



## **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:** March 5, 2019

## Table of Contents

I.	High Level Design	.....	3
II.	Problem Solving Approaches	.....	4
III.	Screens/Screen Navigation	.....	5
IV.	Backend	.....	9
V.	Technology Stack	.....	10
VI.	Goals/Responsibilities	.....	11

## **I. High Level Design**

The purpose of BUBL is to give the user a calm 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 priority for their task. 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 becoming invisible until it is intended to return. Having a screen full of bubbles will encourage the user to complete their tasks.

## **II. Problem Solving Approaches**

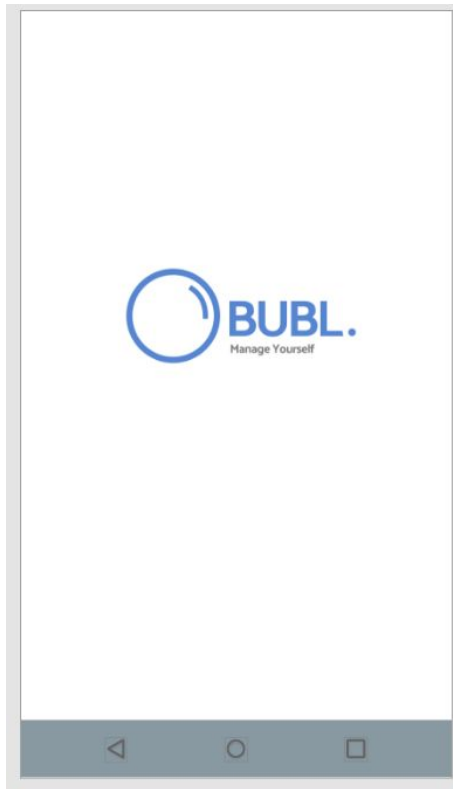
One of the primary problems was deciding what platform to develop for. Initially, it was decided to only use Android since trusted sources had warned against the use of Flutter to develop for both Android and iOS. After some research, it was decided that Flutter is to be used to develop for both platforms.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

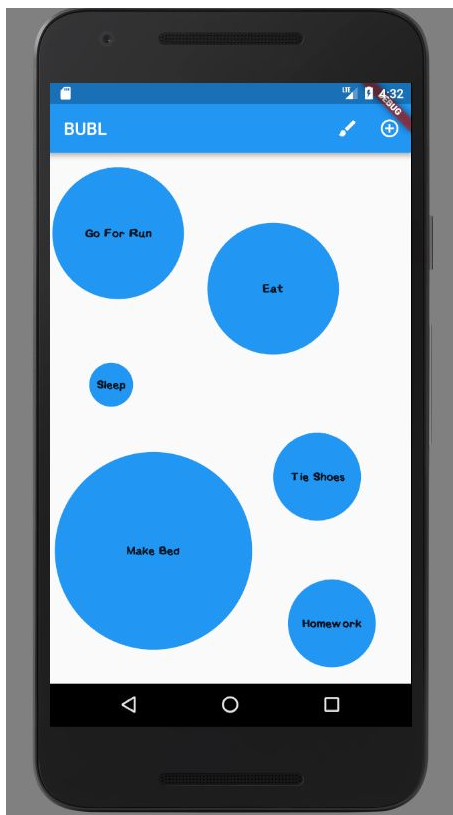
In addition to requiring IRB approval, many of those same initial features would have required a web-server database, authentication, and other research into networking technologies. Considering the in-depth and UI-centric features already planned, the group agreed that narrowing scope to a more local 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.

