RGenius Analytics

RGenius

Evan Murray, Team Lead (evanmurray2@gmail.com)
Nicholas Duncan, Scribe (ncduncan@ucsb.edu)
Yash Rane (yash_rane@ucsb.edu)
Nikita Tyagi (nikitatyagi@ucsb.edu)
Wei Tung Chen (weitung_chen@ucsb.edu)

Introduction

Motivation:

- **What problem the project is solving?**
    - The problem we must solve is finding new ways to analyze data collected by a network of routers and displaying the data in a way that shows the important trends and any issues that need to be addressed.

- **Why is the problem important?**
    - The dashboard will enable engineers to quickly pinpoint router problems and outages, and allow analysis of all routers which assists greatly in the development and maintenance process. This could potentially lead to new use cases and software implementations to improve the customer's experience with the routers, and improve the routers themselves. Customers can also see distribution details of the devices, including geographic and usage details.

Background:

- **About our project**
    - Currently, all SmartRG routers check in once daily with a cloud server and push identifying information and some metrics into a database. The data is transferred in json format. The cloud server converts and pushes that data into a mongo database. The 'customer' information comes from a SalesForce database that is currently a static csv file on the cloud server (mac customer mapping). Currently 400 routers emit health metrics and geographical data. But, there are over 100k routers out there, so those need to be upgraded somehow to emit the health data to be used in the analytic modules.
    - They have tried basic analytics using MongoDB Charts which can be used for all devices or specific devices, so these could be applied with filters to see data for specific customers or any of the other fields the checking provides.
    - As of now, their server is behind a firewall so their data can only be accessed through their company's IP addresses, so it is not True Cloud Computing.

- **How have others solved this problem?**
    - Typically, this problem is solved in unique ways. There are companies that provide analytics by analyzing databases, and then using their software suite to display the analytics they come up with in interesting and useful ways. We plan to employ this ourselves by analyzing the data given to us, and developing our own dashboard for displaying the analytics that we come up with.

- **What issues have they had?**
    - Typically, the issues that come along with developing analytics are those that come with datasets, such as bad/noisy data. There are also issues in that it can be hard to come up with analytics that have clear use cases and benefits. We plan to overcome these both fixing up the database ourselves, and by talking directly with the company about specific use cases they would like to see and their customers would like to see.

Goals:

- **Backend**:
    - Analyze the data from their databases and create a server that will call different libraries to display the appropriate data
    - Specifically, our api will handle calls that organize the data for 4 different aspects of our frontend:
        - Charts
        - Maps
        - A table of Routers
        - Filters
    - And will also handle calls to our database
        - Authentication (Login/Logout/Session)

- **Frontend**:
    - Display the analytics on a cloud-based dashboard in an informative and creative way for both world and ISP views
    - Display maps, charts, and have a table of routers that is filterable

- **How do these goals solve our problem?**
    - These goals solve our problem pretty directly. Our goals will be reached when we are able to integrate the frontend and backend systems directly, we will have reached our overarching goal of displaying useful analytics on the web. From there it is coming up with more and more useful analytics.

Innovation:

- **How will we be reaching our goals?**
    - Through the use of ReactJS for the frontend, Django for the backend, Mapbox and ChartJS to display the data, daily scrum meetings, and an agile development

process, we will be able to implement these features and analytics, and do so according to a prioritized list of use cases.

- **What is innovation, the science, and new core technical advance?**
    - We will be creating new analytics that will clean and organize SmartRG's datapoints into usable information for both SmartRG employees and customers. The information this provides will provide new visual ways for the employees to assess their devices and for the customers to keep track of their own customers' usage that they don't have access to currently. The information in the database is currently behind a firewall, so the customers don't have access to anything so the web interface would provide them with a lot more information.
    - As mentioned, all of SmartRG's information is in the cloud but behind a firewall, so we aim to add security through authentication so that they can shift their technology to True Cloud Computing, adding more flexibility and scalability.

Misc:

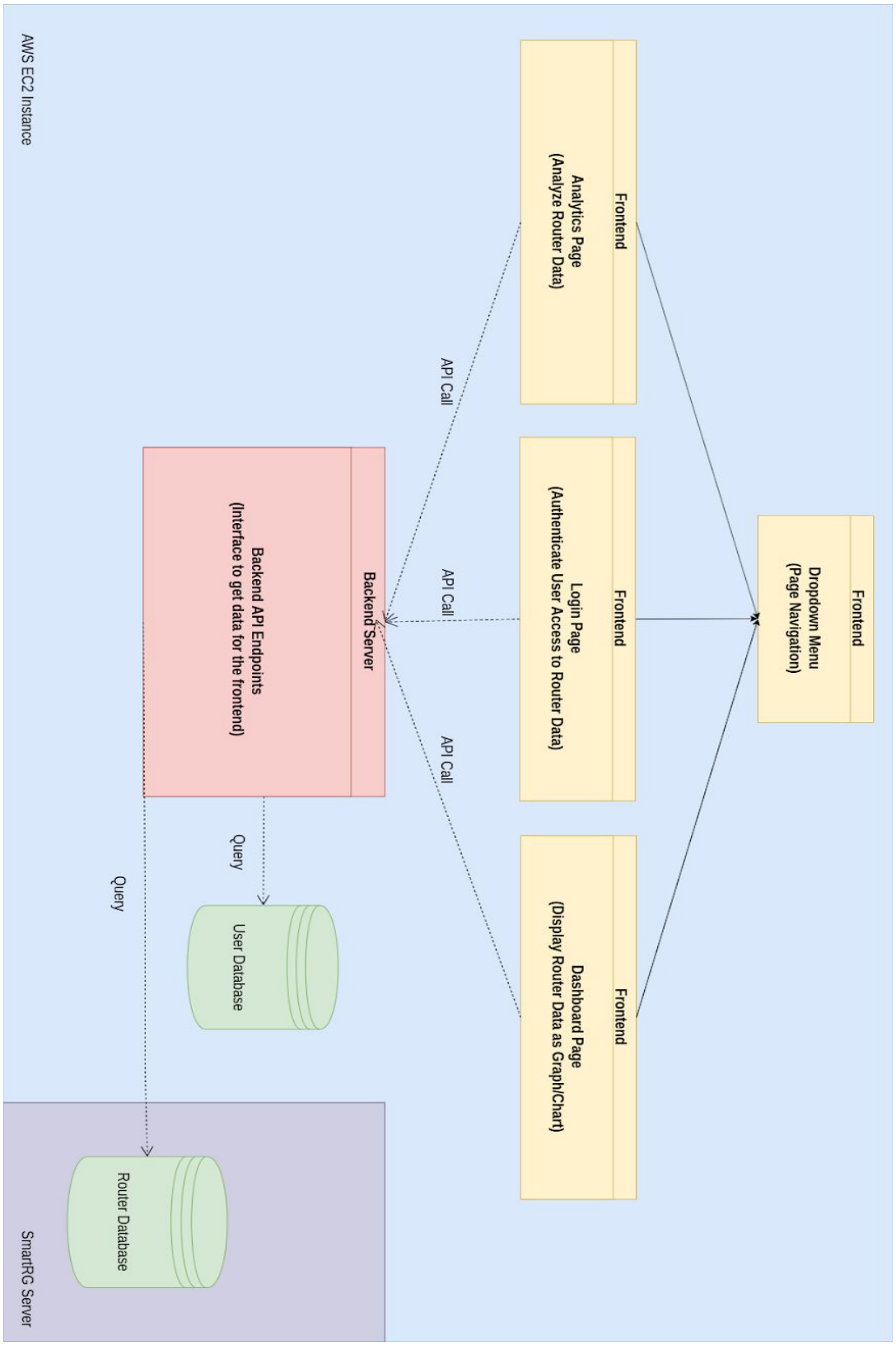
- **Specification**: Develop modules for device analytics that will run in server using company's existing databases
    - Analytics modules:
        - Charts:router models chart, router releases chart, crashes per router chart, active router models chart, active router releases chart
        - Maps: Heatmap of router temperatures, Map of router kernel crashes, Map of locations of router, Map of active router models
            - Be able to Click on routers in map to display in table
        - Table: Sortable grid of router information, and filters to display data in the table in a custom way
    - Create a web services with login/logout, user interaction, and security (user auth)
    - Display the analytics on dashboard for SmartRG perspective and their customers' (ISPs) perspective

- **Design**:
    - Will have a web interface for customers to see different analytics from different views
    - Define API for interactions between frontend web interface and backend server
    - Take data from the database and clean and analyze it in the server
    - Define a set of analytics to implement using different charting and mapping libraries to display the data in a user-friendly way

