UNIVERSITY OF REGINA

ENSE 477: CAPSTONE PROJECT

SYSTEM AND OBJECT DESIGN

Telport: Sasktel Telecommunications Portal

Authors
Dakota FISHER
200 344 336

Quinn BAST
200 352 973

Supervisor
Dr. Yasser MORGAN

Last Modified
February 24, 2019

Revision History

Revision Version	Revision Author	Revision Date
1.0	Quinn Bast	February 23, 2019

Contents

List of Figures

List of Tables

1 Problem Statement

SaskTel requires a communications portal that can interwork with a Telephony Application Server and our core network to present communications and feature capabilities through a browser. This will allow for the exploration of new communications service models.

SaskTel has been pursuing the deployment of a new communications core along with the Cisco/Broadsoft Telephony Application Server branded as Broadworks. One of the drivers is to enable a richer customer experience through a converged architecture that exposes rapid development to enables new capabilities. Broadworks exposes the Application Programming Interface to access service control tools and user information. These tools and information can be used to create new communications applications or add additional value to existing applications.

The main objectives for SaskTel is to gain exposure to new and innovative communications service experience for our customers and to promote the potential internally for aligning resources, time and effort in enabling applications.

2 System Architecture

The application that is being created is a web server which will enable customer to configure various settings about their account's calling settings. This will enable customers to be able to seamlessly change their phone's settings without the need to contact any support lines. Furthermore, SaskTel has realized the need to convert their services to the world wide web. Customers are moving away from home phones because of the development of mobile technologies. Accessibility to your phone no matter where you are is becoming a necessity for customers. Thus, SaskTel has the need to be able to make phone calls from the internet browser. This requires interfacing with SaskTel's WebRTC gateway in order to make communications possible between a web browser and a mobile device.

SaskTel has a large infrastructure already established and thus, have many of the required components for this project already in place, the missing components are the client-facing components of the application. The main components are the IMS core, Home Subscriber System (User Profile Repository), Telephony Application Server, WebRTC Gateway, gateway access to the PSTN, and SIP clients. These network components will interface with our architecture as shown in the image below.