### III. Screens/Screen Navigation

The following graphics are in-app screenshots:



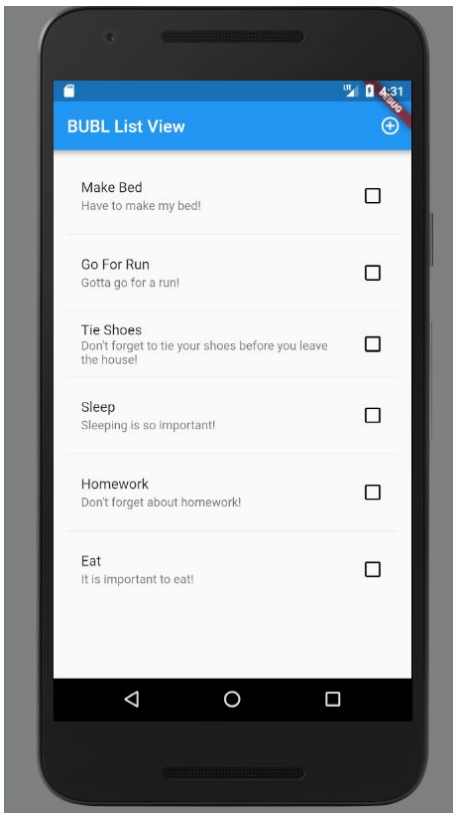
**1. Title/opening screen** - simple design. The simpler, the better. Something clean, something somewhat refreshing, not intimidating, but inviting for a user to explore.



**2. 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)



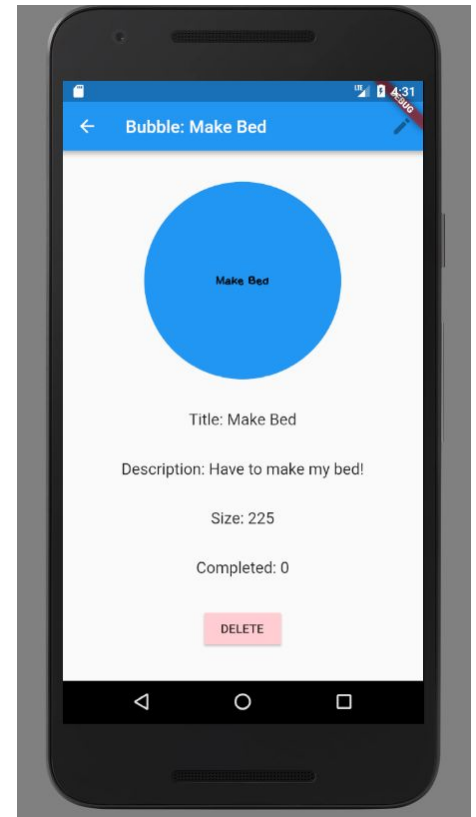
**3. Bubble Movement** - Go ahead and move those bubbles around any way you would like!

