that were invited to a team.

Sprints: Gyroscopic uses a development method where they cram two weeks' work together, complete it, then complete a sprint review process before planning and beginning another 2 weeks of development.

Database diagrams may be helpful for visualizing interactions within the project.

Database screenshots of various different project tables are seen below:

Database of Boards:

The boards database consists of fields such as id, name, date, description, column type, teams, columns, members and owner.

- The id field stores the unique board id
- The name field stores the Board name as a string
- The date field stores the date and time the board was created
- The description field stores the description of the board as a string
- The column type field stores whether the user opted for a default column style or a custom column style
- The members field stores the members that are a part of this board as an Array
- The teams field stores which team the board is assigned to
- The columns field stores the columns inside a board as an Array
- The owner field stores the id of the user that owns or has created the board

The screenshot shows the MongoDB Atlas interface for the 'gyroscopic_1.boards' collection. The interface displays two documents in a JSON format. The first document has fields: _id, name, date, description, columntype, columns, members, owner, and __v. The second document has fields: _id, name, date, description, columntype, team, columns, owner, and __v. The interface also shows a filter bar, a view selector, and a refresh button.

```
{
  "_id": "ObjectId('624592f3693b5202f993e733')",
  "name": "Test",
  "date": "2022-03-31T11:39:31.144+00:00",
  "description": "This is a new test board",
  "columntype": "default",
  "columns": Array,
  "members": Array,
  "owner": "ObjectId('62457340693b5202f993e678')",
  "__v": 4
}
```

```
{
  "_id": "ObjectId('62610fcf72991eaed1f6f3de')",
  "name": "New Board",
  "date": "2022-04-21T08:03:27.800+00:00",
  "description": "This is a new board",
  "columntype": "custom",
  "team": null,
  "columns": Array,
  "owner": "ObjectId('62457340693b5202f993e678')",
  "__v": 4
}
```

Database of Columns in a Board:

The columns database consists of fields such as id, header, column order and comments.

- The id field stores the unique column id
- The header field stores the name that is assigned to the column as a string
- The column order field stores the order type of the specific column
- The comments field stores the comments inside a column as an Array

gyroscopic_1.columns

DOCUMENTS 26 STORAGE SIZE 20.5KB AVG. SIZE 95B INDEXES 1 TOTAL SIZE 36.9KB AVG. SIZE 36.9KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ↺ ⋮

ADD DATA VIEW [List View Icon] [Grid View Icon] [Table View Icon]

Displaying documents 1 - 20 of 26 < > C REFRESH

```
{
  "_id": ObjectId("61f22af3dde056030d85826c"),
  "header": "Went Well",
  "columnOrder": 1,
  "> comments": Array
  __v: 5
}
```

```
{
  "_id": ObjectId("61f22af3dde056030d858271"),
  "header": "To Improve",
  "columnOrder": 2,
  "> comments": Array
  __v: 5
}
```

```
{
  "_id": ObjectId("61f22af3dde056030d858276"),
  "header": "Actions",
  "columnOrder": 3,
  "> comments": Array
  __v: 2
}
```

```
{
  "_id": ObjectId("61f7458df609aea921cd9574"),
  "header": "Went Well",
  "columnOrder": 1,
  "> comments": Array
  __v: 0
}
```

Database of Comments in a Column:

The comments database consists of fields such as id and content

- The id field stores the unique comment id
- The content field stores the content inside the comment as a string

The screenshot shows the MongoDB Compass interface for the 'gyroscopic_1.comments' collection. The top bar indicates 17 documents, 20.5KB storage size, 58B average size, and 1 index. The interface includes tabs for Documents, Aggregations, Schema, Explain Plan, Indexes, and Validation. A filter bar shows the filter '{ field: 'value' }'. The document list displays the following data:

Document
<pre>{ "_id": ObjectId('61f22b00dde056030d85827f'), "content": "random", "__v": 0 }</pre>
<pre>{ "_id": ObjectId('61f22b4adde056030d8582ae'), "content": "Random", "__v": 0 }</pre>
<pre>{ "_id": ObjectId('61fbb7fd22cf8f7c6b275e04'), "content": "Task1", "__v": 0 }</pre>
<pre>{ "_id": ObjectId('61fbb80422cf8f7c6b275e09'), "content": "Task 2", "__v": 0 }</pre>
<pre>{ "_id": ObjectId('61fbb80822cf8f7c6b275e0e'), "content": "Task 3", "__v": 0 }</pre>

Database of Upvotes and Downvotes:

The reacts (upvote/downvote) database consists of fields such as id, comment id, user id, dislike, like.

- The id field stores the unique react id
- The comment id field stores the id of the comment the react is attached to
- The dislike field stores whether or not the user has chosen the downvote option as a boolean
- The like field stores whether or not the user has chosen the upvote option as a boolean

The screenshot shows the MongoDB Compass interface for the 'gyroscopic_1.reacts' collection. The top bar indicates 5 documents, 20.5KB storage size, 125B average size, and 1 index. The interface includes tabs for Documents, Aggregations, Schema, Explain Plan, Indexes, and Validation. A filter bar at the top shows a filter on 'field: 'value''. Below the filter bar, there are buttons for ADD DATA, VIEW, and a refresh button. The main area displays a list of 5 documents, each with the following structure:

```
{
  "_id": ObjectId("624f340180134e7e211e3899"),
  "commentId": "624f340180134e7e211e3899",
  "userId": "62457340693b5202f993e678",
  "_v": 0,
  "dislike": true,
  "like": false
}
```

The documents are displayed in a list view, and the interface shows 'Displaying documents 1 - 5 of 5'.

Database of Teams:

The teams database consists of fields such as id, name, description, members and owner.

- The id field stores the unique team id
- The name field stores the team name as a string
- The description field stores the description of the team as a string
- The members field stores the members that are a part of this team as an Array
- The owner field stores the id of the user that owns or has created the board

The screenshot shows the MongoDB Compass interface for the 'gyroscopic_1.teams' collection. The interface includes a sidebar with the collection name, a top bar with document and index statistics, and a main area displaying a list of documents. The documents are as follows:

Document	_id	name	description	members	owner	__v
1	ObjectId('6260f24a72991eaed1f6f26b')	Team Error	Created in Error account	Array	ObjectId('624f31c6059449febca2090e')	0
2	ObjectId('6265d28723067d0c52fd5fdf')	Team 2	Random	Array	ObjectId('62457340693b5202f993e678')	2
3	ObjectId('6265dd6923067d0c52fd622b')	Team 1	Random	Array	ObjectId('62457340693b5202f993e678')	0

Database of Users:

The users database consists of fields such as id, username, email, salt, hash.