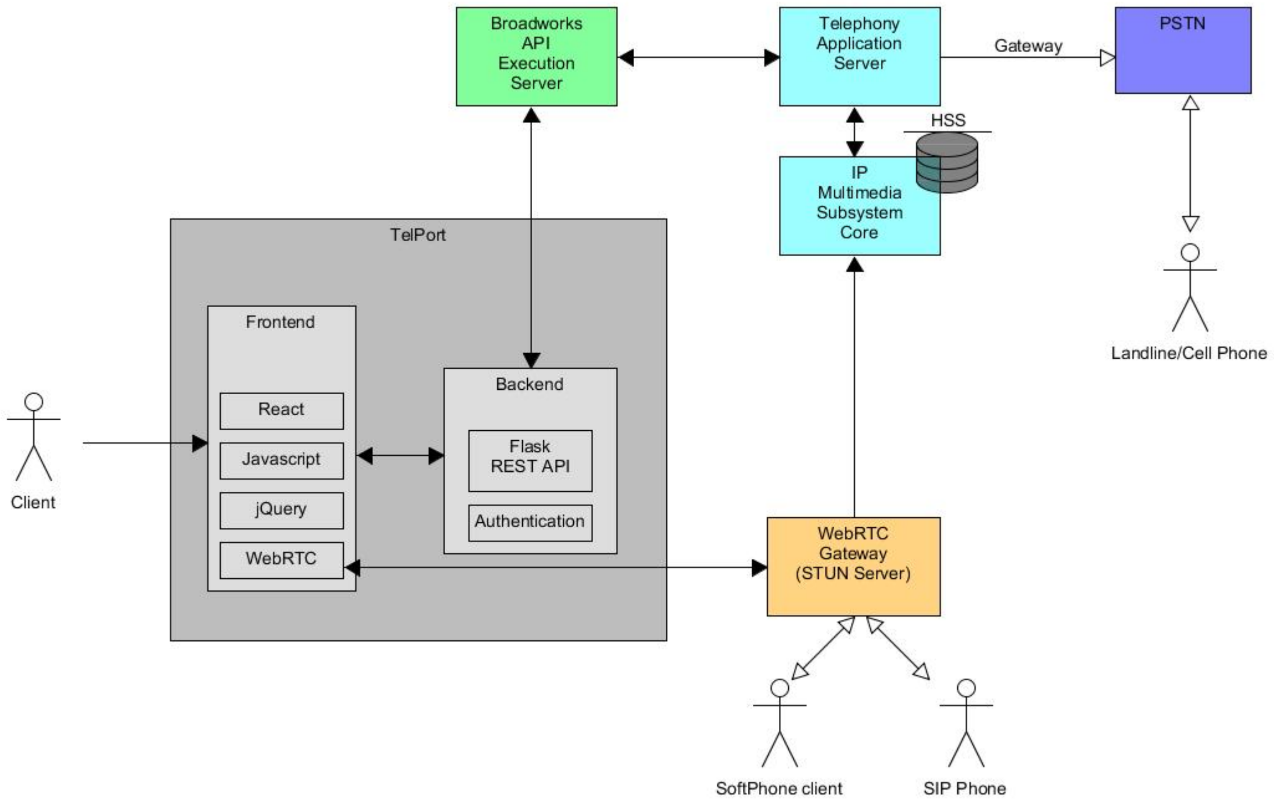


Figure 1: TelPort Network Architecture

2.1 Major SaskTel Components

2.1.1 BroadWorks API Execution Server

BroadWorks VoIP communications application server enables service providers to offer a comprehensive portfolio of business and consumer communications applications and value added applications from a common network platform. BroadWorks delivers communication solutions that integrate video, fax, voice and email communications for businesses and consumers worldwide whether through IP PBX/Centrex, Mobile PBX, Business Line, Trunking and consumer solutions. BroadWorks goes beyond connecting an IP pipe to a user. The advanced calling features allow service providers the ability to offer innovative, compelling, revenue-generating solutions to both business and residential customers.

The BroadWorks Application Server operates at the core of BroadWorks and is responsible for the execution of all enhanced personal and group features. The server's database maintains user and group profiles, as well as service and subscription data. The Application Server supports a variety of access side protocols in including SIP and MGCP.

In addition to the self-care web portal and a variety of desktop client applications, BroadWorks provides a very rich set of open programmable interfaces called the Xtended Services Interface (Xsi) that allows operators, partners and even end users to build their own niche applications around the services offered by BroadWorks. The Xsi is a RESTful HTTP based interface that uses simple commands to enable service configuration, click to dial and other call control primitives over the web. TheBroadSoft Xtended program gives operators access to a growing community of software developers that have adopted the Xsi and are building innovative applications and web mashups based on the BroadWorks platform.

2.1.2 IP Multimedia Subsystem

IMS (IP Multimedia Subsystem – architectural framework) provides control functions within the Telecom network core, allowing for shared use of infrastructure and services across fixed and mobile technologies. IMS provides a path to transition the core of a telecom network to IP. IMS will be used in the Demo to facilitate subscriber registration onto the network, allow the use of directory numbers to address and direct calls and to provide a bridge between the internet and the legacy PSTN.

SaskTel is pursuing the installation of IMS as a foundation to provide broader services, converged services (fixed and mobile), and to generate service differentiators to maintain a competitive edge. SaskTel will strive to align the majority of communications services to be delivered from a single (geo-redundant) core. As solutions evolve, it is expected that the core IMS functions will reside on common computing hardware. SaskTel will leverage IMS to function as a bridge between legacy communications and communications on the internet, thus migrating the PSTN to the internet.