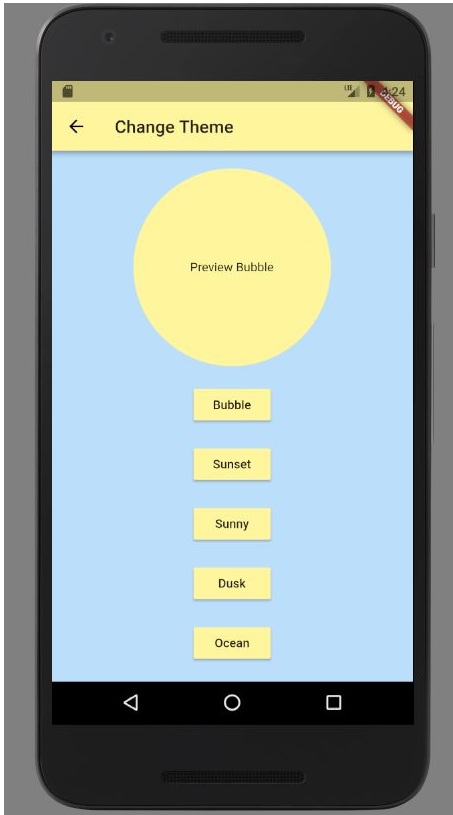


**4. Bubble Detail Screen** - Check out one of those bubbles and see some of the attributes associated with it.

**5. 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*)

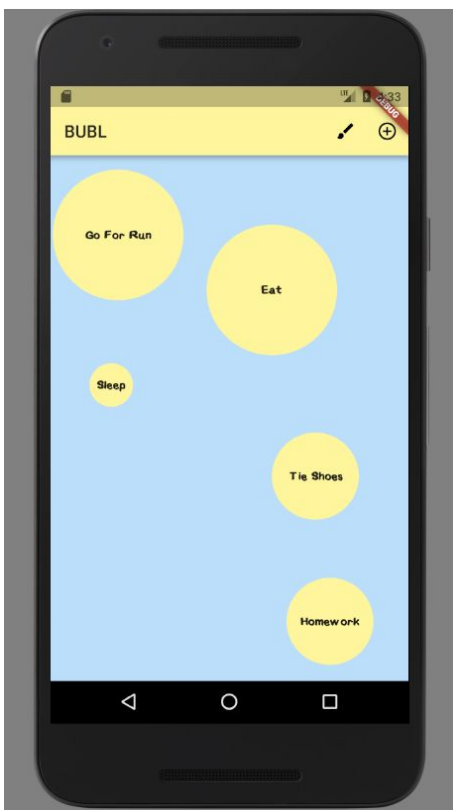
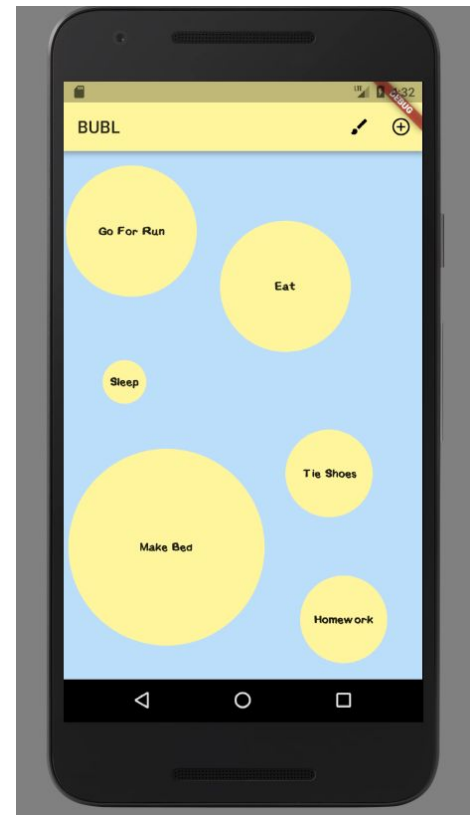
**6. Theme Selection** - Maybe blue isn't your thing. No problem! Go ahead and navigate to the Theme Selection Page by pressing the paint brush at the top right corner!





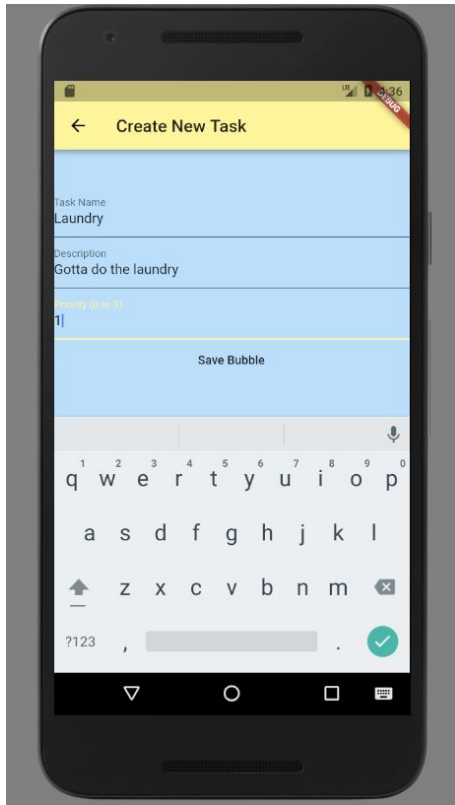
**7. Change Theme** - Let's go ahead and make it Sunny!