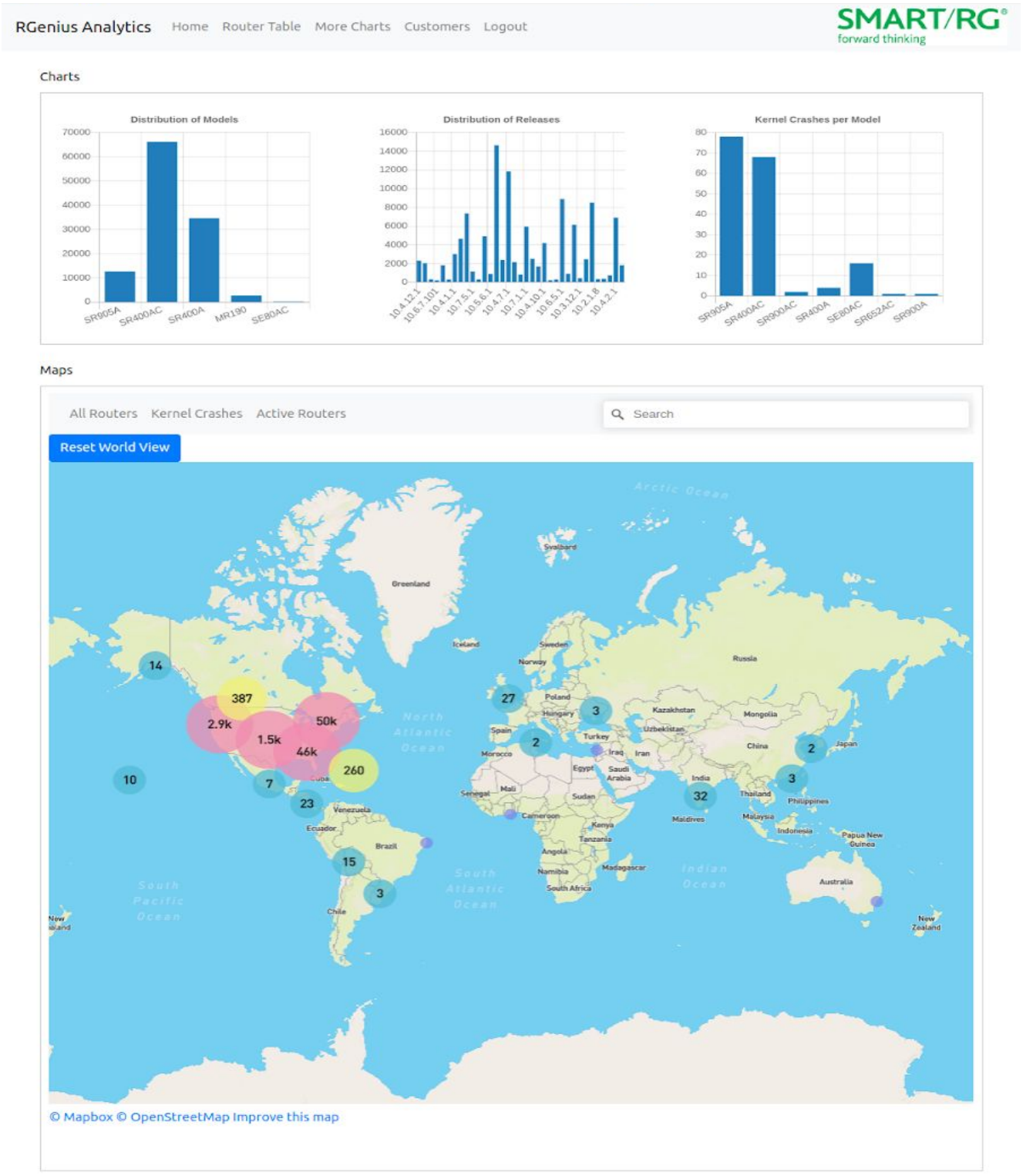
- **Assumptions**:

- Eventually, all of their routers will be emitting the health and geographical data so that the analytics and models we create can provide helpful insight for almost all, if not all their devices.
- We will be constantly designing, developing, and testing according to the Agile Development Cycle so there are bound to be changes throughout the course of the project.
- We will add security and deploy both the backend and frontend on AWS so that the company can shift toward True Cloud Computing.