2.1.3 WebRTC Gateway

Within SaskTel's environment, a WebRTC Gateway will be added to provide an interface between the use of web centric protocols and Telecom IP network protocols. Simply stated, the WebRTC gateway gives the ability for a service on a browser to communicate with IMS. This allows the bridging between WebRTC on the Web and IMS subscription, IMS service and access to the PSTN. By using this interconnection, a user can be identified with a Phone number, access a network address book and place calls between the web and the PSTN.

2.1.4 Home Subscriber System

The Home Subscriber System (HSS) is accessed by the IMS Core in order to obtain information about a user. This is how the BroadWorks API accesses information about the various users and this is also how our application is able to access user information without the need of a database. Because everything is stored externally, our application never needs to store any information about a user.

2.2 Major TelPort Components

2.2.1 Flask REST API

Without our application, the frontend can communicate directly with the BroadWorks API, however, in order to do so, the frontend needs to keep track of a user's login tokens. This exposes the user's tokens and credentials to the frontend in an unsafe manner which is vulnerable to XSS (Cross-site scripting) attacks. As a result, our application makes use of an additional layer of security through our backend before being able to contact the Broadsoft server.

The REST API for our backend is very small, containing only 4 endpoints. The endpoints provided by our backend server are described in detail as follows:

```
POST /rest/login
```

Description: Allows the user to login to the application.

Request Body:

Parameter	**Type**	**Description**	**Example**
username	String	The user's username to login with.	SampleUser123
password	String	The user's password.	hunter123

```
POST /rest/logout
```

Description: Logs the user out of the application. NOTE: Requires a valid token.

Request Body: None

```
POST /rest/token/refresh
```

Description: Refreshes the user's session token to further their login session. This is automatically requested by the frontend when needed. NOTE: Requires a valid token.

Request Body: None

```
POST /rest/broadsoft
```

Description: Accesses the BroadWorks API Execution server. NOTE: Requires a valid token.

Request Body:

Parameter	**Type**	**Description**	**Example**
endpoint	String	A BroadWorks REST API endpoint to access	/user/<user>/services
method	String	The method to access the BroadWorks REST API endpoint through.	"PUT", "GET", "POST", etc...
data	Xml String	An XML formatted string to send to the BroadWorks API. The data to send is as defined int he BroadWorks API.	<?xml version="1.0" encoding="ISO-8859-1"?> <DoNotDisturb> <active> false </active> <ringSplash> false </ringSplash> </DoNotDisturb>

2.2.2 WebRTC

In order to enable the capability to make calls from the web, WebRTC is required for our web application to interface with an SIP client. There is quite a bit of overhead to convert a WebRTC session to an SIP session, however. Therefore, the use of a JavaScript library JsSIP was used in order to help make this process easier. The use of this framework requires some basic server configurations that were provided from SaskTel. Once this information was obtained, making calls from the web browser was quite simple to perform.

