



## **Design Document**

### **Team Members:**

Abigail Eastman  
Brian Intile  
Christopher Malitsky  
Caeleb Nasoff  
Martin Price  
Radames Rivera  
Matthew Rubin

### **Mentors:**

Dr. Danielle Arigo  
Dr. Andrea Lobo

**P.O.C. :** Christopher Malitsky

**Github:** <https://github.com/Gman8r/HARTProject>

**Slack:** cool-cats-project.slack.com

**Date:** May 10, 2019

## Table of Contents

I.	High Level Design	3
II.	Problem Solving Approaches	4
III.	Screens/Screen Navigation	5
IV.	Backend	10
V.	Technology Stack	13
VI.	Goals/Responsibilities	14

## **I. High Level Design**

The purpose of BUBL is to give the user a calm, simple and relaxing application that aids in self-regulation.

When a user first opens BUBL, they will be prompted with the tasks they have created. These tasks are user defined. The user will be able to create a task with a specific title, description, priority and repeatability. All tasks, when created, will be displayed in a bubble on the home screen of the application. Information regarding the tasks and their settings will be stored locally on the user's phone. Data will be stored locally on a user's phone and will not be available to any device besides the device the application is installed on. This will allow privacy for the user and their goals.

Upon completing a task, the user will pop the corresponding bubble. This action will result in the task being shown in an outline until it is intended to return. Having a screen full of bubbles will encourage the user to complete their tasks. The user also has the option of viewing their tasks in a list view, which gives them the opportunity to view tasks in a way that they are accustomed to from other applications. Additionally included is the ability to customize parts of the experience by choosing themes and different font sizes. Users are also able to track their progress through the use of a statistics screen. This helps with motivation and accountability while gamifying the experience.

## **II. Problem Solving Approaches**

One of the primary problems was deciding what platform to develop for. Initially, it was decided to only use Android since trusted sources had warned against the use of Flutter to develop for both Android and iOS. After some research, it was decided that Flutter was to be used to develop for both platforms. This presented several new problems to overcome as the Flutter SDK was new to everyone on the team and it uses a relatively new language called Dart, which the team was also unfamiliar with.

Another problem that needed to be dealt with was that some of our initial features required approval from IRB since they involved storing users' personal data on a backend. This created a challenge where the team needed to cut those features because there would not be enough time to go through the approval process while keeping the project at an appropriate level of complexity for the class. To deal with this, the group discussed what would need IRB approval with a field expert while expanding on ideas, such as a task list, that would not require any storage of personal data.

In addition to requiring IRB approval, many of those same initial features would have required a web-server database, authentication, and other research into networking technologies. Considering the in-depth and UI-centric features already planned, the group agreed that narrowing scope to a more local task-based application was the best decision. Focusing the team's efforts on a local application allows the local features to be more fleshed-out and functional over the single semester provided to create it. Furthermore, the project's central dogmas of task management and self-improvement are equally compatible with a local app.

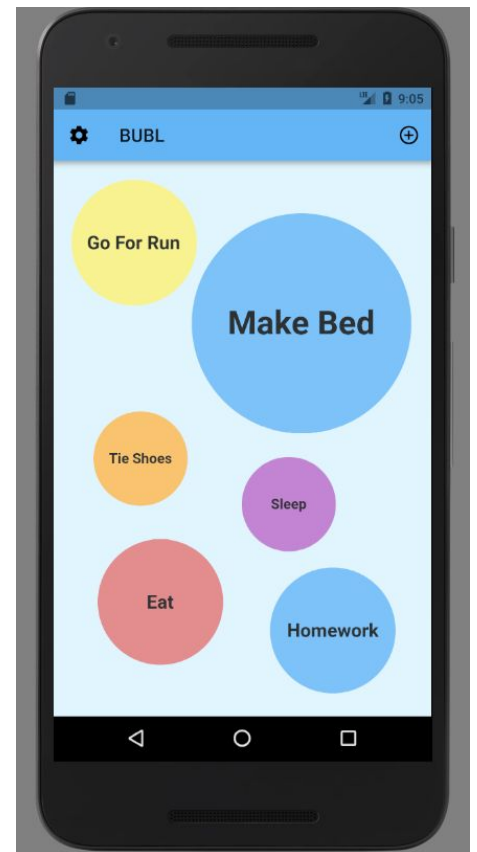
We faced an additional issue when we approached the decision of what happens to a bubble after it gets popped. Our mentor, Dr. Arigo, wanted us to focus on the satisfaction of popping and clearing a bubble when completing a task. At the same time, she also mentioned how the user should be able to see their progress and their completed tasks for a sense of accomplishment. These two principles seemed to conflict with each other, but we pieced together a solution where the bubble disappears completely with a satisfying animation and then reappears after a brief delay as a dotted outline of a bubble. This gives the user the satisfaction of seeing a bubble completely eradicated on completion, but a moment later still gives them the reminder that they have completed that task for the day.