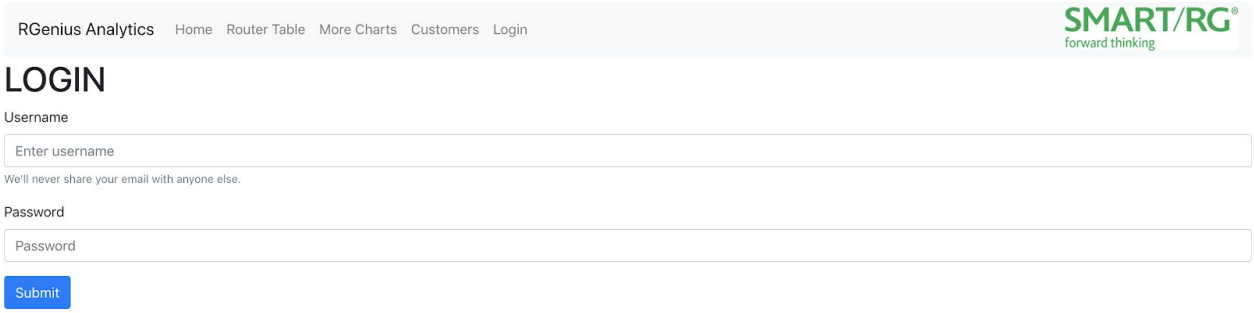
High Level System Diagram



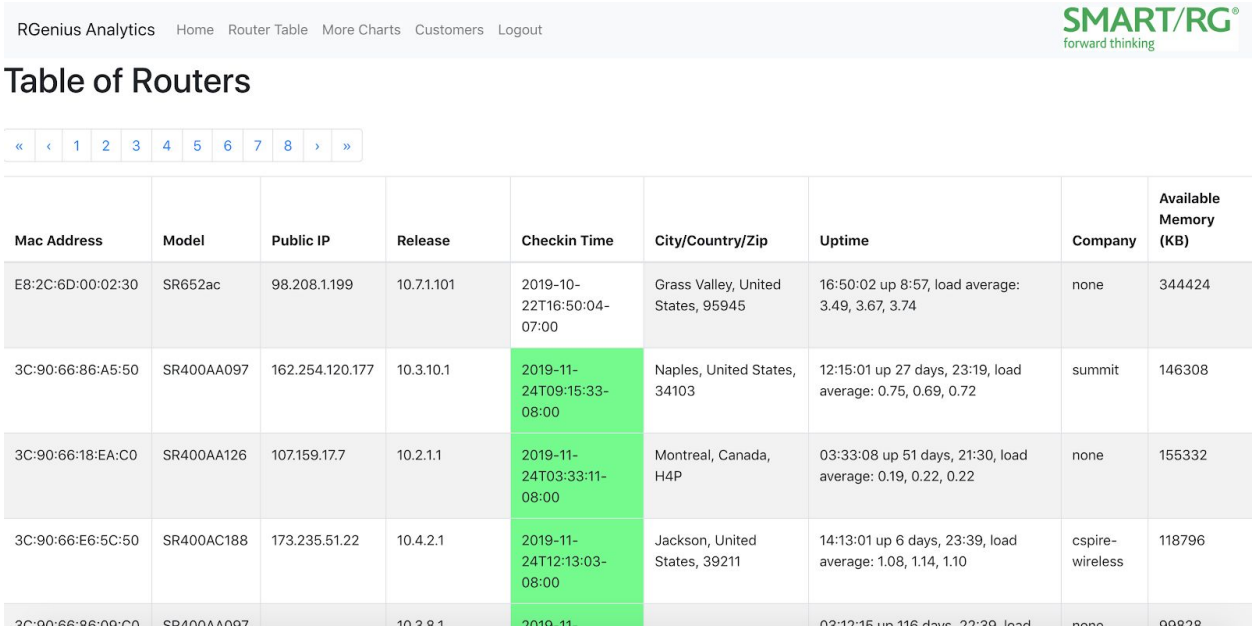
UI Design


RGenius Analytics
Home
Router Table
More Charts
Customers
Logout
SMART/RG
forward thinking
Charts
Distribution of Models
SR905A
SR400AC
SR400A
MR190
SE80AC
Distribution of Releases
Kernel Crashes per Model
SR905A
SR400AC
SR900AC
SR400A
SE80AC
SR652AC
SR900A
Maps
All Routers
Kernel Crashes
Active Routers
Search
Reset World View
Arctic Ocean
Greenland
Iceland
Svalbard
Sweden
Norway
Russia
Poland
Hungary
Spain
Turkey
Kazakhstan
Uzbekistan
Mongolia
China
Japan
Morocco
Iraq
Iran
Egypt
Saudi Arabia
India
Thailand
Philippines
Malaysia
Indonesia
Papua New Guinea
Australia
New Zealand
Maldives
Senegal
Mali
Sudan
Cameroon
Kenya
Tanzania
Angola
Namibia
South Africa
Madagascar
Venezuela
Ecuador
Brazil
Chile
North Atlantic Ocean
South Atlantic Ocean
South Pacific Ocean
Indian Ocean
14
387
2.9k
1.5k
50k
46k
260
10
7
23
15
3
27
3
2
2
3
32
© Mapbox © OpenStreetMap Improve this map