- The id field stores the unique user id
- The username field stores the username that the user chooses as a string
- The email field stores the email that the user signs up with as a string
- The salt field stores the salted password
- The hash field stores the hashed password

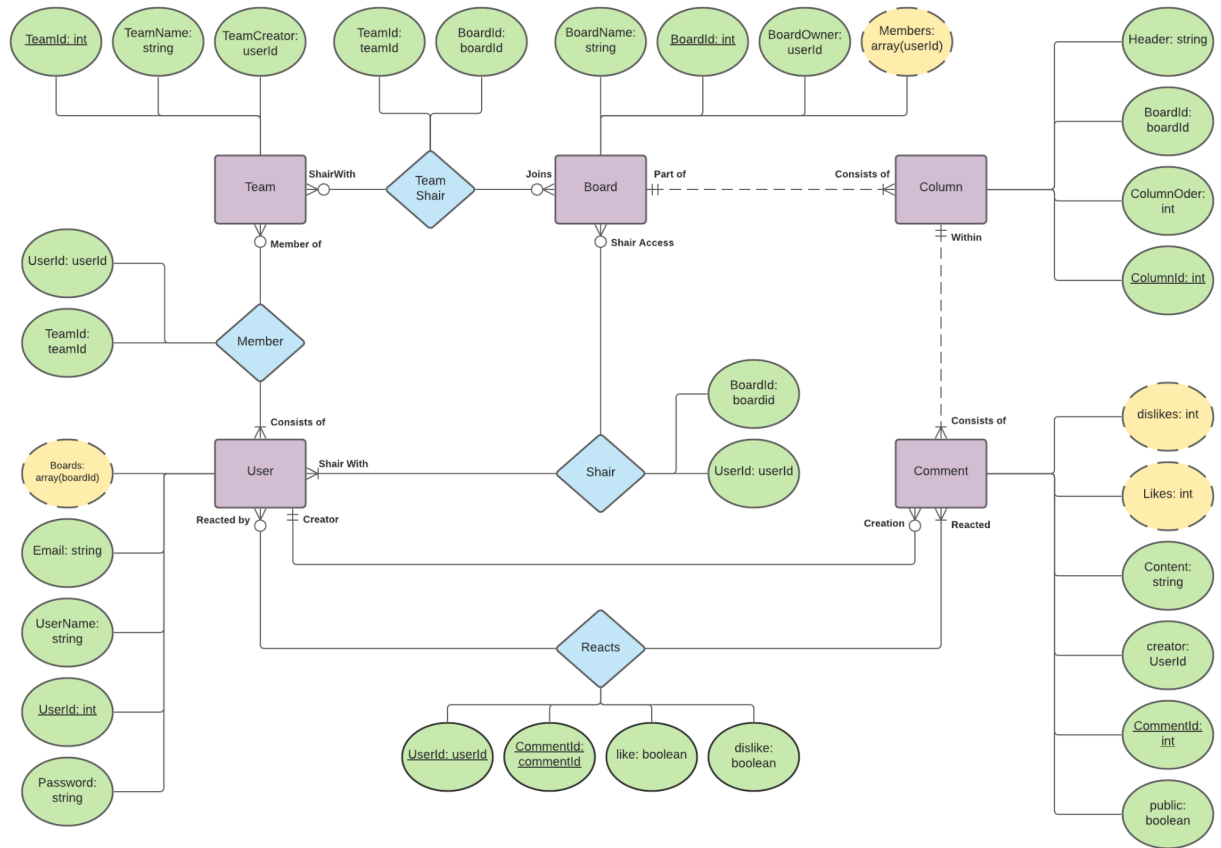
The screenshot shows the MongoDB Compass interface for the 'gyroscopic_1.users' collection. The top bar indicates 9 documents, 28.7KB storage size, 1.1KB average size, 2 indexes, 73.7KB total size, and 36.9KB average index size. The 'Documents' tab is active, showing a list of documents. The filter bar is empty, and the view is set to 'Table'. The documents displayed are:

```
{ "_id": "ObjectId('62457340693b5202f993e678')", "username": "Ayaan", "email": "test1@domain.com", "salt": "e4059834fb12cdf790814ff6b8a7d9e0789e34b617bc1e409bf9ebf92ef25c85", "hash": "2791884bf278ed000e70319f979a74d05b7ae0e2525c6f18734887ce36d50dd2d81e77...", "__v": 0 }
```

```
{ "_id": "ObjectId('624574f8693b5202f993e687')", "username": "RandomTest", "email": "test12@domain.com", "salt": "1afb6c3c1e08c4c2fcf7de16f00ec339bd7c2b0a72001b8e0ba6da339fc3a9e2", "hash": "3ab4fc099957e8bf11b30a66334c9bc0ee2ca2f4920e3d98clf340439a07e2066e1764...", "__v": 0 }
```

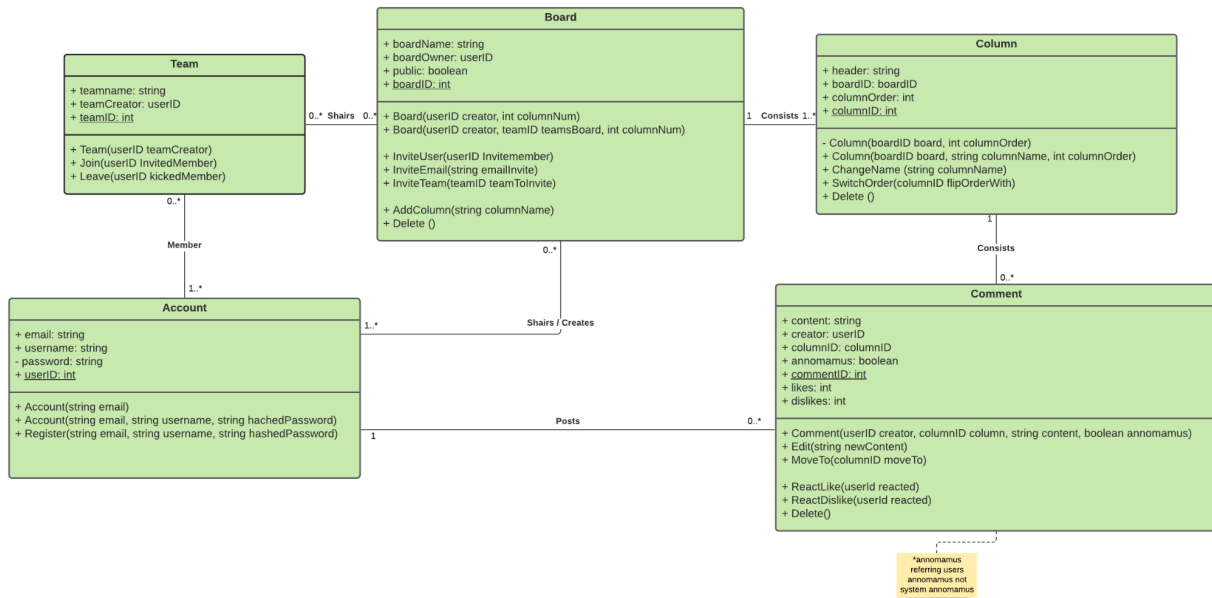
```
{ "_id": "ObjectId('6245787b693b5202f993e68f')", "username": "FinalTest", "email": "test3@domain.com", "salt": "61438af928f2936530db5052cd65455aee683a2aed649b6730cbdf8fc76a948e", "hash": "52d69a7de3672010df2b618498dd94c5cd43c8f1c845350ecf4c98a9481f457b21b064...", "__v": 0 }
```

```
{ "_id": "ObjectId('624580bd693b5202f993e6a1')", "username": "RandomTest1", "email": "test4@domain.com", "salt": "348b1831b24642bce6ae576941a0e358698beaeb5754c6eb13bab2206485cc74", "hash": "...", "__v": 0 }
```