### III. Screens/Screen Navigation

The following graphics are in-app screenshots:

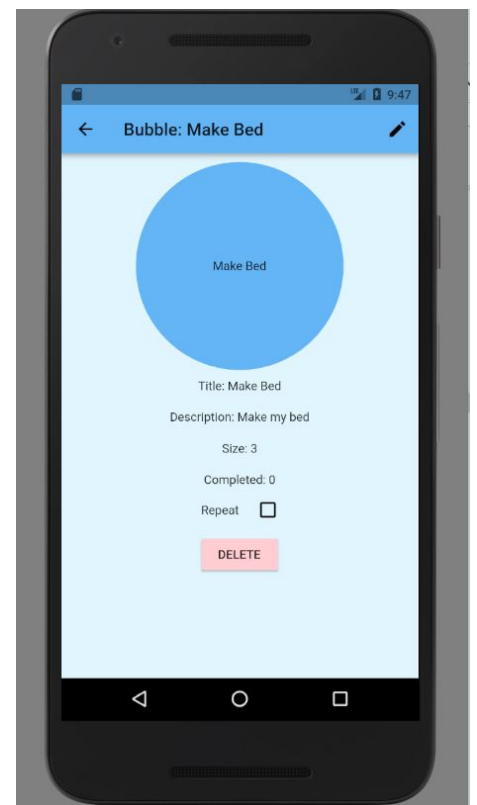


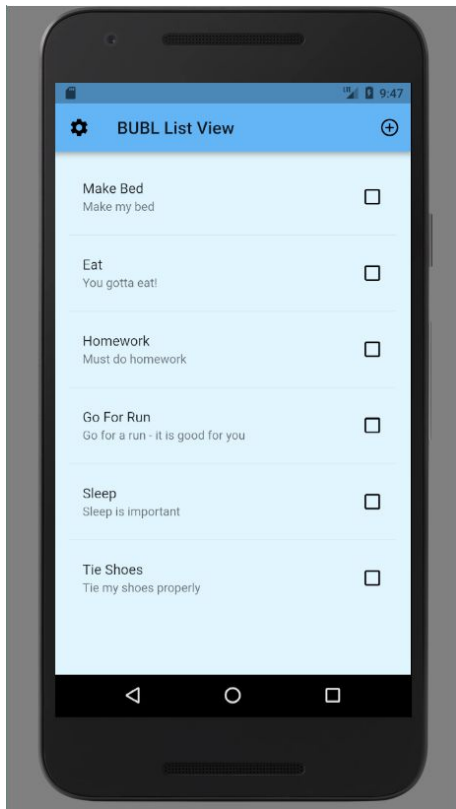
**1. Initial Bubble View** - a display of bubbles which are tasks that need to be completed. These tasks have been created by users with a Title, Description and a Priority (the higher the priority, the larger the bubble)



**2. Bubble Movement** - Go ahead and move those bubbles around any way you would like!

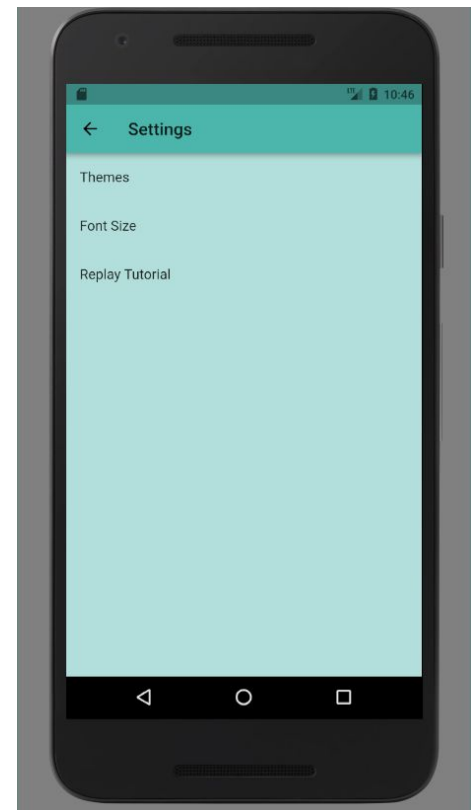
**3. Bubble Detail Screen** - Check out one of those bubbles and see some of the attributes associated with it.



