VibeCheck

https://github.com/comp195/Spring2020Project-vibecheck
Devin French (d_french3@u.pacific.edu)
Seth Kith (s_kith@u.pacific.edu)

Last Revision: 02/19/2020

Table of Contents

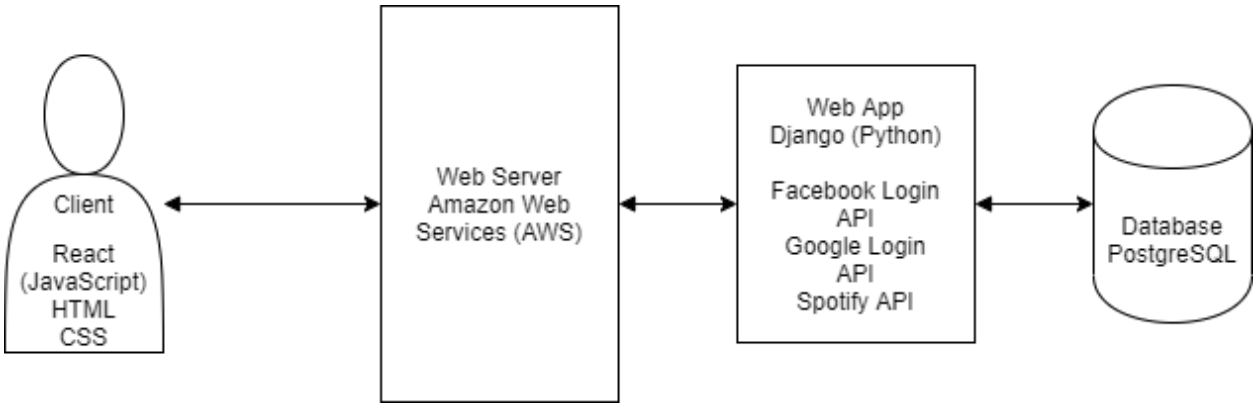
Background:

VibeCheck is a new web social media platform. Users will be able to sign up free or use their current facebook accounts to sign-in. It will provide users a new way to connect with others like friends, co-workers, or sometimes meeting new people online. Users will be able to share thoughts, statuses, locations, music, videos, and many more.

Features:

- Share
    - Music
    - Status - Opinions, Thoughts, etc.
    - Videos
    - Locations
- Custom Profiles - Similar to Myspace profiles
    - Music player
    - Images
    - Themes
    - Edit information
- View
    - Home Feed (Posts from other users)
    - Profiles
    - Friends

System Architecture:



The client's computer will submit HTTP[2] requests to the web server (Amazon Web Services) which will reach end points controlled by the Django web framework. Depending on what end point is requested, the PostgreSQL database will be queried for data or various third party APIs will be queried. The React front end framework will then render the data retrieved from the web server and output it to the user.

Hardware, Software, and System Requirements:

Hardware Requirements: Accessible with any device both mobile and computer that has an internet browser. These devices should be able to connect to the internet using either WiFi, Ethernet connections, or cellular networks.

Software Requirements:

- Facebook login API[3] - Facebook is currently popular amongst everyone. Facebook has created an API for their logins. It's fast and convenient for users to login and create accounts.
- Google login API - Likewise for Facebook, it's fast and convenient for users to login and create new accounts.
- Spotify web API - Users will be able to access their playlist and music
- Web Django (Python)
- Amazon Web Services (AWS)
- PostgreSQL