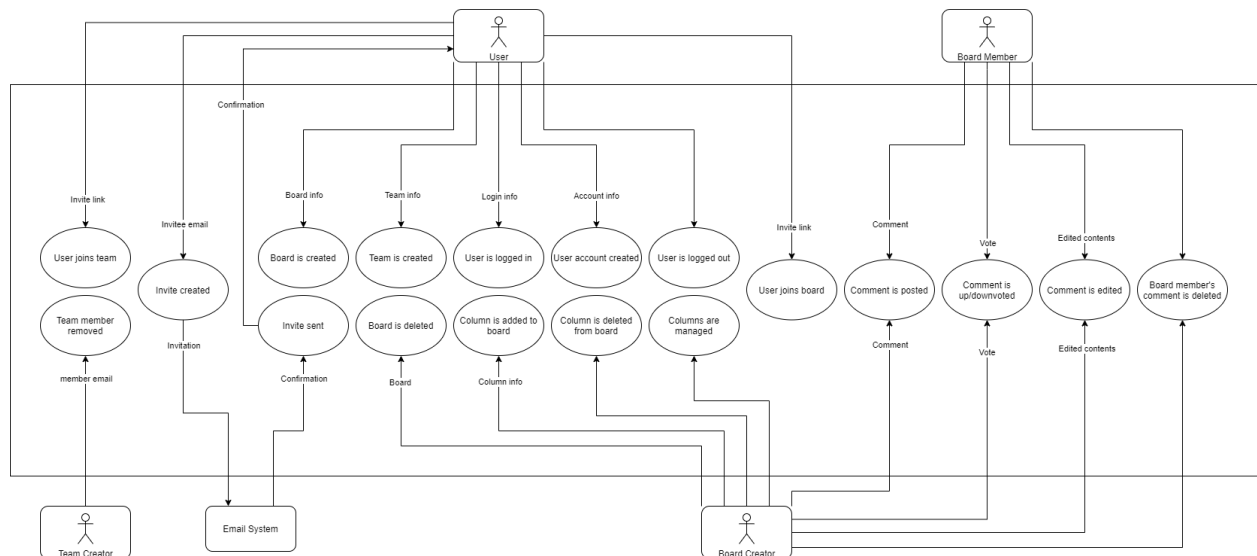


- 4) All functional requirements and hopefully some way to verify they are working correctly for testing. Key functional requirements could include user stories, and or UML diagrams.

Class diagram:



Product Use Case Diagram:



List the business events and their inputs/outputs:

Business Events we were not able to complete:

Event Name	Input and Output
User is able to manage account	User changes account details (in) Confirmation (Out)
Team creator remembers the board layout (custom column layout) from the previous board, so that I do not need to recreate the custom board layout that my team is using	List (out)

Business events that we were able to complete and accomplish:

Event Name	Input and Output
User creates an account	User account details (in) Confirmation (out)
User logs out and logs into the system	Login info(in)
User creates a team	Team info (in) Confirmation (out)
Board member creates a board	Board name (in) Sharable link (out)
Board member adds comments to a column	Comment (in)
Board member removes comments that were added to a column	User confirmation (in)

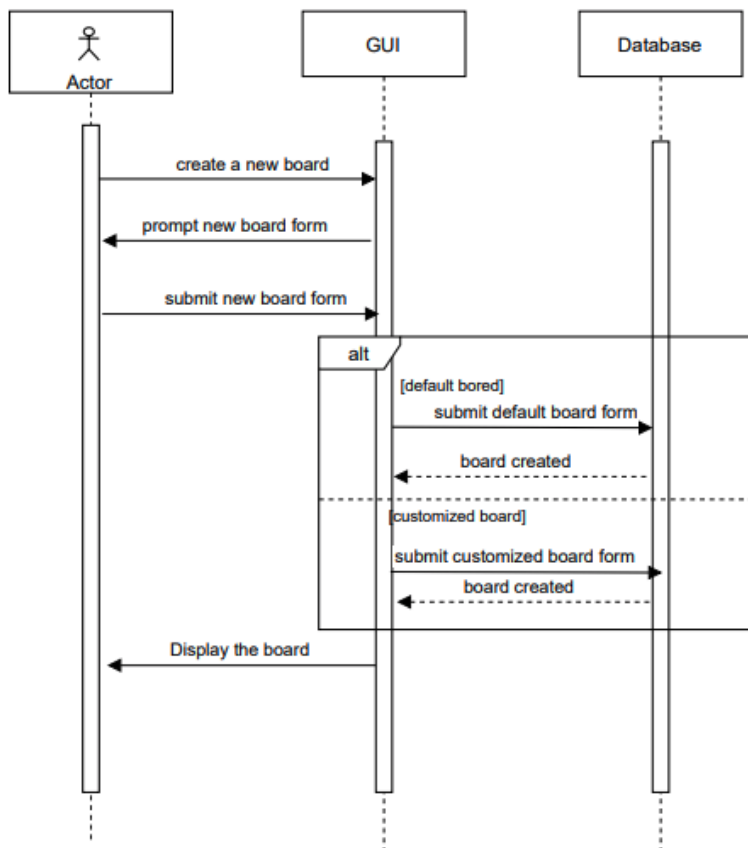
Board member sees the updates made to a board in real-time	Real time update (out)
Board member upvotes or downvotes a comment	User vote (in)
Board creator removes comments that anyone has added to a column	User confirmation (in)
Board creator adds additional columns with custom names	Column info (in)
Board creator removes columns	User's choice (in) User confirmation (in)
Board creator edits a column's name	Column info (in)
Board member moves my comments from one column to another	User's choice (in)
Board creator moves comments that anyone has added to a column	User's choice (in)
Board creator edits comments that anyone has added to a column	User's edit (in)
Board member edits my comments that I have added to a column	User's edit (in)
Team creator invites users via email to join my team (even if they are not yet registered)	Invitees' info (in) Confirmation (out)
Team creator adds a new board to my team	Board info (in) Confirmation (out)
Team creator removes a member from my team	User confirmation (in)

Team creator provides custom names to my team's boards	Names (in)
Team creator removes a board	User confirmation (in)
Logged in user sees a list of teams that I belong to	List (out)
Logged in user sees the list of boards belonging to my various teams	List (out)
Board member switches between my team's boards from the board view	

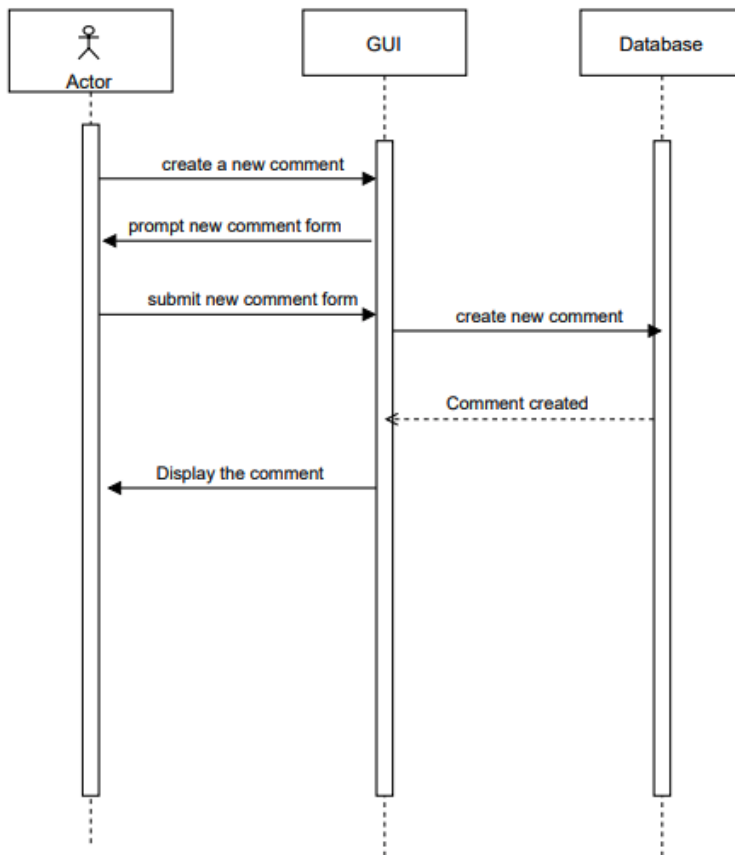
User Stories:

The below user stories were successfully incorporated into our project

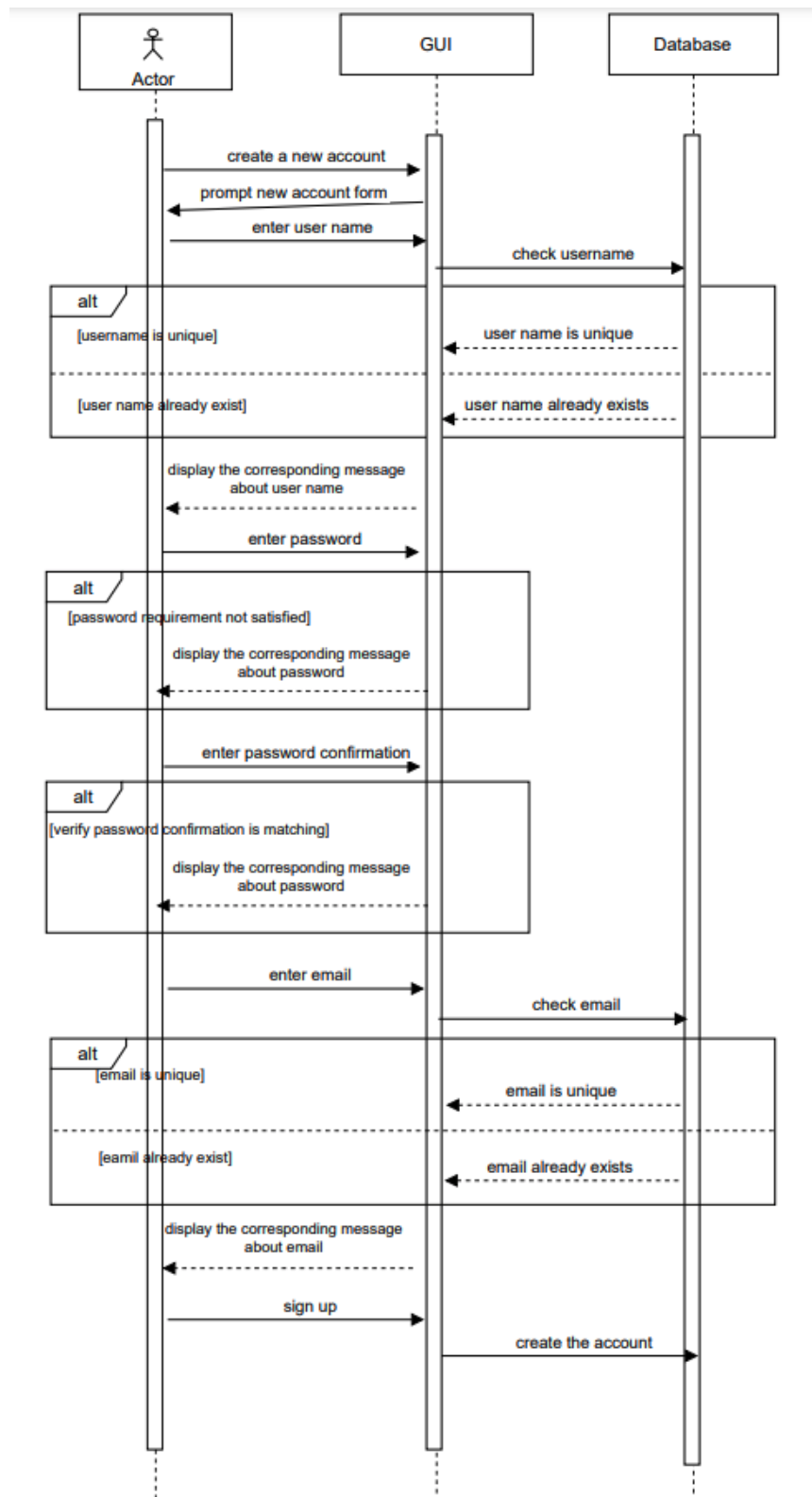
User want to create a new board:



User create a new comment :



User create a new account:



- 5) **A list of non-functional requirements. Non-functional requirements are ‘how does it do it well’. This includes look and feel, ease of use, performance (if any), maintainability, security, cultural, legal or privacy? Feel free to show mock-ups of things you are working on, also make sure to list if your non-functional requirements are implicit**

4.1 - Board updates in Realtime:

We were able to successfully implement the following ticket into our program.

All the changes made to our boards get updated in real time. This helps maintain continuity and reduces wait time for all users linked to the board.

Requirement #:	Requirement Type: Performance	Event /BUC/PUC #: All
Description: Board should update quickly		
Rationale: To reduce time user spends waiting		
Originator: Waun Broderick - Chief Technology Officer		
Fit Criterion: the board will update within a second 90% of the time		
Customer Satisfaction: 6		Customer Dissatisfaction: 9
Dependencies:		Conflicts: None
Supporting Materials:		
History: Created October 1st		

4.2 - Creating a board should be intuitive:

We were able to successfully implement the following ticket into our program.

Any user is able to simply click on “Create New Board” button to create a new board. During the creation of the board, the user also has an option to edit the name of the column and add extra columns or delete a column if they wish to do so. Columns can also be added or deleted after the creation of the board.

We were not able to implement the functionality to edit the name of the column once it has been created.

Requirement #:	Requirement Type: Usability	Event /BUC/PUC #: All
Description: Board creators should easily be able to create a board and edit columns.		
Rationale: For limited onboarding time		
Originator: Waun Broderick - Chief Technology Officer		
Fit Criterion: A board creator should be able to create a board, edit columns, Save template, delete the board, recreate from template within 30 minutes of their first attempt.		
Customer Satisfaction: 7		Customer Dissatisfaction: 3
Dependencies:		Conflicts: None
Supporting Materials:		
History: Created October 1st		

4.3 - Creating comments should be easy:

We were able to successfully implement the following ticket into our program.

Creating a comment inside a column is very easy as all the user has to do is type in their comment and choose whether or not to make the comment anonymous and simply click on send to post the comment. The user and the team owner can also delete comments very easily by clicking on the ‘x’ button on the top-right of the comment and edit the comments by double clicking on the comment.

Requirement #:	Requirement Type: Usability	Event /BUC/PUC #: All
Description: Board members should easily be able to add/edit/delete their own comments		
Rationale: For limited onboarding time		
Originator: Waun Broderick - Chief Technology Officer		
Fit Criterion: A board member shall be able to understand the product within 15 minutes without any out of product help.		
Customer Satisfaction: 7		Customer Dissatisfaction: 3
Dependencies:		Conflicts: None
Supporting Materials:		
History: Created October 1st		

4.4 - Simple User View:

We were able to successfully implement the following ticket into our program. The user will not be overwhelmed with the interface. There is a sufficient amount of whitespace on all pages to allow the user to focus on only what is important. All buttons and working of the application are pretty self explanatory and do not require a lot of previous knowledge to understand.

Requirement #:	Requirement Type: Usability	Event /BUC/PUC #: All
Description: The Logged in users interface should be easily understood.		
Rationale: For limited onboarding time		
Originator: Waun Broderick - Chief Technology Officer		
Fit Criterion: a logged in user should be able to understand the product within 15 minutes without any out of product help.		
Customer Satisfaction: 7	Customer Dissatisfaction: 3	
Dependencies:	Conflicts: None	
Supporting Materials:		
History: Created October 1st		

4.5 - Simple Team creator interface:

We were able to successfully implement the following ticket into our program. inviting a user to a board is very simple and all it takes is an email address and a simple click to send out the invitation. The team creator can also edit or delete a board with a simple click.

