

The Buzz

Developer Manual

Contents

- **Contributors & Roles** [\(3\)](#)

Resources

- Repository Location & Branch Structure [\(4\)](#)
- **Repository Structure** [\(5\)](#)
 - Backend [\(6\)](#)
 - React [\(7\)](#)
 - Mobile [\(8\)](#)
 - Admin CLI [\(9\)](#)

Development

- Requirements, Configurations, & Settings [\(10\)](#)
- Run, Build, & Deploy [\(11\)](#)

Models

- System [\(12\)](#)
- Data [\(13\)](#)
- Interaction – OAuth [\(14\)](#)
- Interaction – Idea Interaction [\(15\)](#)

- **Behavioral Notes** [\(16\)](#)
- **Contact** [\(17\)](#)
- **License** [\(18\)](#)

Contributors & Roles

(by sprint)

Sprint 1

Project Manager

Chris W.

Backend (Java Spark)

ThetaBird

Frontend (React)

Lauren W.

Mobile (Flutter)

Hangyi L.

Admin CLI (Java)

Maanasa T.

Sprint 2

Project Manager

Lauren W.

Backend (Java Spark)

Hangyi L.

Frontend (React)

ThetaBird

Mobile (Flutter)

Maanasa T.

Admin CLI (Java)

Chris W.

Sprint 3

Project Manager

Maanasa T.

Backend (Java Spark)

Chris W.

Frontend (React)

Hangyi L.

Mobile (Flutter)

ThetaBird

Admin CLI (Java)

Lauren W.

Sprint 4

Project Manager

ThetaBird

Backend (Java Spark)

Maanasa T.

Frontend (React)

Chris W.

Mobile (Flutter)

Lauren W.

Admin CLI (Java)

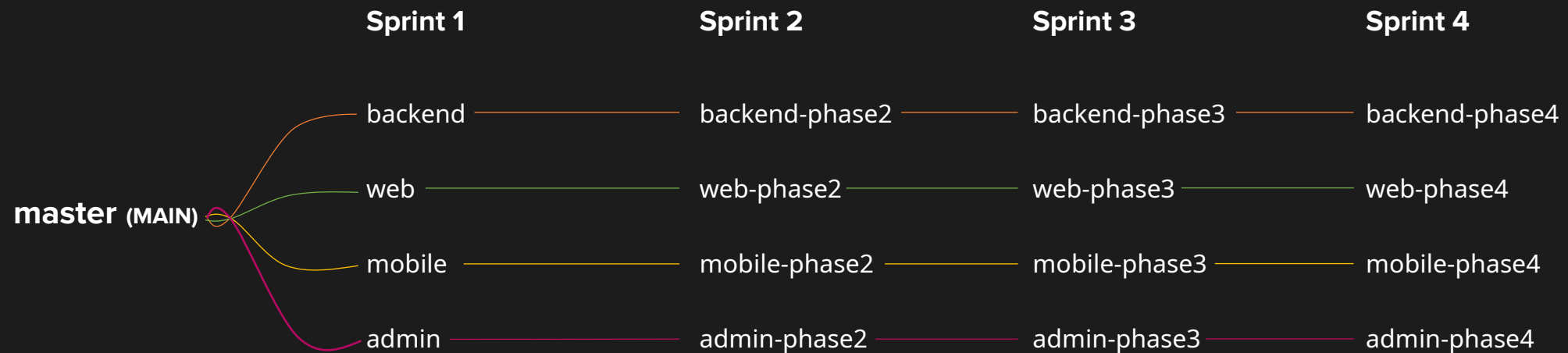
Hangyi L.

Resources

(Repository Location & Branch Structure)









Private Repository

https://bitbucket.org/chm321/cse216_spring22_group4



Resources

(Repository Structure)

	<code>.vscode</code>	Irrelevant IDE directory. Should've been included in <code>.gitignore</code>
	<code>admin-cli</code>	Admin CLI – Directly interacts with Heroku DB Manipulate tables, columns, rows, etc. Standalone program.
	<code>backend</code>	Backend – Interface between frontend & Heroku DB. Provide routes, accept requests from frontend and respond appropriately. Deployed to Heroku.
	<code>mobile</code>	Frontend (Flutter) – Mobile application, interact with backend. Make requests to backend, show responses in a user-friendly format. Deployed to Android and/or iOS emulator/device.
	<code>react</code>	Frontend (React) – Web application, interact with backend. Make requests to backend, show responses in a user-friendly format. Packaged with npm and deployed alongside backend.
	<code>web</code>	Deprecated web application from Phase 0
	<code>.gitignore</code>	File containing addresses that shouldn't be committed (i.e., local configurations, IDE-specific files, etc.)
	<code>README.md</code>	Basic repository information. Includes brief description & route examples for use in frontends.