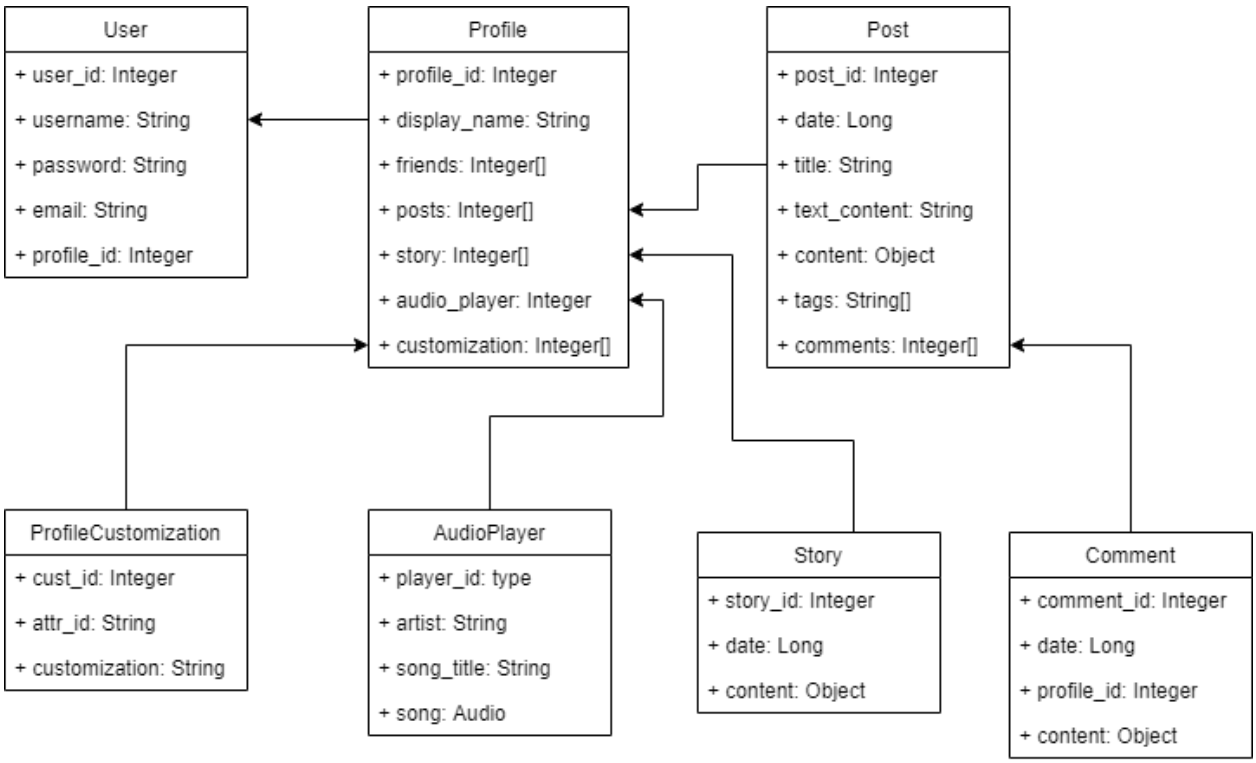
System Requirements: Accessible with any OS that has an internet browser installed.