Requirement #:	Requirement Type: Usability	Event /BUC/PUC #: All
Description: The team creator should easily be able to invite/disinvite users and easily be able to create / delete team boards.		
Rationale: For limited onboarding time		
Originator: Waun Broderick - Chief Technology Officer		
Fit Criterion: A team creator should be able to create a team, add and remove additional users, and create a team board within 30 minutes of their first attempt.		
Customer Satisfaction: 5	Customer Dissatisfaction: 3	
Dependencies:	Conflicts: None	
Supporting Materials:		
History: Created October 1st		

4.6 - System security:

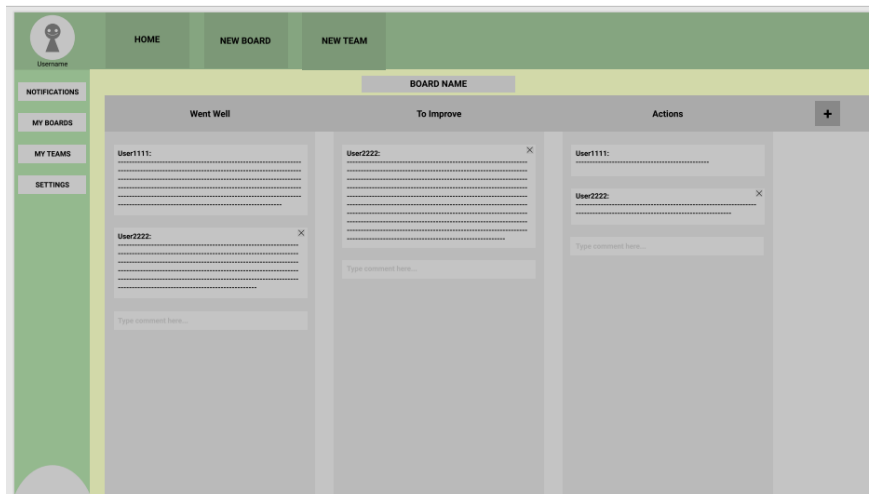
We were able to successfully implement the following ticket into our program.

If a user is not invited to a board they will not be able to view it. A user will only be able to view a board if they are part of the team the board is assigned to or else have been specially invited to view the board.

Requirement #:	Requirement Type: Security	Event /BUC/PUC #: All
Description: The product should ensure security from non-invited users. .		
Rationale: Maintain users trust in the product		
Originator: Waun Broderick - Chief Technology Officer		
Fit Criterion: A non-invited, non-user should not be able to access data through the product		
Customer Satisfaction: 2		Customer Dissatisfaction: 8
Dependencies: None		Conflicts: None
Supporting Materials: None		
History: Created October 1st		

Older Mock ups (*pre implementation, without CSS*)

Board view:



My Board:



My Team:



New Board:

Username

HOME

NEW BOARD

NEW TEAM

NOTIFICATIONS

MY BOARDS

MY TEAMS

SETTINGS

Enter board name...

Columns:

Went Well × To Improve × Actions × +

Invite members:

User1111 user1111@example.com ×

User2222 user2222@example.com ×

Enter email...

New Team:

Username

HOME

NEW BOARD

NEW TEAM

NOTIFICATIONS

MY BOARDS

MY TEAMS

SETTINGS

Enter team name...

Invite members:

User1111 user1111@example.com ×

User2222 user2222@example.com ×

Enter email...

Notifications:

Username

HOME

NEW BOARD

NEW TEAM

NOTIFICATIONS

MY BOARDS

MY TEAMS

SETTINGS

User1111 commented on Board 1 → ×

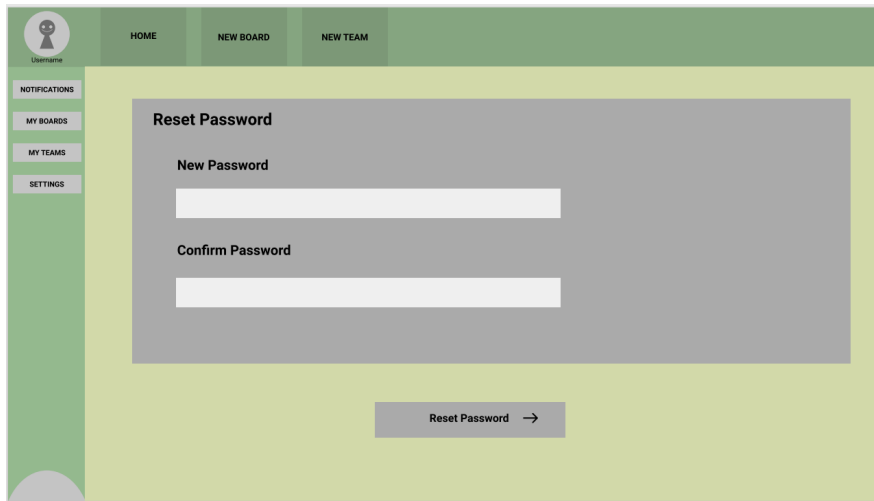
User3333 joined Board 2 → ×

User1111 joined Team 4 → ×

User1111 commented on Board 2 → ×

User4444 commented on Board 3 → ×

Reset password:



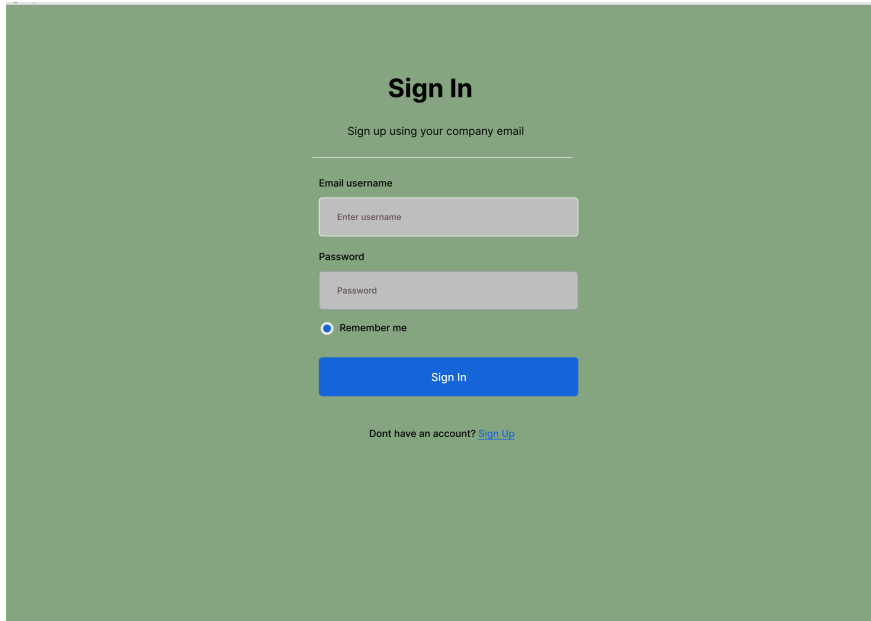
The image shows a web application interface for resetting a password. At the top, there is a navigation bar with a user profile icon and the text 'Username' on the left, and three menu items: 'HOME', 'NEW BOARD', and 'NEW TEAM'. Below the navigation bar is a sidebar with a list of options: 'NOTIFICATIONS', 'MY BOARDS', 'MY TEAMS', and 'SETTINGS'. The main content area is titled 'Reset Password' and contains two input fields: 'New Password' and 'Confirm Password'. Below these fields is a button labeled 'Reset Password →'.

Settings page:



The image shows a web application interface for the settings page. At the top, there is a navigation bar with a user profile icon and the text 'Username' on the left, and three menu items: 'HOME', 'NEW BOARD', and 'NEW TEAM'. Below the navigation bar is a sidebar with a list of options: 'NOTIFICATIONS', 'MY BOARDS', 'MY TEAMS', and 'SETTINGS'. The main content area is titled 'Settings' and contains four sections: 'Notifications' with a toggle switch set to 'On', 'Reset Password →', 'Change Display Name →', and 'Change Profile Picture →'.