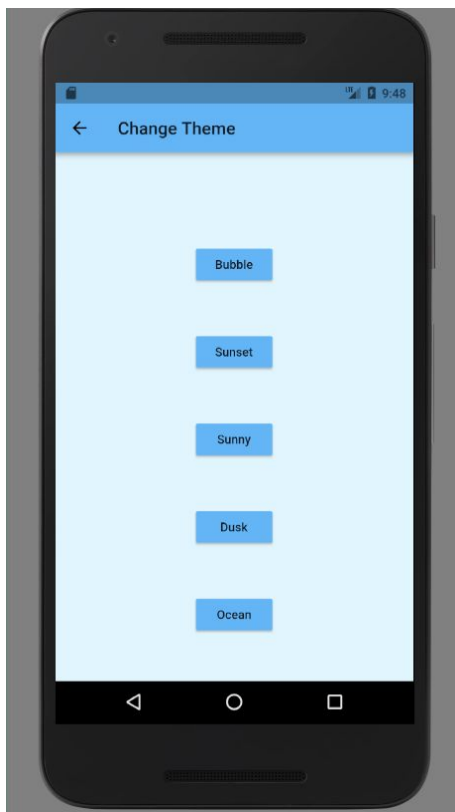


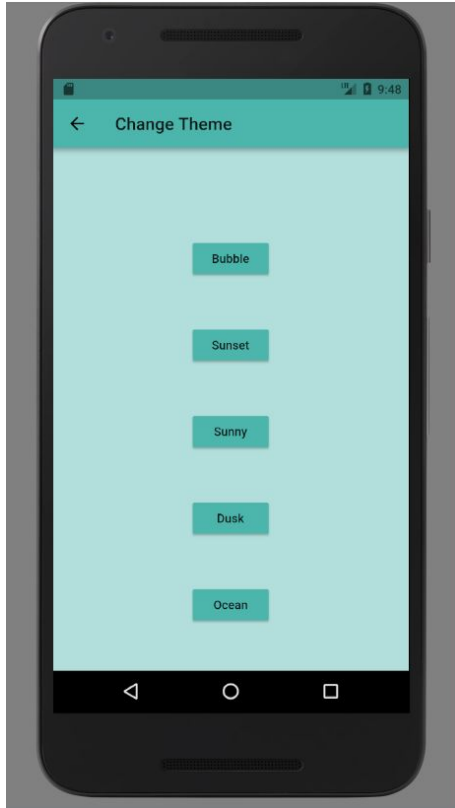
**4. Checking List View** - Perhaps you are kind of uncomfortable with seeing bubbles everywhere. No problem! Swipe left to see the *List View* (and swipe right from *List View* to go back to *Bubble View*)

**5. Settings Page** - Changing settings couldn't be easier with the settings page, just tap the gear icon in the top left!



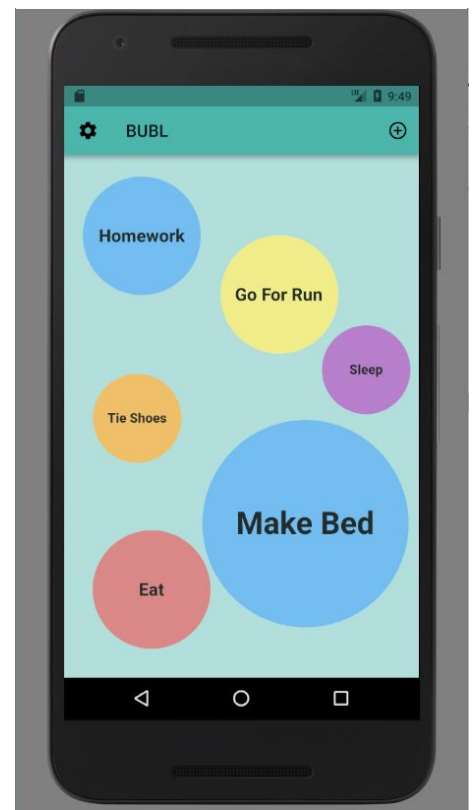
**6. Theme Selection** - Maybe blue isn't your thing. No problem! Go ahead and navigate to the Theme Selection Page through the Settings screen!