Resources

(Repository Structure — Backend)

- **App.java**: Init all services, enable CORS headers. Calls Routes, Database, DataStore, DriveService, & OAuth
 - **Censor.java**: Interact with Purgomalum Profanity API
 - **Database.java**: Directly interact with Heroku DB via SQL statements
 - **DataStore.java**: Indirectly interact w/ Heroku DB via Database.java by including logic & parsing data
 - **DriveService.java**: Interact w/ Google Drive API to fetch and upload attachments from Google Drive
 - **FileUpload.java**: Deprecated/Unused
 - **OAuth.java**: Interact w/ Google OAuth 2.0 & MemCachier (via DataStore) for Google Login functionality
 - **OAuthUser.java**: Data Structure – Server response for Google Login functionality
 - **RequestIdea.java**: Data Structure – Frontend POST request for idea functionality
 - **RequestOAuth.java**: Data Structure – Frontend POST request for Google Login functionality
 - **RequestReaction.java**: Data Structure – Frontend POST request for idea/comment likes/dislikes functionality
 - **RequestUser.java**: Data Structure – Frontend POST request for user data functionality
 - **Routes.java**: Set all Spark routes, verify tokens, apply logic to request, send responses
 - **StructuredResponse.java**: Data Structure – Server response for all functionality
-
- **AppTest.java**: Test – verify test functionality
 - **RequestTest.java**: Test – verify data structure functionality
 - **RoutesTest.java**: Test – verify route functionality
-
- **pom.xml**: Maven configuration file, include project info & dependencies
 - **README.md**: Deprecated/Unused

```

  ✓ backend
    > .vscode
  ✓ src
    ✓ main
      ✓ java\edu\lehigh\cse216\group4\backend
        ① App.java
        ① Censor.java
        ① Database.java
        ① DataStore.java
        ① DriveService.java
        ① FileUpload.java
        ① OAuth.java
        ① OAuthUser.java
        ① RequestIdea.java
        ① RequestOAuth.java
        ① RequestReaction.java
        ① RequestUser.java
        ① Routes.java
        ① StructuredResponse.java
      > resources
    ✓ test\java\edu\lehigh\cse216\group4\backend
      ① AppTest.java
      ① RequestTest.java
      ① RoutesTest.java
    > target
  ◆ .gitignore
  📄 pom.xml
  ① README.md
```

Resources

(Repository Structure — **React**)

- **Comment.tsx**: Component – Show data for a single comment
- **CommentReactions.tsx**: Component – Show reaction data (likes/dislikes) for a single comment
- **FileUploader.tsx**: Component – Button to bring up file selection menu
- **Greeting.tsx**: Component – Display “Welcome, [user’s first name]”
- **Idea.tsx**: Component – Show data for a single idea
- **IdeaList.tsx**: Page Component – Show list of Idea components
- **IdeaNewForm.tsx**: Sub-Page Component – Show form with inputs for a new idea submission
- **IdeaReactions.tsx**: Component – Show reaction data (likes/dislikes) for a single idea
- **IdeaSpecific.tsx**: Sub-Page Component – Show entire idea, with comments, on right-hand side drawer
- **OAuth.tsx**: Page Component – Show login page with Google Login button
- **Profile.tsx**: Page Component – Show user profile info
- **App.tsx**: Main Component – Show appropriate page according to global state
- **Hello.tsx**: Deprecated/Unused
- **index.html**: Barebones HTML template, load bootstrap & google fonts, provide “app” div for index.tsx
- **index.tsx**: Root Component – Load/render App.tsx
- **style.css**: Additional CSS styling outside of bootstrap
- **cypress.json**: Testing configuration file
- **deploy.sh**: Build w/ webpack and copy to backend resources for Heroku deploy
- **package.json**: node/npm configuration file, include project info & dependencies
- **tsconfig.json**: typescript configuration file

```
▼ react
  ▼ src
    ▼ Components
      ▼ test
        TS Apptest.tsx
        TS HomeTest.tsx
        TS IdeaTest.tsx
        TS Comment.tsx
        TS CommentReactions.tsx
        TS FileUploader.tsx
        TS Greeting.tsx
        TS Idea.tsx
        TS Idealist.tsx
        TS IdeaNewForm.tsx
        TS IdeaReactions.tsx
        TS IdeaSpecific.tsx
        TS OAuth.tsx
        TS Profile.tsx
      TS App.tsx
      TS Hello.tsx
      <> index.html
      TS index.tsx
      # style.css
      {} cypress.json
      $ deploy.sh
      {} package-lock.json
      {} package.json
      tsconfig.json
```