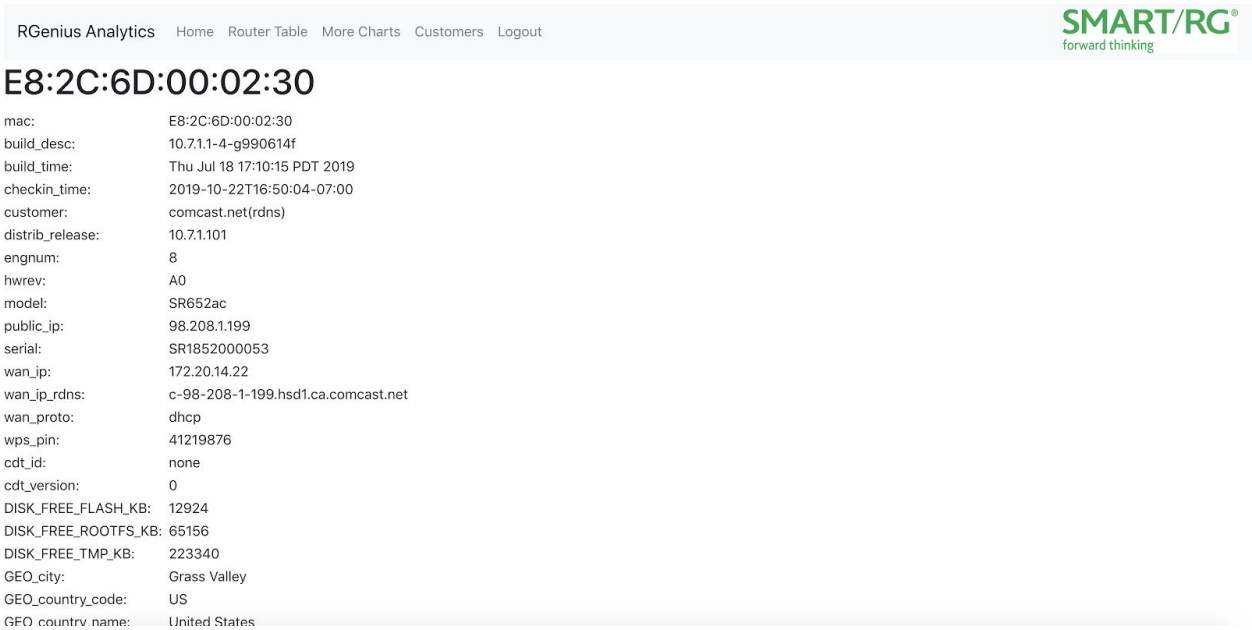
Above is our home page which will be the default page that opens in a user's login information is saved on the browser.



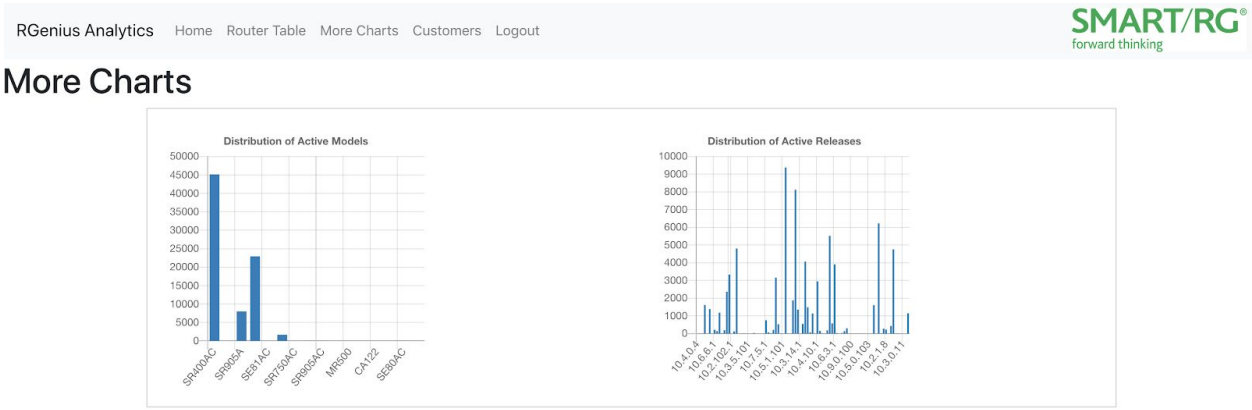
If the user is not currently logged in, they will be directed to this login page.