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



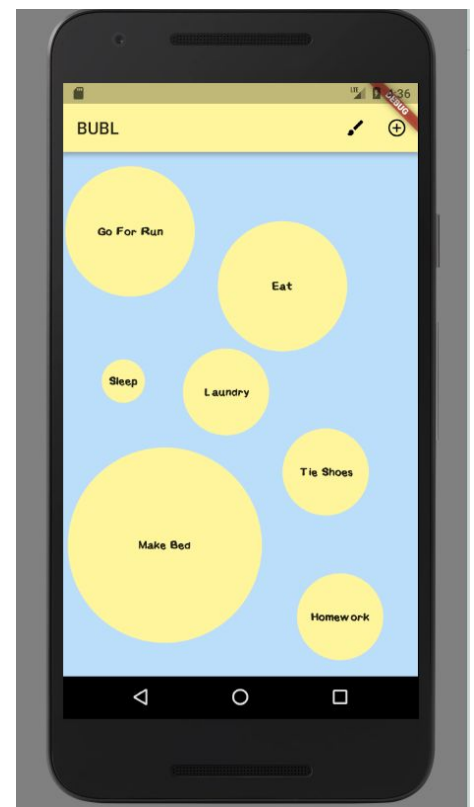
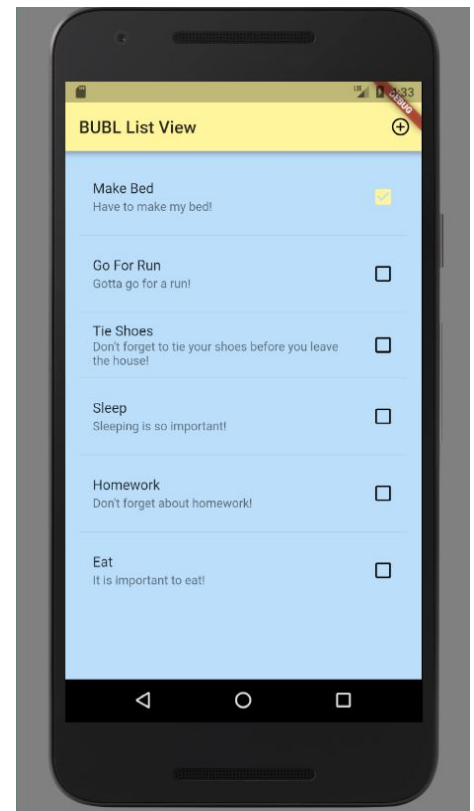
**9. Complete a task/ pop a bubble** - This is an important achievement! Popping a bubble means you completed a task. Good job making your bed - this is a great start to a day! Now isn't that screen just begging to be cleared of all those bubbles?

**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 bottom left of the menu. This will bring up a short form - Title, Description, Priority. Let's name this "Laundry" so we can remember to do the laundry.

**12. Observe New Task** - we can now see the task *Laundry*! Time to complete it and clear up the rest of those bubbles for today - a clean, bubbleless screen is a beautiful sight!





## IV. Backend

In the early stages of BUBL, a backend was not initially considered. As an app that would have personal information to each user, it was thought that keeping all data in the app itself would be best. This solution would eliminate any risk of personal, confidential data being lost or stolen.

In the current design plan, BUBL will be implemented with access to the mobile device's built in SQLite database. The need for a database came from the issue of what happens to bubbles when they are popped or removed. With bubbles being saved to a database, this keeps information stored in the app itself kept to a minimum. Now, BUBL will query the database for active bubbles to populate the main screen, and when a bubble is popped, delete it from the app and update its state in the database.

The database for BUBL, as of this prototype, will consist of two tables:

### **bubble**

bID	title	description	size	display_order	posX	posY	color	time_created	time_deleted	frequency
int	varchar	varchar	int	int	float	float	string	timestamp	timestamp	int

### **pop\_record**

pID	bID	time_popped	action
int	int	timestamp	enum

The bubble table, holds all the information that from the variables of the Bubble.dart class. This information is queried everytime the app is reopened to recreate the bubbles that are still active, time\_deleted is null. The pop\_record table, holds the times that a bubble has been popped. This table is used in tandem with bubble to only show the bubbles that have not been popped today when reopening the app later in the day.

## **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**
  - Martin Price
  - Brian Intile
- **Miscellaneous Functionality and Feature Integration**
  - Caeleb Nasoff
  - Abigail Eastman
  - 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>