Resources

(Repository Structure — Mobile)

- **GlobalState.dart**: Data Structure – Global state information
- **IdeaData.dart**: Data Structure – Idea information for a single idea
- **IdeaData.g.dart**: Generate JSON factory for IdeaData
- **BuzzDrawer.dart**: Widget – Left-hand side drawer with navigation menu
- **Comment.dart**: Widget – Display a single comment
- **Idea.dart**: Widget – Display a single idea (show all data if viewing it specifically)
- **LoginButton.dart**: Widget – Google Login button, provide login functionality & logic
- **IdeaList.dart**: Widget (Scaffold) – Display a list of all ideas
- **IdeaSpecific.dart**: Widget (Scaffold) – Display a list of all ideas
- **Login.dart**: Widget (Scaffold) – Display page with login button
- **MainWrapper.dart**: Deprecated/Unused
- **NewIdea.dart**: Widget (Scaffold) – Display new idea form submission
- **Profile.dart**: Widget (Scaffold) – Display user profile data
- **Wrapper.dart**: Widget (Scaffold) – Manage which page is displayed according to global state
- **generated_plugin_registrant.dart**: Generated file for dependency functionality
- **main.dart**: Widget (Main) – Init all services and provide global state to entire app
- **widget_test.dart**: Verify widget functionality
- **pubspec.yaml**: Flutter configuration file, include project info & dependencies
- **README.md**: Deprecated/Unused

```
✓ mobile
  > android
  ✓ lib
    ✓ Components
      ✓ Data
        ⚙ GlobalState.dart
        ⚙ IdeaData.dart
        ⚙ IdeaData.g.dart
        ⚙ BuzzDrawer.dart
        ⚙ Comment.dart
        ⚙ Idea.dart
        ⚙ LoginButton.dart
      ✓ Pages
        ⚙ IdeaList.dart
        ⚙ IdeaSpecific.dart
        ⚙ Login.dart
        ⚙ MainWrapper.dart
        ⚙ NewIdea.dart
        ⚙ Profile.dart
        ⚙ Wrapper.dart
        ⚙ generated_plugin_registrant.dart
        ⚙ main.dart
    ✓ test
      ⚙ widget_test.dart
  ! pubspec.yaml
  ⓘ README.md
```


Resources

(Repository Structure — Admin CLI)

- **App.java**: Provide CLI interface, interact with Database.java after parsing input
- **Database.java**: Connect to Heroku DB and manipulate via SQL statements
- **AppTest.java**: Verify admin functionality
- **pom.xml**: Maven configuration file, include project info & dependencies
- **README.md**: Deprecated/Unused

```
✓ admin-cli
  ✓ src
    ✓ main\java\edu\lehigh\cse216
      ✓ group4\admin
        ● App.java
        ● Database.java
    ✓ test\java\edu\lehigh\cse216
      ✓ group4\admin
        ● AppTest.java
    > target
  ◆ .gitignore
  📄 pom.xml
  ⓘ README.md
```

Development

(Requirements, Configurations, & Settings)

Backend

- Min. Windows 10 or MacOS Sierra
- VSCode (Software)
- Java JDK 14.0.2 (Dev Kit)
- Maven (Installed and configured)
- Heroku (Installed and configured)
- Google Cloud (Enabled & Configured)

Web

- Min. Windows 10 or MacOS Sierra
- VSCode (Software)
- Node v17.9.0 (Dev Kit)
- npm 8.5.5 (Package Manager)
- Chromium-based browser
- React (Installed and configured)

Mobile

- Min. Windows 10 or MacOS Sierra
- VSCode (Software)
- Android Studio (Software)
- Android Emulator w/ API 29+
- Flutter (Installed and configured)
- Firebase (Enabled & Configured)

Admin

- Min. Windows 10 or MacOS Sierra
- VSCode (Software)
- Java JDK 14.0.2 (Dev Kit)
- Maven (Installed and configured)

Development

(Run, Build, & Deploy)

Backend

(Install dependencies)

- mvn clean
- mvn install

(Package and execute program)