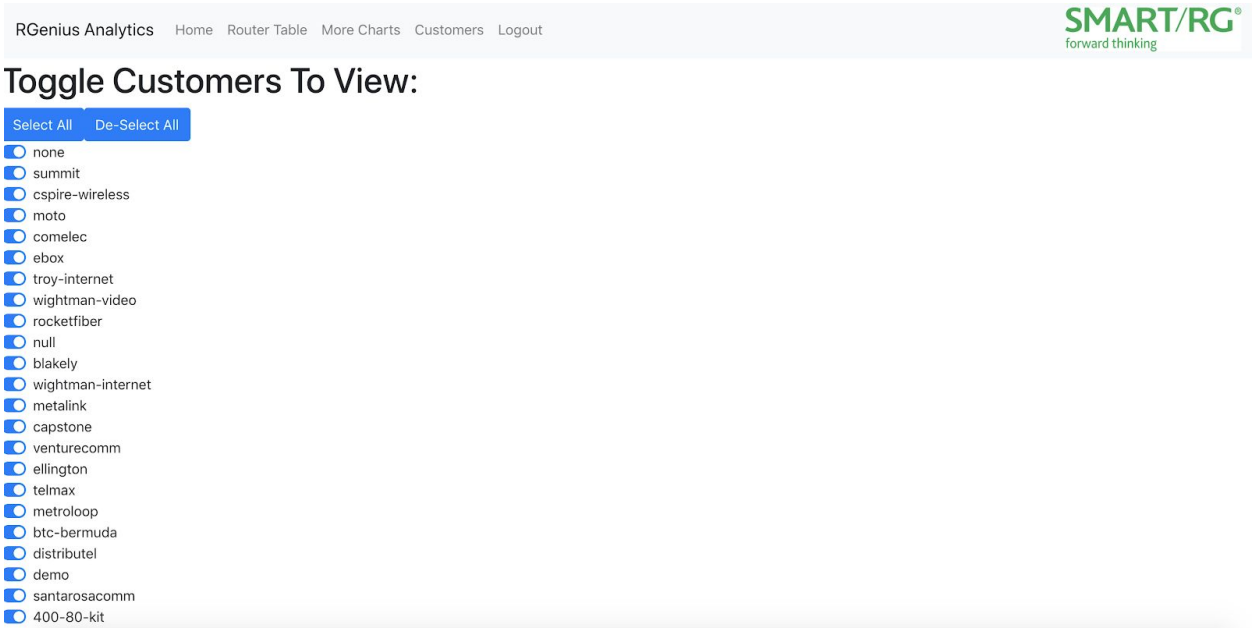
This is our router table which displays all routers linked to an account.



Upon clicking on a MAC address in the router table, the user can view all of that router's information.



This is the page that pops up when user clicks on the More Charts page on the navigation bar. This page will have more charts displayed as we develop.



This is a customer filter page only available to SmartRG employees, a specific user group that can view all of the routers.

User Stories

1. *Use case*: As a developer, I can develop and run all of our codebase on the cloud, so users can access our website through an ip address.
   Acceptance Test: Given an AWS environment, when running our application, then both the frontend and backend code are either running on the cloud or running in a virtual environment that can be easily moved to a cloud server, and users can access our website directly.

2. *Use case*: As a User Information Database, I can store user information such as permission and login credentials so that users can properly access their account.
   Acceptance test: Given user account information, when a request is made to store or retrieve data during user login or create account, then the data is added or retrieved from the database.
   Github Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/54854ea8d56898c3faea6b0778a49d9280f9c758
   Test Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/445e4dba2424acc2e5fe63084b4ee808092e06e3

3. *Use case*: As an HTTP Request, I can retrieve data from a cloud server so that it can be used in the frontend.
   Acceptance test: Given an API request to the server, when the frontend requests a specific endpoint, then the data is transmitted in JSON format.

4. *Use case*: As a user, I can login to my account with the appropriate user permissions so that I can access the dashboard services.
   Acceptance Test 1: Given a valid username and password, when a login request arrives, then an authentication token is returned that will be added to all future requests requiring authentication (test: create an authenticated user and see if they can access a protected endpoint)
   Acceptance Test 2: Given an invalid username and password, when a login request arrives, then an error message is returned declaring login attempt invalid (test: create a false username and password and ensure not authenticated/can't access protected endpoint)

Github Commit:
https://github.com/evanmurray2/RGenius-Analytics/commit/775b56afa94843fb961c57af416a4eebb024f558
Test Commit:
https://github.com/evanmurray2/RGenius-Analytics/commit/4063b45c5cc69142c35b3d6911775ba0e2d92458
Frontend test commit:
https://github.com/evanmurray2/RGenius-Analytics/commit/eebaf4af5a6a60a5a083534d8984499d2755a749

5. *Use case*: As a router I can upload my data each day so that it can be stored and used to compute analytics.
   Acceptance Test: Given a Smart/RG router, when a router sends its daily update data, then database will accept the data and store it in mongodb and show new data points each day as the routers check in.