3 Class Diagrams

3.1 Backend Class Diagram

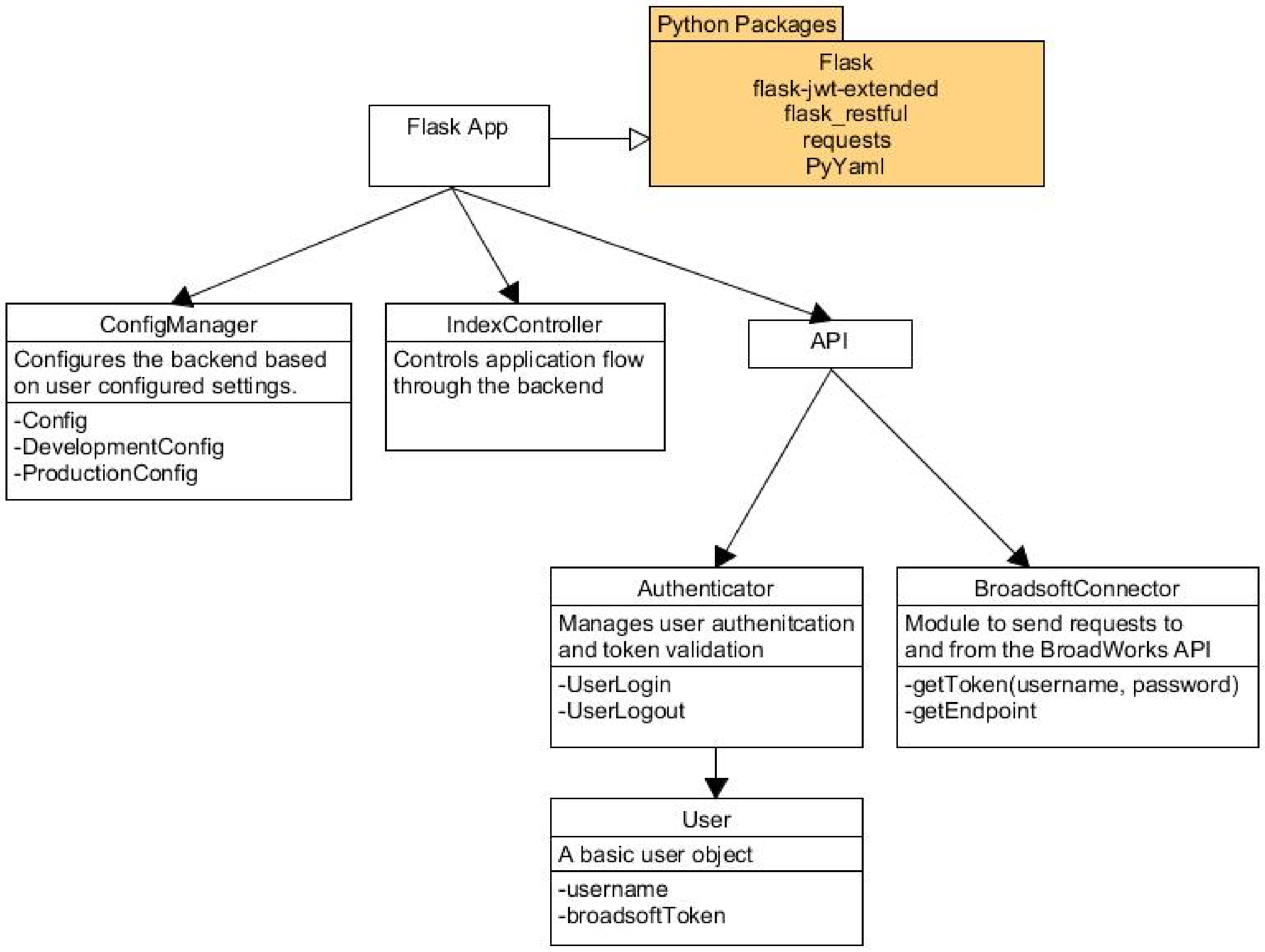
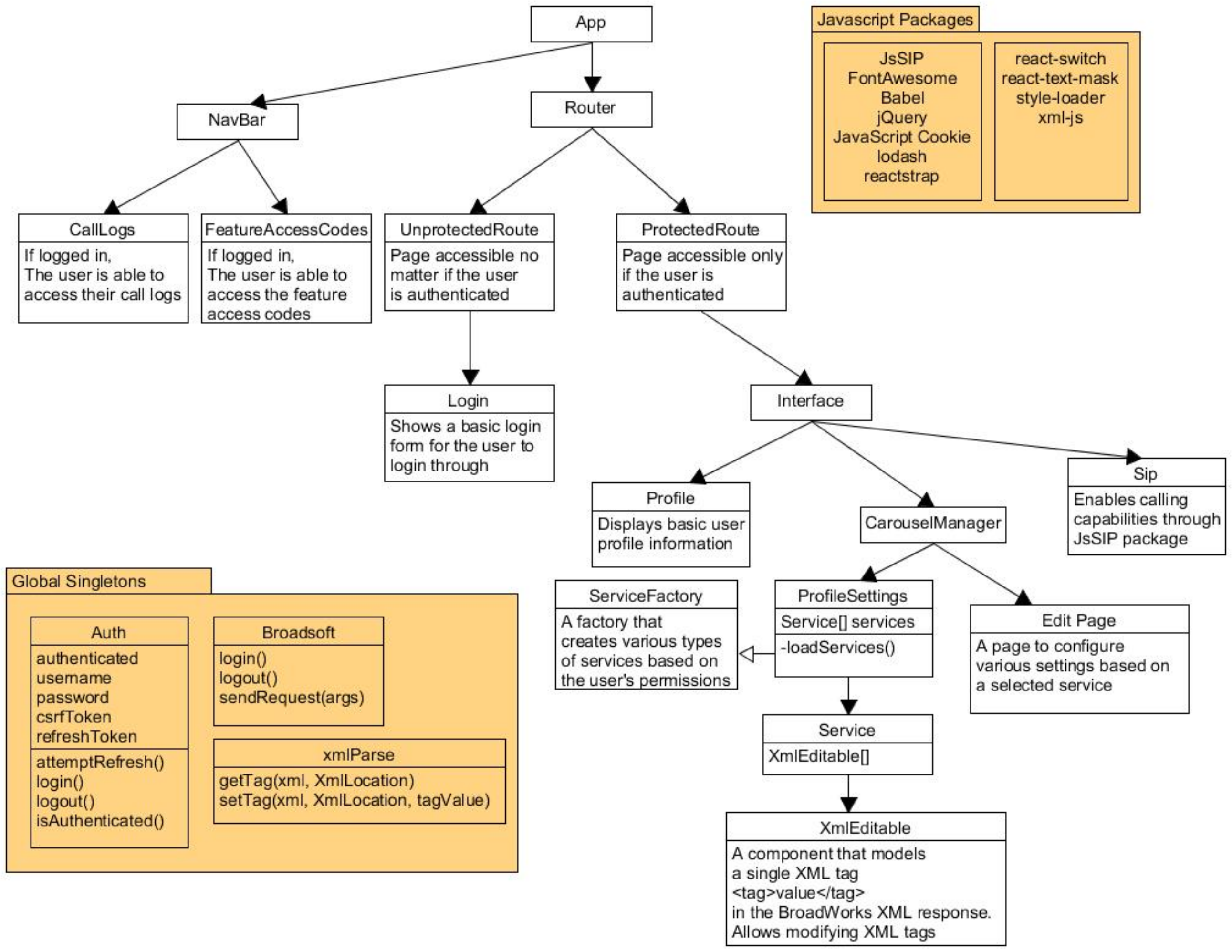