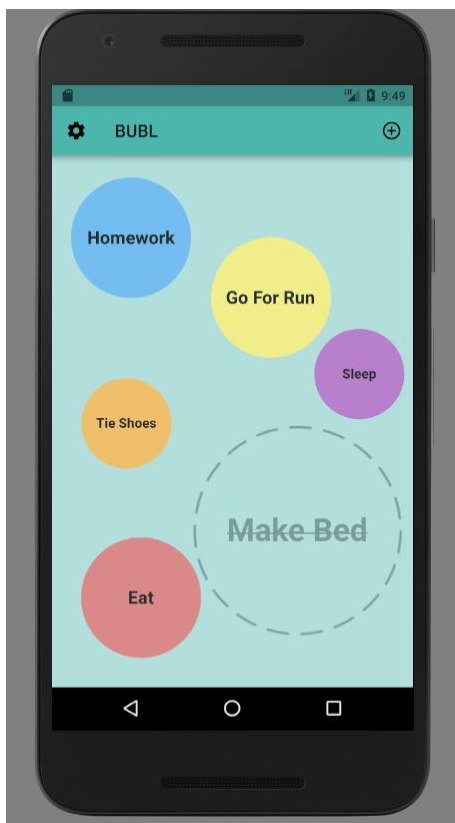


**7. Change Theme** - Let's go ahead and make it Ocean themed!

**8. Theme in the App** - Let's see the new theme in action.

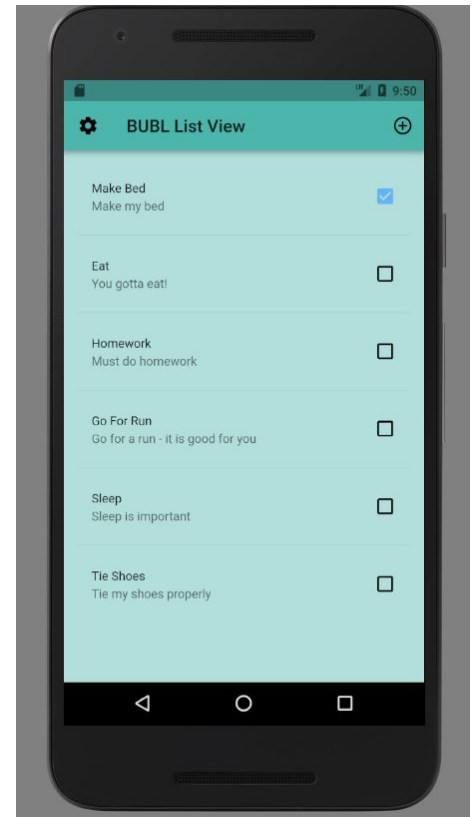
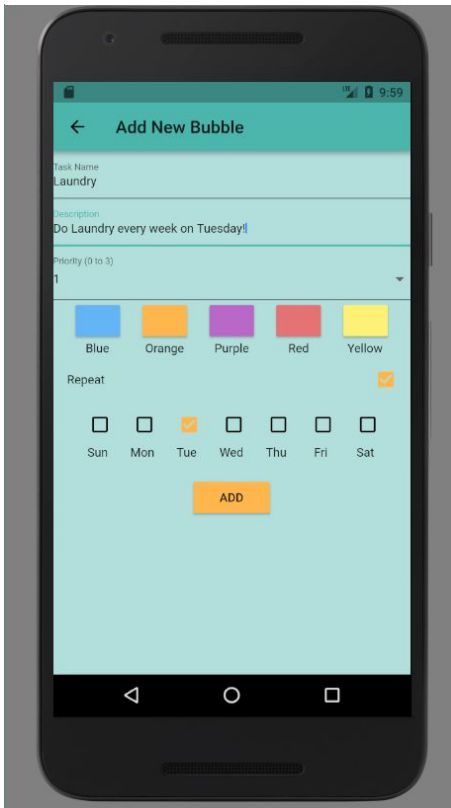


**9. Complete a task/ pop a bubble** - This is an important achievement! Popping a bubble means you completed a task. Good job making your bed - this is a great start to a day! Now isn't that screen just begging to be cleared of all those bubbles?