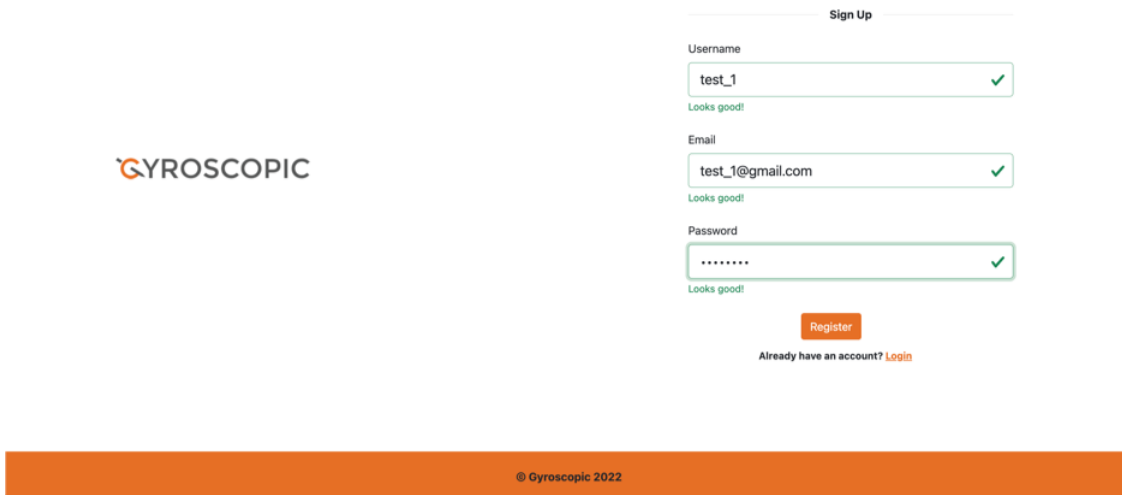
Sign In page:

A mockup of a 'Sign In' page with a green background. The title 'Sign In' is centered at the top. Below it, a subtitle reads 'Sign up using your company email'. There are two input fields: 'Email username' with a placeholder 'Enter username' and 'Password' with a placeholder 'Password'. Below the password field is a radio button labeled 'Remember me'. A blue 'Sign In' button is centered below the inputs. At the bottom, a link says 'Dont have an account? [Sign Up](#)'.

Non-mocks (current screenshots from the project)

Signup & Log In :

The user is prompted to enter their username, email and password for the sign up page and their username and password for the login page and all fields are validated accordingly.

A screenshot of a 'Sign Up' page. The title 'Sign Up' is at the top. There are three input fields: 'Username' with the value 'test_1', 'Email' with the value 'test_1@gmail.com', and 'Password' with masked characters '.....'. Each field has a green checkmark and the text 'Looks good!' below it. An orange 'Register' button is below the password field. At the bottom, a link says 'Already have an account? [Login](#)'. The Gyroscopic logo is on the left. The footer is an orange bar with '© Gyroscopic 2022'.

Forgot Password:

On this page the user can enter the email address linked with their account and a unique link will be sent to their email address to help them reset their password.

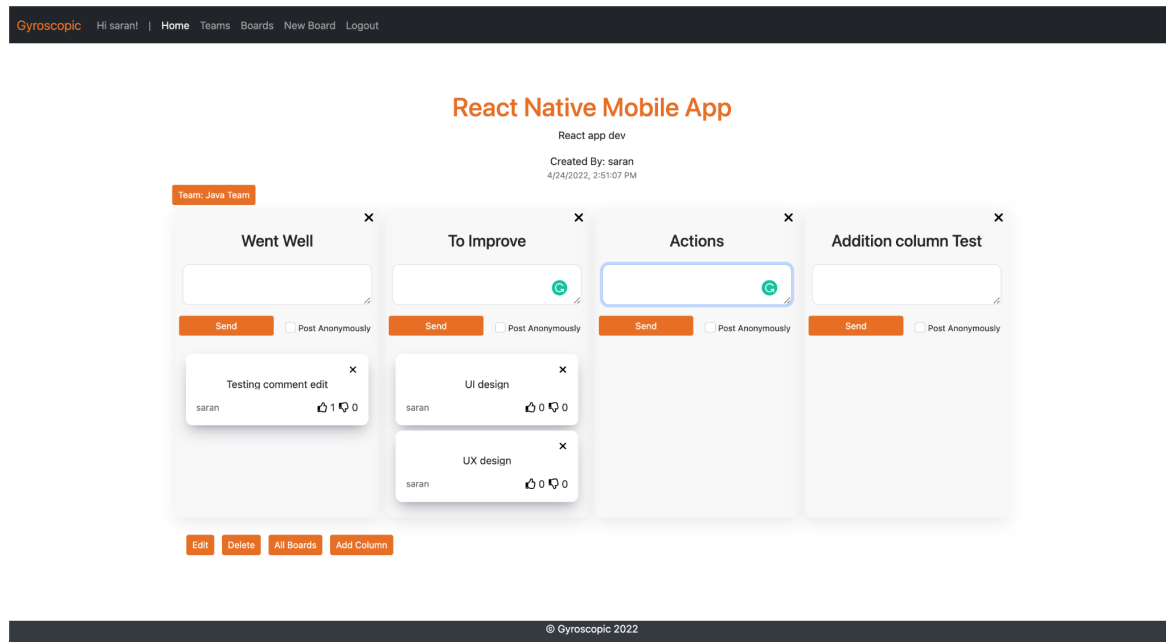


Email

Send Link

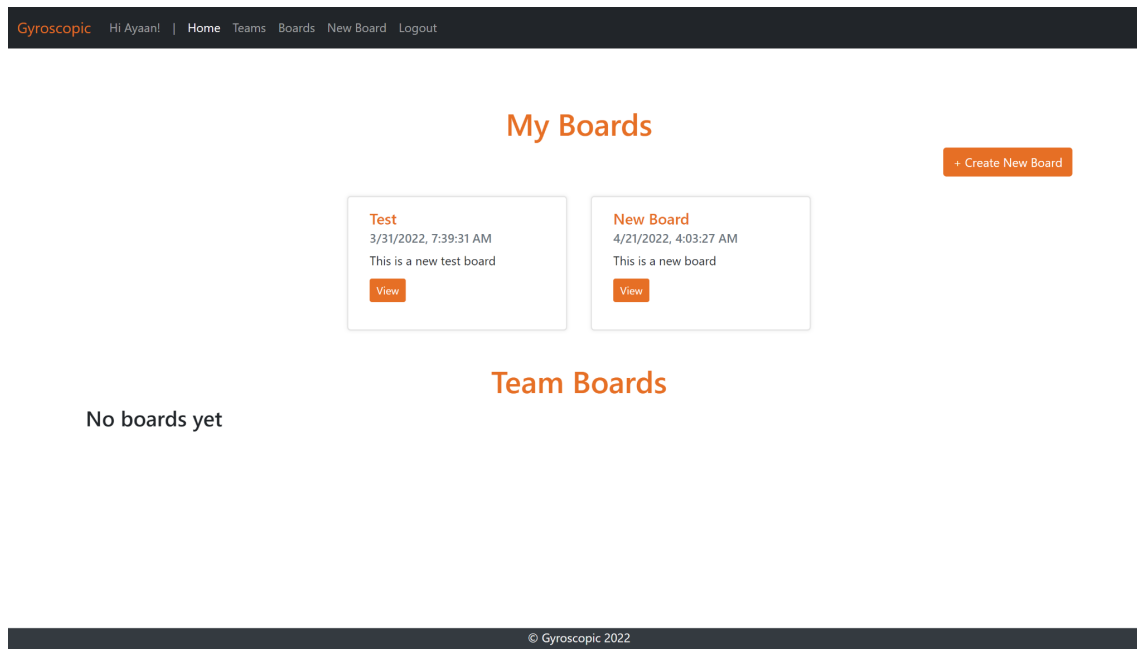
Board Page:

The user is able to view their board, delete columns, add comments and edit their board on this page.