Figure 2: Backend Class Diagram

3.2 Frontend Class Diagram



App
NavBar
Router
Javascript Packages
JsSIP
FontAwesome
Babel
jQuery
JavaScript Cookie
lodash
reactstrap
react-switch
react-text-mask
style-loader
xml-js
CallLogs
If logged in, The user is able to access their call logs
FeatureAccessCodes
If logged in, The user is able to access the feature access codes
UnprotectedRoute
Page accessible no matter if the user is authenticated
ProtectedRoute
Page accessible only if the user is authenticated
Login
Shows a basic login form for the user to login through
Interface
Profile
Displays basic user profile information
CarouselManager
Sip
Enables calling capabilities through JsSIP package
Global Singletons
Auth
authenticated
username
password
csrfToken
refreshToken
attemptRefresh()
login()
logout()
isAuthenticated()
Broadsoft
login()
logout()
sendRequest(args)
xmlParse
getTag(xml, XmlLocation)
setTag(xml, XmlLocation, tagValue)
ServiceFactory
A factory that creates various types of services based on the user's permissions
ProfileSettings
Service[] services
-loadServices()
Edit Page
A page to configure various settings based on a selected service
Service
XmlEditable[]
XmlEditable
A component that models a single XML tag <tag>value</tag> in the BroadWorks XML response. Allows modifying XML tags