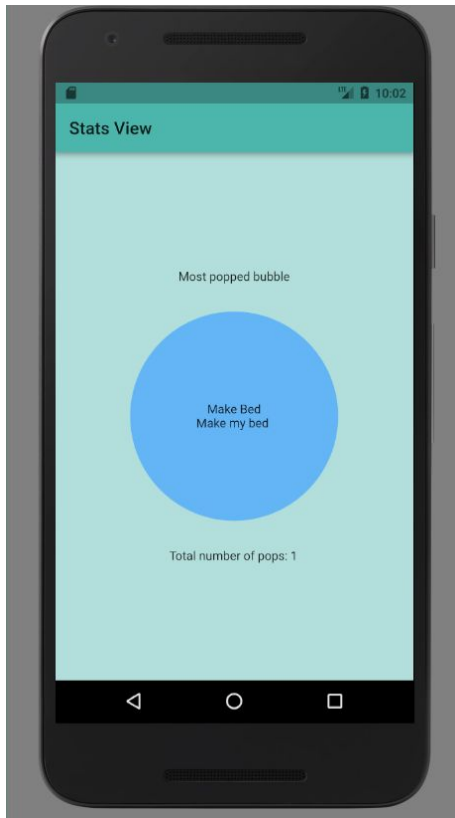


**10. Bubble Completion in List View** - You can see your accomplishment reflected in the List View.

**11. Add New Task/Bubble** - Let's navigate back to the menu and add a new task with the plus button at the top right of the menu. This will bring up a short form - Title, Description, Priority, Color, and Repeat. Let's name this "Laundry" so we can remember to do the laundry and let us set it to every Wednesday.

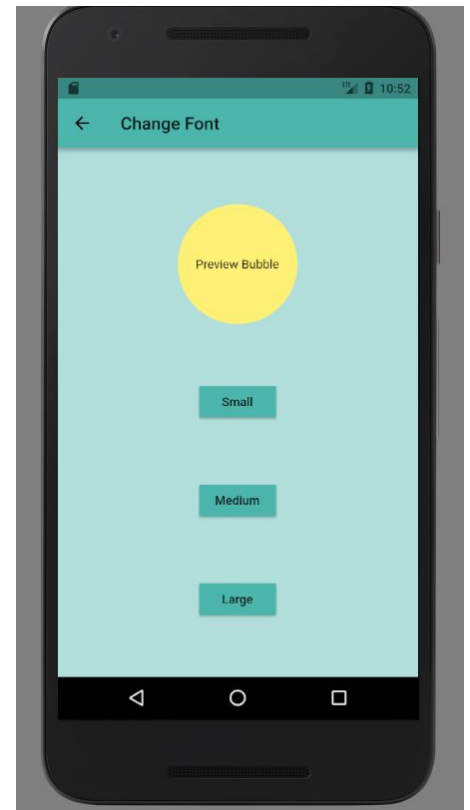






**12. Statistics View** - Want to see your progress? No problem - check out the statistics view!

**13. Text Size Page** - Change the text size so you can see better (or worse?).



## IV. Backend

BUBL implements a local SQLite database which is created and accessed within the code of BUBL itself. The use of an SQLite database is very beneficial in that both Android and iOS devices have this preinstalled. Dart as a language offers a local package, SQFlite, that allows for Dart to interact directly with any SQLite database.

The original schema for BUBL's database had two tables: bubble and pop\_record. As the app progressed, the app\_state and color\_themes tables were added into the schema. App\_state is used as a way to keep track of the times the user has opened up BUBL on their phones. This login feature is vital to the bubble's core functionality of popping, or completing the task, and it remains in this state until the user logs in on a new day, in which the bubbles are refreshed.

The bubble table has remained relatively the same, in that it keeps the attributes needed for bubbles to be repopulated into the app. Over development, more attributes were added as functionality grew. An example of this is the days\_to\_repeat column. This column is vital to having bubbles repeat on specific days of the week. Since SQFlite does not support booleans, bubbles store and retrieve a formatted string which contains all days that the bubble repeats.