6. *Use case*: As a user, I can view regular updates on my dashboard so that I have the most recent and accurate data.
   Acceptance test: Given the daily check-in process, when the database is updated with the recent data, then the frontend receives it and at least one of the charts will show updated data automatically.
   Github Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/2b656eea3342a9039a2847a9c3b6f6c78a9e5a7a
   Test Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/c837d8d8cfd41432e48e047093dc7dc713aa93c6

7. *Use case*: As a SmartRG employee, I can access the world view and analytics so that I can see all of the routers' data.
   Acceptance Test: Given the username and password successfully logged in to the website with the correct user permissions, when the employee accesses the world view and analytics, then they will be able to see all the geographic and health data for all the routers on the dashboard and all the updated and desired analytics for all the routers on the analytics page.
   Github Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/9bf625b5b374c5611f4390b3496da9f8f711631e

Test Commit:
https://github.com/evanmurray2/RGenius-Analytics/commit/c201f98b45cff391601ea202a48fe8fcd38c405a

8. *Use case*: As an ISP/ISP customer, I can view my analytics dashboard so that I can see the status of my deployed routers.
   Acceptance Test: Given a user that has logged in successfully, when they go to the dashboard page of the website, they will see a view denoting various analytics like downtime, temperature, kernel crashes etc. (test: demonstrate a logged in user navigating to the analytics page of the website and showing different analytics views)
   Github Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/b8d7cbcfe3a3cc0f1d17828c91cc818d5059ec69
   Test commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/cb5836cfd5632a9af1f973b36cc91b35a1d275cc

9. *Use case*: As a user, I can view a map of all the kernel crashes given all my routers so that I can see where there are issues and trends.
   Acceptance test: Given user info, when the user logs in and goes to their dashboard, then there will be a map option of viewing kernel crashes with possible filters.
   Github Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/ca54a4b7ef397fe1620de9c615b2848aee33bdd7
   Frontend test commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/fa159bdb0eec3e1b2cdc44cdbda49f65b3db39b6

10. *Use case*: As a user, I can view a table of all my routers so that I can filter and see details about any specific one.
    Acceptance test: Given user info, when the user logs in and goes to their router table page, then they will see all routers listed by page with filters and if they click on a specific router, they can see more information about it.
    Github commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/8dc19fe341267e19838acd29ca6c32a2c29ff56b
    https://github.com/evanmurray2/RGenius-Analytics/commit/41b5d239ae2726f7a1be11cdc8786ecb785c60e7

https://github.com/evanmurray2/RGenius-Analytics/commit/379501adf5b92410377588ad13682980d8388091
https://github.com/evanmurray2/RGenius-Analytics/commit/4f4edacfffb545e3794bc7b03b3b339a117946f0
Frontend test commit:
https://github.com/evanmurray2/RGenius-Analytics/commit/177bea5e0fe55553fea3fe86cab8dc4110675b2a

11. *Use case*: As a user, I can view a chart of the models vs. kernel crashes so that I can assess if there is a correlation between the two.
    Acceptance test: Given user info, when the user logs in and goes to their charts page, then they will see all their charts displayed and see the model vs kernel crashes chart and any correlation between these fields.

12. *Use case*: As a user, I can view charts of the models and releases in usage so that I can see what needs to be deprecated and assess compatibility.
    Acceptance test: Given user info, when the user logs in and goes to their charts page, then they will see all their charts displayed and see the model and releases usage for their routers.
    Github Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/b98649840e4303290fd2bba2c4e0d193d1ce1150
    Test Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/299cf8c3a86c7372d735dada47813adeb6550782

13. *Use case*: As a SmartRG employee, I can filter which data I am looking at by customer or other fields so that I can focus on certain data I need to look at.
    Acceptance test: Given the employee account, when the employee chooses a filter with the given field options, like customer and location and others, then their view will update with the given preferences.
    Github Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/bb818cd00c08008d7b51fd4cac7a53ab87e93c64

14. *Use case*: As a user, I can filter my maps by zipcode so that the map zooms in and only displays those routers.

Acceptance test: Given a zipcode in the map filter, when the user clicks enter, then the map will zoom in to that area and display data there, and when the user clicks reset world view, then the map will reset to the original zoom.

15. *Use case*: As a user, I can view router info from the map so that I can view the info by location.
    Acceptance test: Given a user click, when the user left clicks on an individual point on the map, then a new tab will open with that router's info and when the user right clicks on a cluster, then that will link to a table will all the routers in that cluster.

16. *Use case*: As a SmartRG employee, I can verify a customer account once it has been created so that only valid customers can access their data.
    Acceptance Test: Given valid company details for an account, when a customer pays for the new service an account can be created with specified permissions group (test: verify that an account is generated in backend database when supplied with credentials)
    Github Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/565b95015138c84fd3a251358f2b155b4d912911
    Test Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/e710c73292380fac7bea08847c090394beedc3fc

17. *Use case*: As a user, I can pin charts so that I can view them on my dashboard.
    Acceptance test: Given a user account with router data, when the user views all their charts and pins up to 3, then those pinned charts will show up on the home page dashboard.