My Boards Page:

The user on this page is able to view their boards as well as the their different team boards



New Board:

On this page the user can fill up the form to enter details about their new board

Gyroscopic

Hi Ayaan! | [Home](#) [Teams](#) [Boards](#) [New Board](#) [Logout](#)

New Board

Name

Description

Column Style

☐ Default

☐ Custom

Team:

None

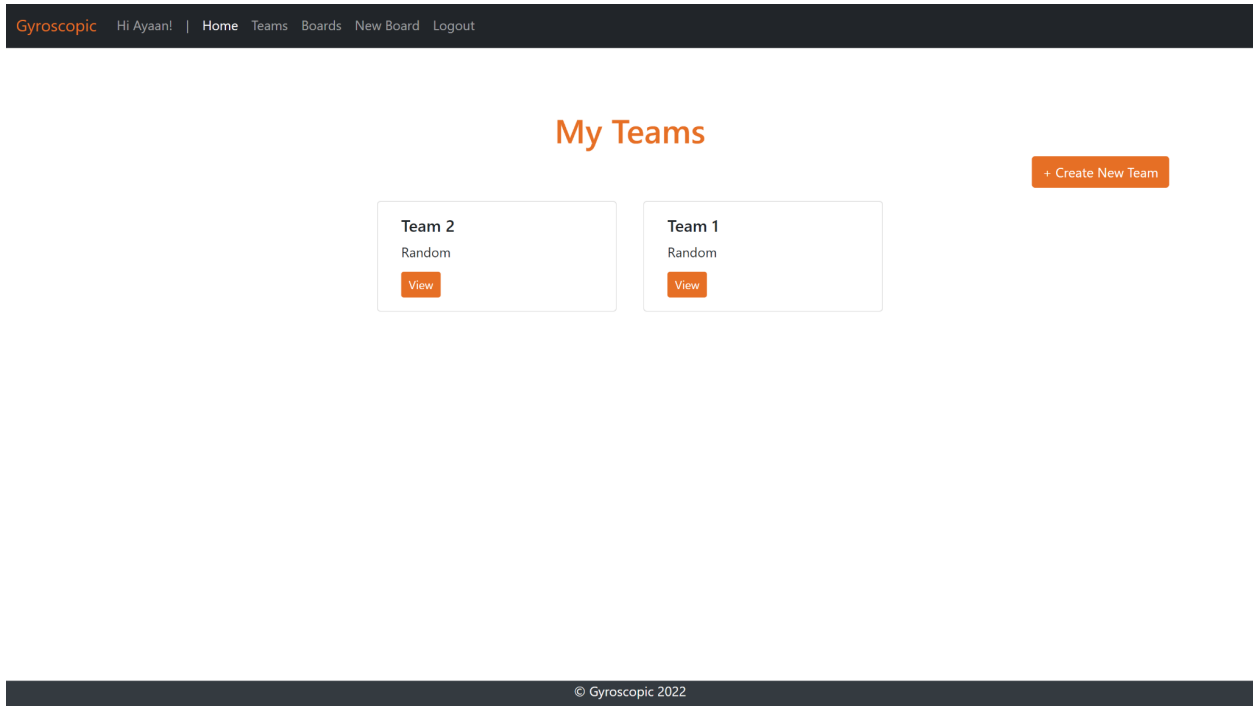
Add Board

< < All Boards

© Gyroscopic 2022

My Teams Page:

This page shows the users all the teams that they are a part of. The user can view the boards in their teams and also add comments to the boards



Teams Page:

On this page the user will be able to view their team members, who the team owner is and also the boards assigned to the team. On this page the team owner can also edit the team name, description and send out invites to other users to view and make changes to the boards assigned to the team. The team owner can also remove members from the team simply by clicking on the remove button next to their name.

Gyroscopic | Hi Ayaan! | [Home](#) [Teams](#) [Boards](#) [New Board](#) [Logout](#)

Team 2

Random

Created By: Ayaan

Team members:

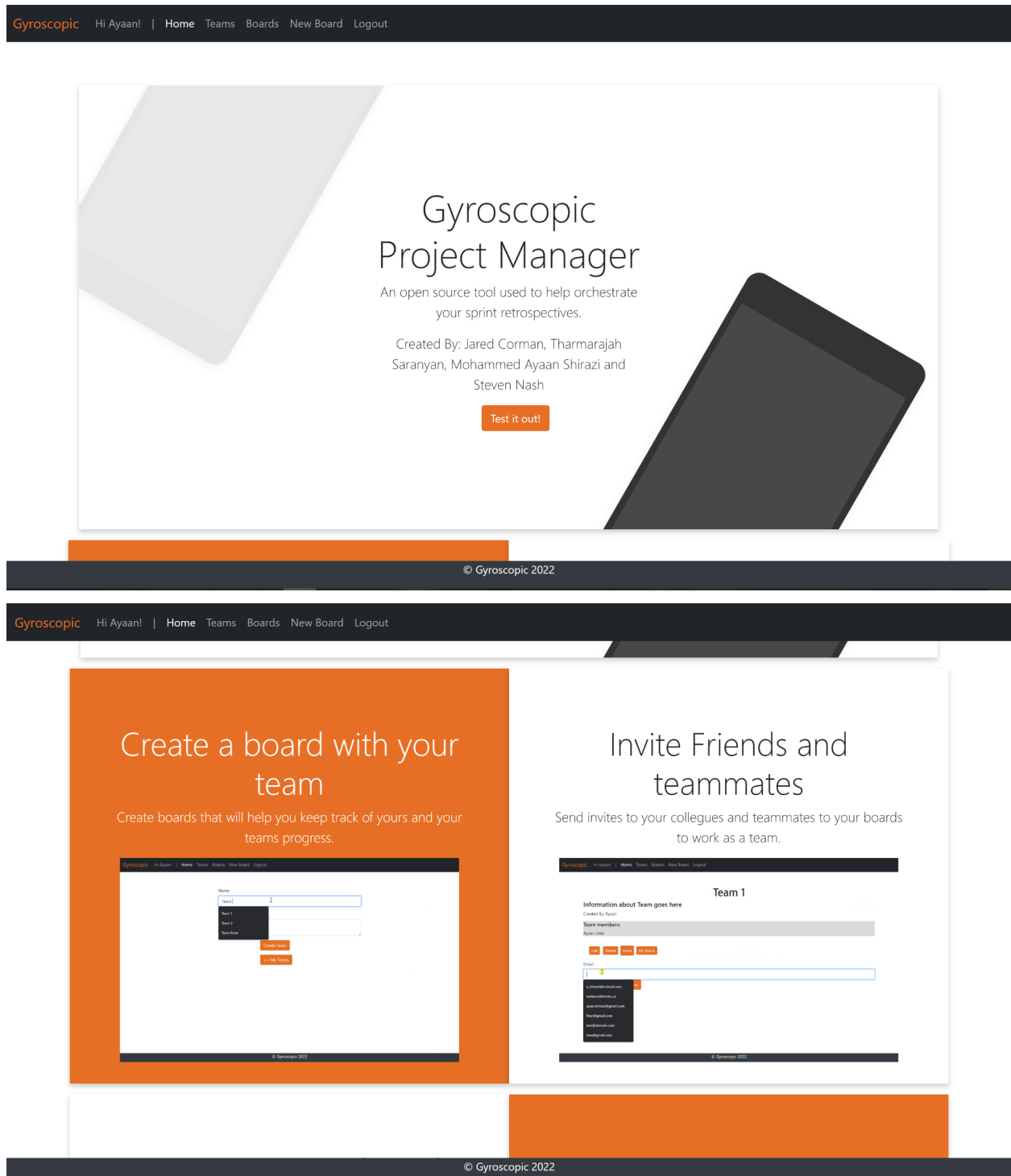
Ayaan (me)	
Error	Remove
RandomTest1	Remove