Figure 3: Frontend Class Diagram

4 Frameworks, Libraries, and other tools

4.1 Backend python packages

4.1.1 Flask JWT Extended

Flask-JWT-Extended is a python library that adds support for using JSON Web Tokens (JWT) to Flask for protecting views. This allows our application to send information to the frontend while also only allowing authorized users to access the server. Additionally, Flask JWT Extended provides many helpful features built in to make working with JSON Web Tokens easier and more secure. This includes creating tokens from complex objects or complex object from received tokens, creating refresh tokens, token freshness and separate view decorators to only allow fresh tokens, token revoking/blacklisting, storing tokens in cookies and CSRF protection.

4.1.2 Flask Restful

Flask-RESTful is an extension for Flask that adds support for quickly building REST APIs. Flask Restful makes creating REST endpoints and server responses extremely easy. Flask Restful requires minimal setup and only requires you to define the URL's for each REST endpoint and what function/class methods correspond to the endpoint. Flask Restful is extremely lightweight but extremely powerful.

4.1.3 Requests

Requests is a python library for sending HTTP requests from a server. Because our backend needs to interface with the BroadWorks API Execution server, using the Requests library is required for our application in order to talk to the BroadWorks API. Compared to the built-in python http library, Requests is significantly easier to use. Because of this, using this library saves a significant amount time and headache compared to using the default library.

4.1.4 PyYaml

Flask uses Yaml configuration files in order to setup some of its components. Because we wanted to allow our Flask application to output logs from the server, we needed to customize the Flask logging system which requires a Yaml configuration file. Using PyYaml allows python to read the Yaml file and turn it into a python object that is able to be easily parsed by Flask.

4.2 Frontend Javascript libraries

4.2.1 JsSIP

WebRTC allows communication across two different browsers through the user of web sockets. In order to create a call from our web application, the client needs to be using a WebRTC connection, however, the RTC connection needs to be converted into the SIP protocol in order to communicate with a user's cellular device. Converting between RTC and SIP protocols is a complicated task, luckily, JsSIP does it automatically. By simply configuring the SIP settings, creating a call from the web is extremely easy with the use of this framework.

4.2.2 FontAwesome

Font Awesome is a collection of vector icons and logos that are free to use for the web. The use of icons within a web application lets a user more easily understand what a button or action will do based on the icon that is provided. This provides consistency between websites and also makes using the application easier for the user as they can simply recognize the icons that perform specific actions instead of having to search around.

4.2.3 Javascript Cookie

Javascript Cookie allows Javascript code to access the browser's cookies. Because of the use of Flask JWT Extended in the backend, CSRF tokens are sent in cookies to the browser. Accessing the cookies from the browser is possible, but somewhat complicated. Javascript Cookie makes accessing browser cookies much easier, and allows future Ajax requests to be configured with the cookies that have been sent from the backend to validate the user.

4.2.4 ReactStrap

Reactstrap is a javascript package to allow the use of Bootstrap 4 components within React. Bootstrap makes styling a website extremely, taking time away from styling CSS sheets to get the website how you want it to look. By using boostrap, only minor CSS changes need to be made while the default components will position elements in the website with minimal configurations.

4.2.5 React Switch

Toggle buttons and sliders are not a default element in HTML5 and are not an included element in bootstrap. However, in our application, toggling various settings between "On" and "Off" is a large part of the application. Toggle buttons are the easiest way to indicate if something is on or off. React Switch provides the ability to put a toggle buttons on a react page. This allows us to easily show if a setting is on or off.