- mvn package
- mvn heroku:deploy

Web

(Install dependencies)

- npm install

(Run debug react)

- npm start

(Create bundle for Heroku deployment)

- npm build
- sh deploy.sh
- mvn heroku:deploy (from backend directory)

Mobile

(Install dependencies)

- flutter clean
- flutter pub get

(Run debug flutter)

- flutter run

(Build flutter apk)

(<https://docs.flutter.dev/deployment/android>)

- flutter build apk

Admin

(Install dependencies)

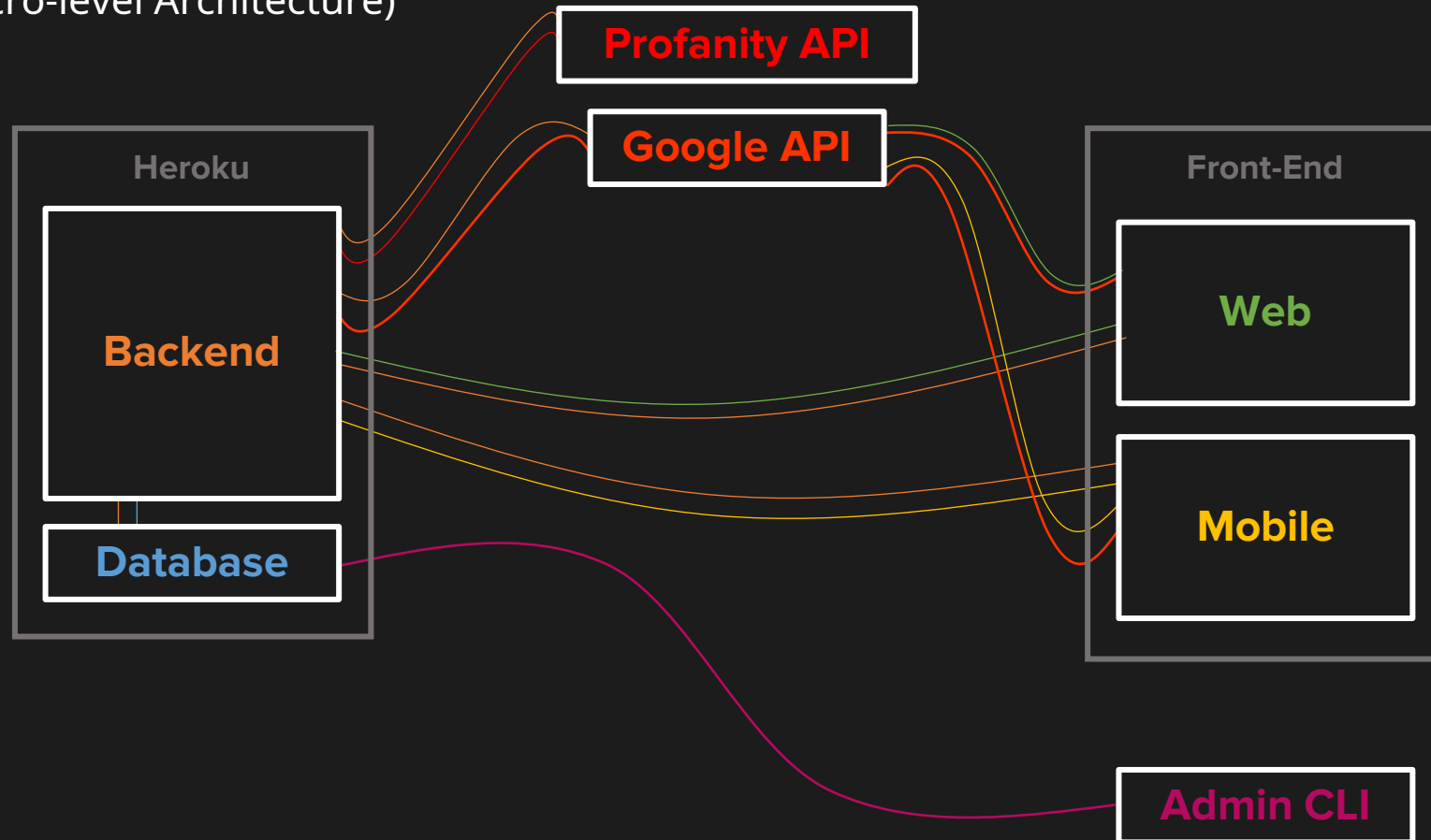
- mvn clean
- mvn install

(Package and execute program)