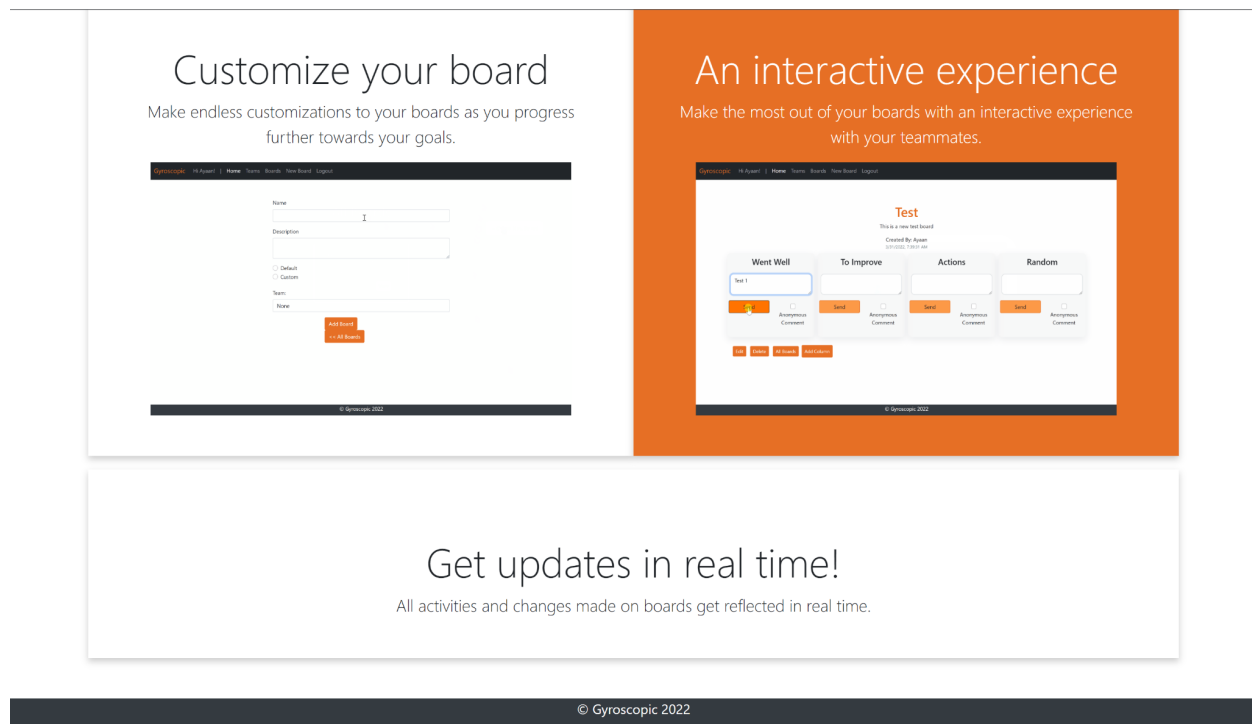
[Edit](#) [Delete](#) [Invite](#) [My Teams](#)

© Gyroscopic 2022

Landing Page:

This page can be accessed by a user regardless of whether or not they are logged in. This page just shows the user the different functionalities that are available for the user to use in their projects and gives them a rundown of what the purpose of the project is.





6) Open project issues (your client hasn't given you the full list of things you need for 4 and 5 is an Open project issue)

a) Are you using any off the shelf solutions you haven't figured out yet?

- We were able to figure out all the off shelf solution that we used in this project and were able to implement them successfully into our project

b) Any major programming features that don't yet work?

- None. We were able to tackle all the major programming features that Gyroscopic wanted us to implement in the system and all high priority features have been included and implemented in the project.

c) Any new tasks that need to be assigned (generally driven by major changes in requirements)

- None

d) Any risks which you face which you can't easily mitigate

- No major issues were seen while testing the project

7) Deployment planning

a) Plan for deployment

Gyroscopic requested the project be a docker container to allow for easy deployment. Since all relevant libraries and correct versions are all stored within the container. Currently we've yet to set up docker containers but we have a detailed list of libraries and tools being used within the project.

b) Training of the Client

Part of the non-functional requirement is the project be intuitive, defined in short as board members and team members not needing external resources to be capable of using the product within 15 minutes. While leading roles such as team creator and board creators within 30 minutes with the possibility of external support. Ideally the project would be intuitive enough for all users to but if such leading roles have a slow learning curve understanding the variety of features they have access to then linking to a series of short youtube videos may be necessary.

c) Training/documentation of whomever is supposed to maintain your project when you're done

There will be a "read me" page describing general information and the overall structure of the project in brief. Additionally there will be detailed documents and documentation within the project that were created over the course of the project life cycle such as this one you are reading now.

8) Breakdown of remaining tasks and how you intend to allocate that work amongst your group.

Priority 1 (critical features)

1. As a user, I would like to be able to create a board - Completed
2. As a board creator, I would like to be able to share a link so that other people can join my board session - Completed
3. As a board creator, I would like there to be 3 default columns "Went well", "To improve" and "Actions" on my newly created board, so that I do not have to manually add them each time a board is created - Completed
4. As a board member, I want to be able to anonymously add comments to a column - Completed
5. As a board member, I want to be able to remove comments that I have added to a column - Completed
6. As a board member, I would like to be able to see the updates made to a board in real-time --Completed

Priority 2

7. As a board member, I want to be able to upvote a comment - Completed
8. As a board member, I want to be able to downvote a comment - Completed

Priority 3

9. As a board creator, I want to be able to remove comments that anyone has added to a column - Completed
10. As a board member, I want to be able to edit my comments that I have added to a column - Completed
11. As a board creator, I want to be able to edit comments that anyone has added to a column - Completed

Priority 4

- 12. As a board creator, I want to be able to add additional columns with custom names - Completed
- 13. As a board creator, I want to be able to remove columns
- 14. As a board creator, I want to be able to edit a column's name

Priority 5

- 15. As a board member, I want to be able to move my comments from one column to another - Completed
- 16. As a board creator, I want to be able to move comments that anyone has added to a column - Completed

Priority 5

- 17. As a new user, I would like to be able to register on the system - Completed
- 18. As a user, I would like to be able to log in to the system - Completed
- 19. As a user, I would like to be able to logout of the system - Completed
- 20. As a user, I would like the boards that I have participated in to persist so that I can view them Later - Completed

Priority 6

- 21. As a user, I would like to be able to create a team - Completed
- 22. As a team creator, I would like to be able to invite users via email to join my team (even if they are not yet registered) - Completed
- 23. As a team creator, I would like to be able to add a new board to my team - Completed
- 24. As a team creator, I would like to be able to remove a member from my team - Completed
- 25. As a team creator, I would like to be able to provide custom names to my team's boards - Completed
- 26. As a team creator, I would like to be able to remove a board - Completed
- 27. As a logged in user, I would like to see a list of teams that I belong to - Completed
- 28. As a logged in user, I would like to see the list of boards belonging to my various teams - Completed

Priority 7 (stretch goals)

- 29. As a team creator, I would like the system to remember the board layout (custom column layout) from the previous board, so that I do not need to recreate the custom board layout that my team is using - Not Completed
- 30. As a board member, I would like it to be easy to switch between my team's boards from the board view - Completed