External Interfaces

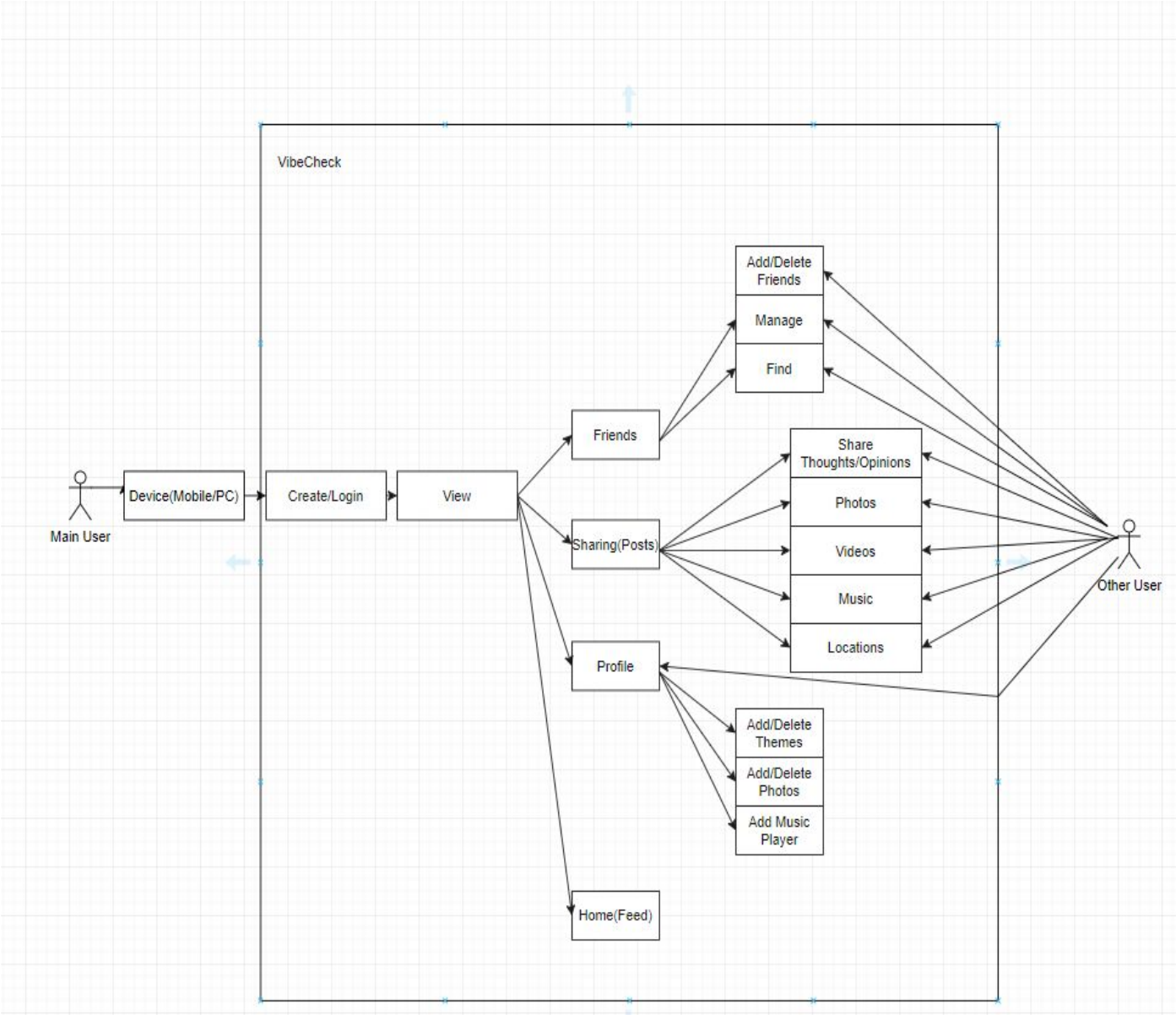
Not applicable.