- mvn package
- mvn exec:java

System Models

(Macro-level Architecture)



Data Models

(Database Table Structures)

USERS

name	type	description
userId	VARCHAR(6)	Id of user
note	VARCHAR	User profile description
email	VARCHAR	User email
name	VARCHAR(50)	User's full name
avatar	VARCHAR	URL to profile picture
companyRole	SMALLINT	Unused

IDEAS

name	type	description
ideaId	BIGINT	Id of idea/comment
replyTo	BIGINT	If comment, id of parent idea
userId	VARCHAR	Id of author
timestamp	BIGINT	Date of creation (in milliseconds)
subject	VARCHAR(64)	Subject of idea
content	VARCHAR(500)	Content of idea
attachment	VARCHAR(50)	Link to attachment
allowedRoles	SMALLINT[]	Unused

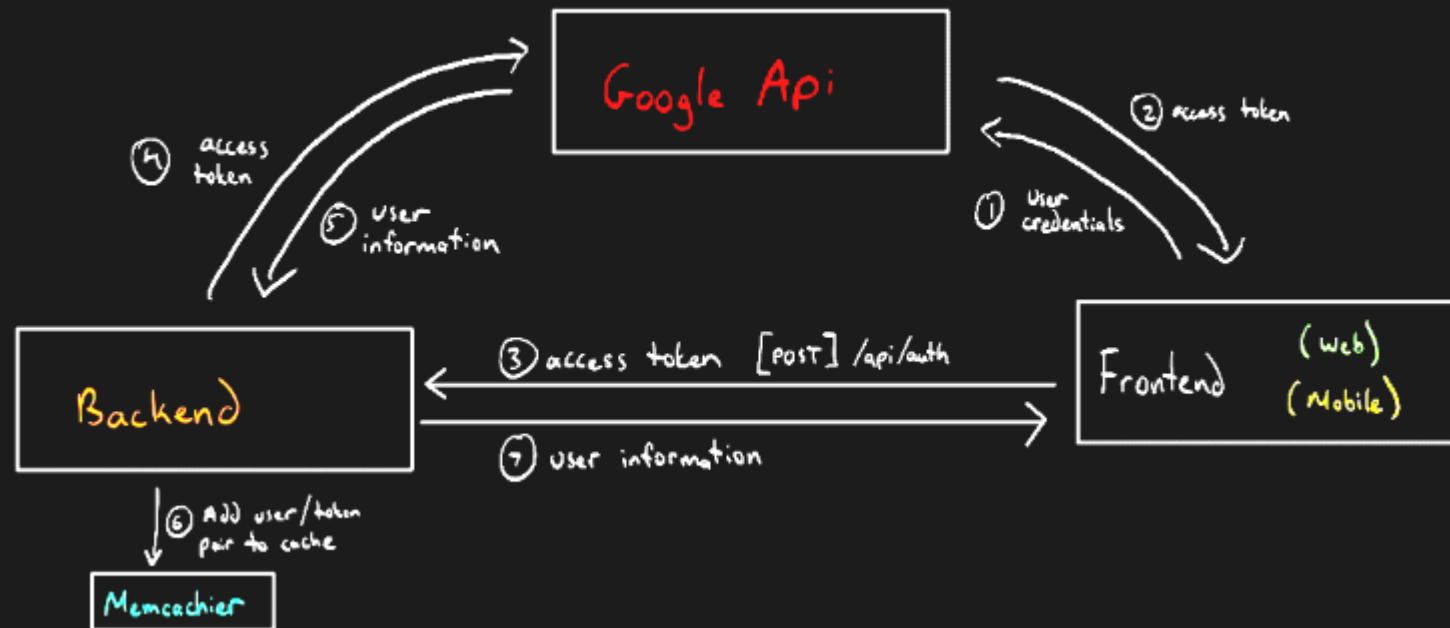
REACTIONS

name	type	description
ideaId	BIGINT	Id of parent idea
likes	VARCHAR[]	Array of user ids who liked parent idea
dislikes	VARCHAR[]	Array of user ids who disliked parent idea

Interaction Models

(REST API Behavior & Endpoints)