The backend of BUBL also implements XML to store global settings the user may make in the app. The XML document is created and stored in the same location as the path location for the database on the phone. The decision was made to use an XML file rather than another database because an XML file is faster to read and write to than querying and updating a database table. The current settings that are available for the user to change are the font size and the current color theme. These two settings are stored as fields in the XML and are read in on app startup.

### **bubble**

bID	title	description	color_red	color_green	color_blue	opacity
INT	TEXT	TEXT	INT	INT	INT	REAL

size	posX	posY	time_created	time_deleted	frequency	days_to_repeat	times_popped
INT	REAL	REAL	TEXT	TEXT	INT	TEXT	INT

bID : Stores the bubbles ID, primary key

title: Bubble's entry field, text shown in the bubble

description: Bubble's description field

color\_red: Bubble's RGB red value

color\_green: Bubble's RGB green value

color\_blue: Bubble's RGB blue value

opacity: Bubble's opacity value, used in tandem with RGB values

size: Bubble's priority and how big it appears on screen

posX: Bubble's X - coordinate on screen

posY: Bubble's Y - coordinate on screen

time\_created: Timestamp of when the bubble was created

time\_deleted: Timestamp of when the bubble was deleted

frequency: Holds if the bubble repeats or not

days\_to\_repeat: Formatted text of all days the bubble repeats

times\_popped: Number of times the bubble has been popped

### **pop\_record**

pID	bID	time_of_pop	action
INT	INT	TEXT	TEXT

pID: the pop's ID, primary key

bID: the ID of the bubble popped, foreign key

time\_of\_pop: timestamp of when the bubble was popped

action: describes if the pop is still valid, unpopped

### **app\_state**

loginID	last_opened
INT	TEXT

loginID: The logins ID, primary key

last\_opened: timestamp of the login time

### **color\_themes**

colorID	color_name	color_red	color_green
INT	TEXT	INT	INT

color_blue	opacity	theme_name	theme_pair
INT	REAL	TEXT	TEXT

colorID: The color's ID, primary key

color\_name: The name of the color

color\_red: Color's RGB red value

color\_green: Color's RGB green value

color\_blue: Color's RGB blue value

opacity: Color's opacity, used in tandem with RGB values

theme\_name: The name of the theme that the color belongs to

theme\_pair: Whether the theme has a partner theme, if so, the partners name

### **Xml settings document**

```
<xml>
  <settings>
    <font_size> </font_size>
    <current_theme> </current_theme>
  </settings>
</xml>
```

## **V. Technology Stack**

We will be developing BUBL for both iOS and Android devices using the Flutter framework and Dart language. The development will be done using Android Studio/VS Code with testing done using emulated hardware running the latest versions of Android and potentially iOS. We came to this conclusion after exploring the possibility of either solely developing for Android (using Android Studio and Java) or using Flutter. Although as a team we are all familiar and versed in Java, the key reason we are using Flutter is how we would like our application to eventually be available on both iOS and Android to reach the most of our target audiences as possible.

All information pertaining to the user and their Bubble tasks will be saved locally, split amongst the app itself and the user's phone SQLite database. Additionally, there could be user-sensitive information inputted into Bubble which would be dangerous to store on a server. Not using a web-server database will give us more time to work on making Bubble a better and more user friendly, local application.

## VI. Goals/Responsibilities

By the mid-assessment, BUBL plans to be in its *core functionality* stage. This includes bubble functionality and detail available to the user as well as different screens (Main Bubble view, List view, Detail screen, Edit screen) that can be accessed. All of the bubbles will have their states stored into the SQLite database on the mobile device.

### Responsibilities

- **Core App Functionality, Features and UI Design**
  - Caeleb Nasoff
  - Abigail Eastman
  - Chris Malitsky
- **Back End/Database**
  - **Design**
    - Brian Intile
    - Caeleb Nasoff
    - Martin Price
  - **Implementation**
    - Martin Price
- **Miscellaneous Functionality and Feature Integration**
  - Caeleb Nasoff
  - Abigail Eastman
  - Brian Intile
  - Chris Malitsky
  - Matthew Rubin
  - Radames Rivera

For specific task descriptions and assignments, please refer to our Trello Board

### Trello Board Invite Link:

<https://trello.com/invite/b/6qJolGH0/dc58d8200adeeb78870501f0ab342fae/scrum-board>