Software Design

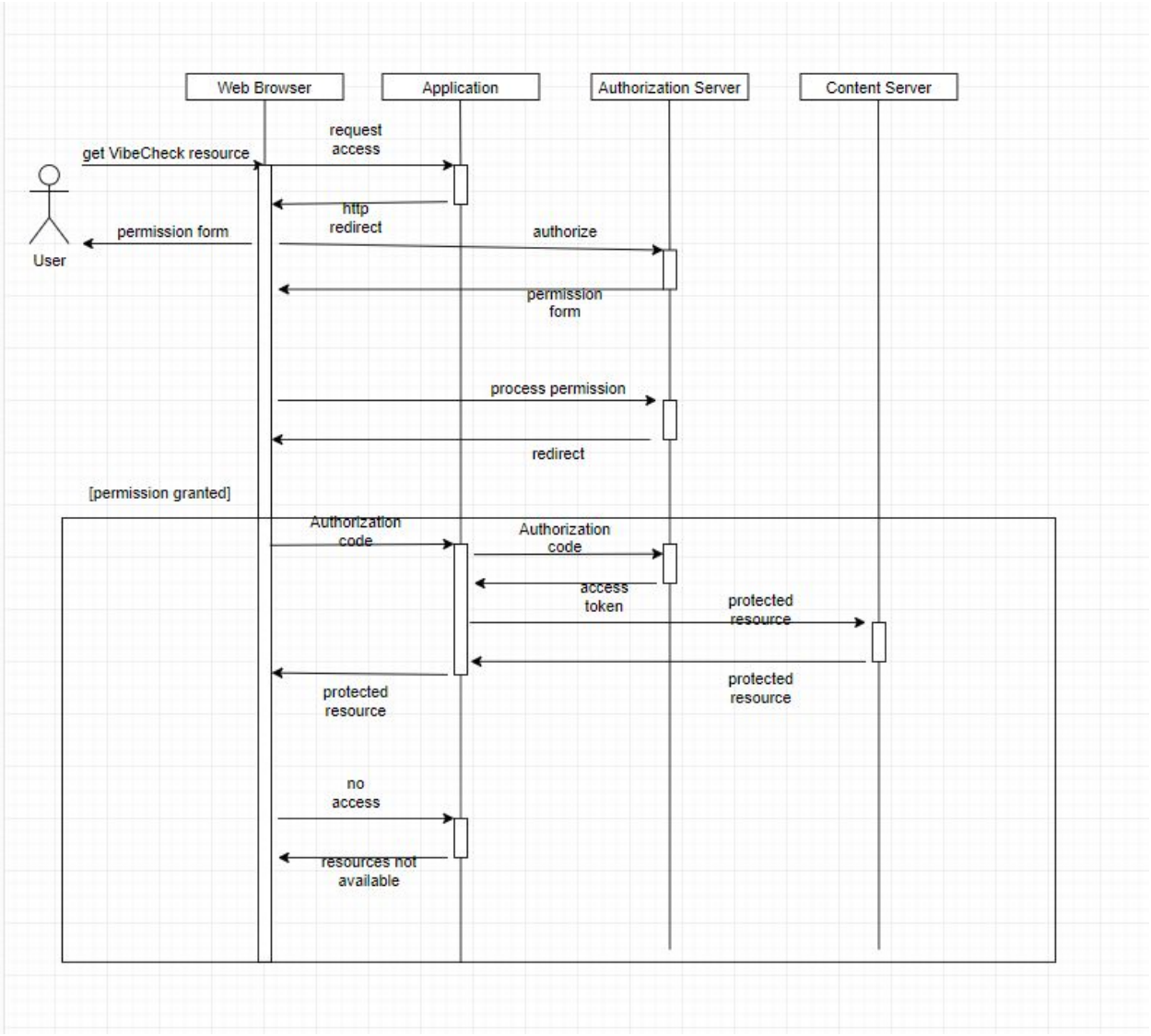
Class Diagram:



Interaction Design:



UML Interaction Diagram:



User Design Interface

Getting started: Users will first access the website through a web browser. Depending if the user has an account, the user will be required to login to gain access. Users will be greeted with a welcome screen and a prompt to either login or sign up. When signing up, users will have the option to either use pre existing logins such as Facebook or Google, since most users have either a Facebook or Google account. If not applicable, users can create an account without the use of an pre existing Facebook or Google account. It will ask for name, birthdate, password, and an email address.

Home: After logging in/signing up, users will be greeted to the home screen. This is where users will see posts, find friends, and interact with others. There will be an option to customize your home feed such as changing it to either dark or light[1] mode.

Posts: A text box will appear in the home feed. Users can share: music, videos, opinions/thoughts, or etc. There will be a character limit of 250 characters, but if the user wants to post something that exceeds the character limit, they'll be given an option to share it as a story. These stories will have a short preview and if other users wish to read to it, they can expand and read it.

Profiles: Users have the option to customize their profiles. They can display who their friends, a music player, and change the theme of their profile with either simple premade themes, or upload pictures for their theme. For advanced users, they can create their own customized profiles.

Glossary of Terms

Dark and Light[1] - Most common background themes. Users seem to like either dark or light themes.
HTTP[2] - HyperText Transfer Protocol, the foundation of data communication for the World Wide Web.
API[3] - Application Programming Interface is an interface or communication protocol between different parts of the program. It allows easier implementation and maintenance within the software.

References