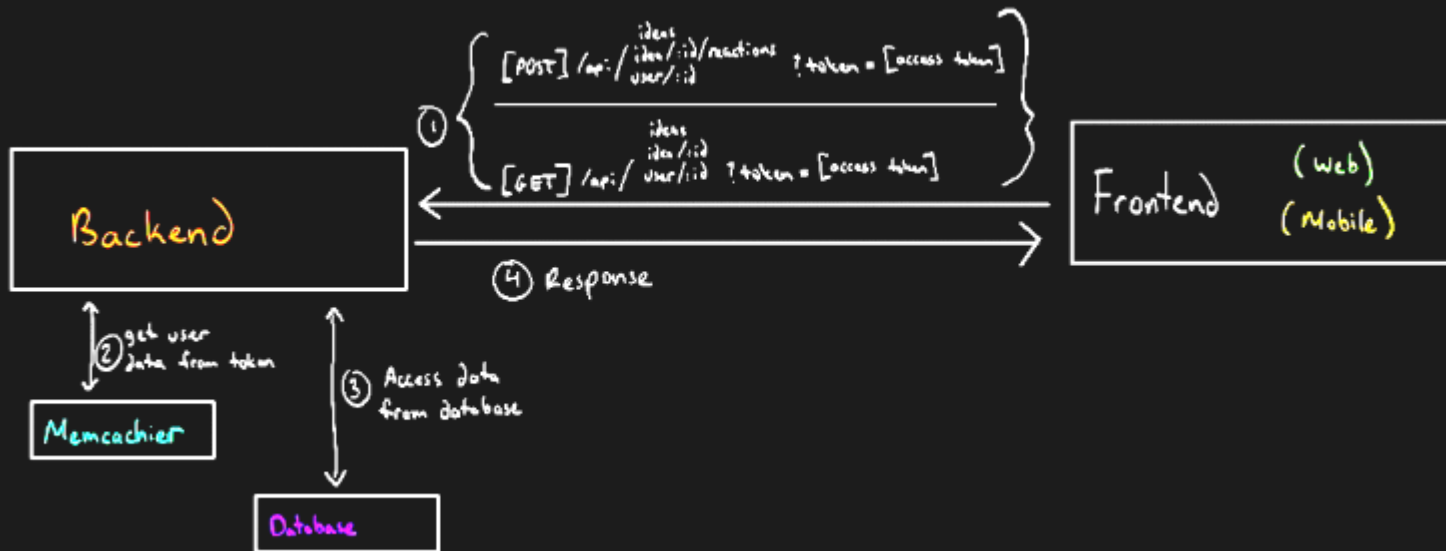
Google OAuth



Interaction Models

(REST API Behavior & Endpoints)

Idea Interaction



Behavioral Notes

(Technical Debt, Bugs, & Limitations)

- IDs for ideas/comments are assigned on a timestamp basis, and an error will occur when two idea requests are processed within the same millisecond. There were plans to transition to a twitter snowflake id, or at least to a SHA-128 key for each idea/comment.
- The **admin** app has yet to successfully connect to the Heroku DB and provide a CLI for the admin user to interface against.
- The **web** and mobile front-ends still need to implement a button for reporting ideas/users, and the backend needs to implement functionality to satisfy said requests.
- The **web** still must implement viewing profile pages of users via clicking on their ID on a comment or idea.
- The **web** still must implement a file selection button from phase 3.
- The **web** still must implement an alert for when an idea has profanity (based off the backend response)
- The **backend** still must implement interaction with the Google Drive API for file storage.

Contact

(Personnel Information, sort by field confidence)

Chris W. [REDACTED]
Hangyi L. [REDACTED]
Lauren W. [REDACTED]
ThetaBird (Discord: ThetaBird#7701)
Maanasa T. [REDACTED]

Backend

1. ThetaBird
2. Hangyi L.
3. Chris W.
4. Maanasa T.

Web

1. ThetaBird
2. Lauren W.
3. Hangyi L.
4. Chris W.

Mobile

1. ThetaBird
2. Lauren W.
3. Hangyi L.
4. Maanasa T.

Admin

1. Chris W.
2. Lauren W.
3. Hangyi L.
4. Maanasa T.

License

(Relevant Information)

The entirety of the codebase for this project is held under a proprietary license. You are not free to share the code from this project. You may not put it, or any derivative thereof, on any freely accessible version control page (e.g. Github, Bitbucket) or file service (e.g. Google Drive). You may not copy it to a new Bitbucket repository, and give permission to students who are currently enrolled in CSE216, or who may take the class in the future. In compliance with the Family Educational Rights and Privacy Act (FERPA) restrictions, you may not use the code, or any derivative thereof, as a reference or foundation for future entrepreneurial or financial purposes without explicit written permission from every member of the team, as outlined previously.