4.2.6 React Text Mask

When a user types into a textbox, their input is unreliable. They might make a typo or format the text improperly. This is where React Text Mask is extremely useful. Typing a phone number might require an area code, or it might require a "+1" at the front. In order to eliminate any confusion for the user, React Text Mask allows us to specify a mask: (___)___-____ and this mask guides the user's input. The user cannot break elements from the mask, ensuring that their input is properly formatted. The users are forced to input a specific character set which is verified based on a RegEx. This allows us to ensure that a user is inputting the correct information and takes any confusion away the users because they know the information they are entering is correct. Of course, this does not replace backend security and validation, but simply provides end users with a better experience.

4.2.7 Xml-js

Xml-js allows converting XML into Javascript or JSON objects. Because the BroadWorks API sends an XML response, we need to be able to parse, read, and modify XML. This library allows parsing XML objects in order to read values to display to the user, and modify values to send back to the BroadWorks API. This package is a necessity for our application as 80% of the application needs to examine XML data.

4.3 Other tools

The following sections denote tools that are utilized or were considered to be of use during the project's development. Programs and tools, alongside their effective alternatives are discussed to get an understanding of what, and why tools were picked or not picked to be used to benefit the work flow of the project. Every reference to an application in the following sections has a direct hyperlink reference if clicked on using an electronic medium.

4.3.1 Github

Git is a powerful source version control solution that allows for multiple developers to collaborate on a project. The Git provider used during this project is Github. Not only does it allow for collaborative features, it also allows for parallel streams of development of many features. The tool also allows for public display of the project, to allow for open source development if the repository is not kept private. A key feature and benefit of using Git is that it allows for projects to commit checkpoints and milestones in development, such that if an update isn't beneficial, it can easily be reverted to a previously confirmed viable version.

4.3.2 Toggl

Toggl is a time tracking web application that allows teams to track their hours along with the project, and tags that correspond to the work that they are doing. This allows for us to track documentation time seperate from programming time and research time. While still keeping a total cumulative track of our time spent on the project. Ideally, the time tracking entries are relative to our commits on Github. Of the options online at the time, this is the only option that we felt met the requirements we needed to easily and painlessly manage our project time.

4.3.3 GitKraken

The tool GitKraken is a powerful Git version control client that has an easily manageable interface. It also has the ability to see how each branch correlates to one another, and resolve any conflicts within the application directly. There isn't any particular reason to use this tool over another alternative such as GitHub Desktop, SourceTree, or GitBash, but it definitely was a nice discovery for our quality of life work flow.

The company that created GitKraken, also has a product called Glo Boards. The application is functionally equivalent to Trello. Both programs allow for a team to manage issues on a GitHub repository easily and visualize changes into KanBan board styles. The decision to use a KanBan board software instead of just the issue tracking software on GitHub was because it provides additional work flow improvements while still allowing GitHub issues to still work normally.

4.3.4 Google Drive

For file containment and collaborative efforts, Google Drive was utilized to create rough drafts and documentation. The documents on Google Drive are deliberately kept out of the project as they provide no benefit and are unpolished in comparison to the products that should be present in the project. As files become absolute, and documentation gets refined and compiled into information that has enough value to be displayed, it is transferred to GitHub to be displayed alongside other documentation. Other alternative data solutions that could have been used involve OneDrive and DropBox, but the simultaneous access and collaborative nature of the Google Drive made it an ideal environment for rough collaborative drafting.

4.3.5 Discord

Being able to communicate with each other during the project is essential for collaboration. Discord allows us to keep each other up to date on our day-to-day progress and ensures that we can both stay in contact throughout the creation of the project. Discord provides a significant amount of functionality to enable functionality. This main functionality of discord is its text and voice chat capabilities. Keeping each other up to date is easy through text chat and allows us to send quick ideas back and forth while developing the project. Furthermore, voice chat allows us to communicate more vital information at a faster speed. While in a voice chat, Discord provides the ability to share screens between users, making discord extremely useful for paired programming sessions and bug fixing.