18. *Use case*: As a user, I can choose which data to be visualized with filters and such so that I can create custom charts.
    Acceptance test: Given the user selections, when the data is processed by the frontend, then the frontend will make the appropriate API call so that the user will view exactly the data they wanted to view.

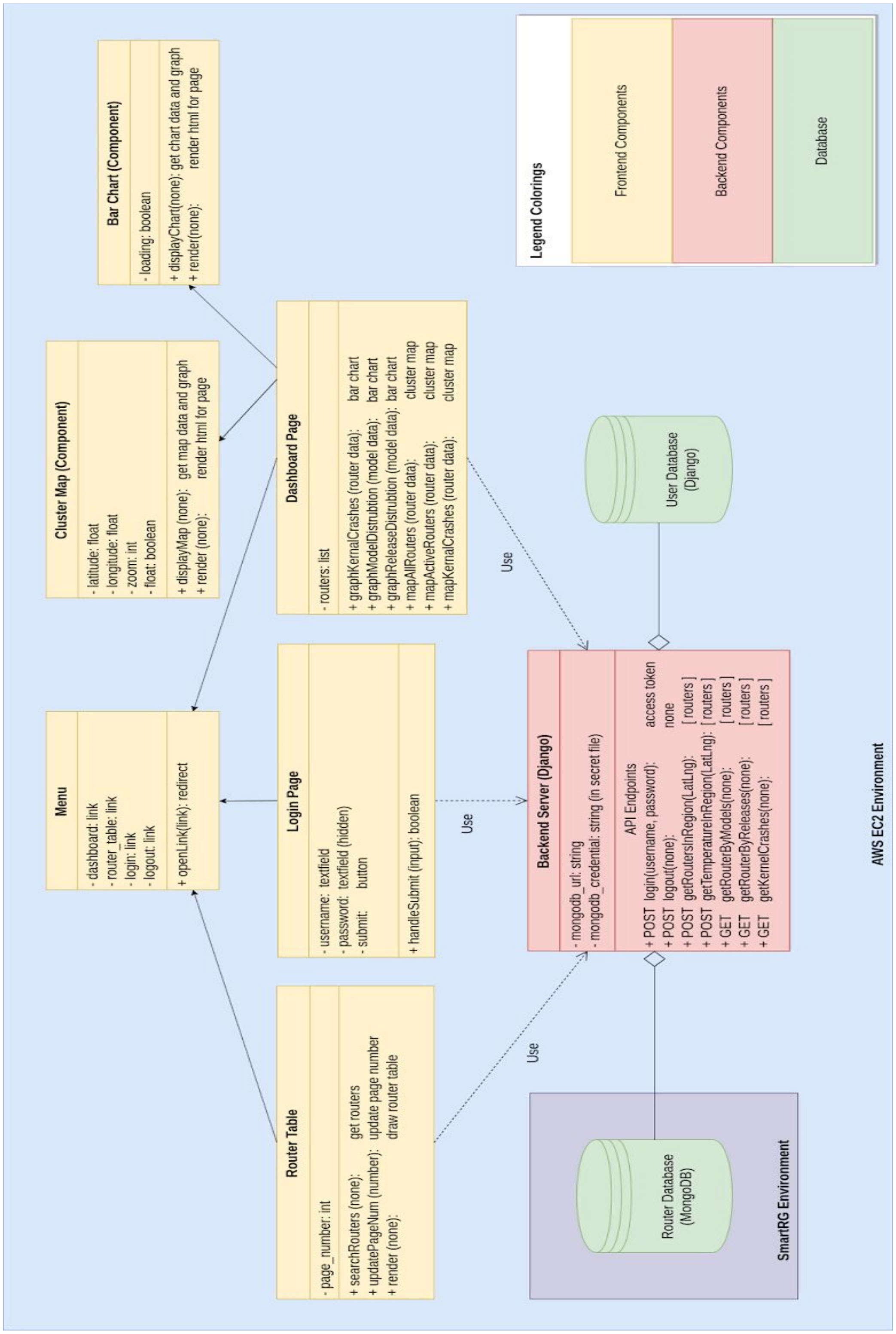
19. *Use case*: As a User Info Database, I can store pinned charts so that the pinned charts are saved to that login.
    Acceptance test: Given the user account, when the user pins charts on the more charts page, then this information will be stored in the database so that it can be saved and the user can view it whenever they log in.

20. *Use case*: As a User info Database, I can update user preferences so that users can save their custom chart preferences.
    Acceptance test: Given the user account, when the user selects custom preferences for a chart, then this information will be stored in the database so that it can be saved and the user can view it whenever they log in.

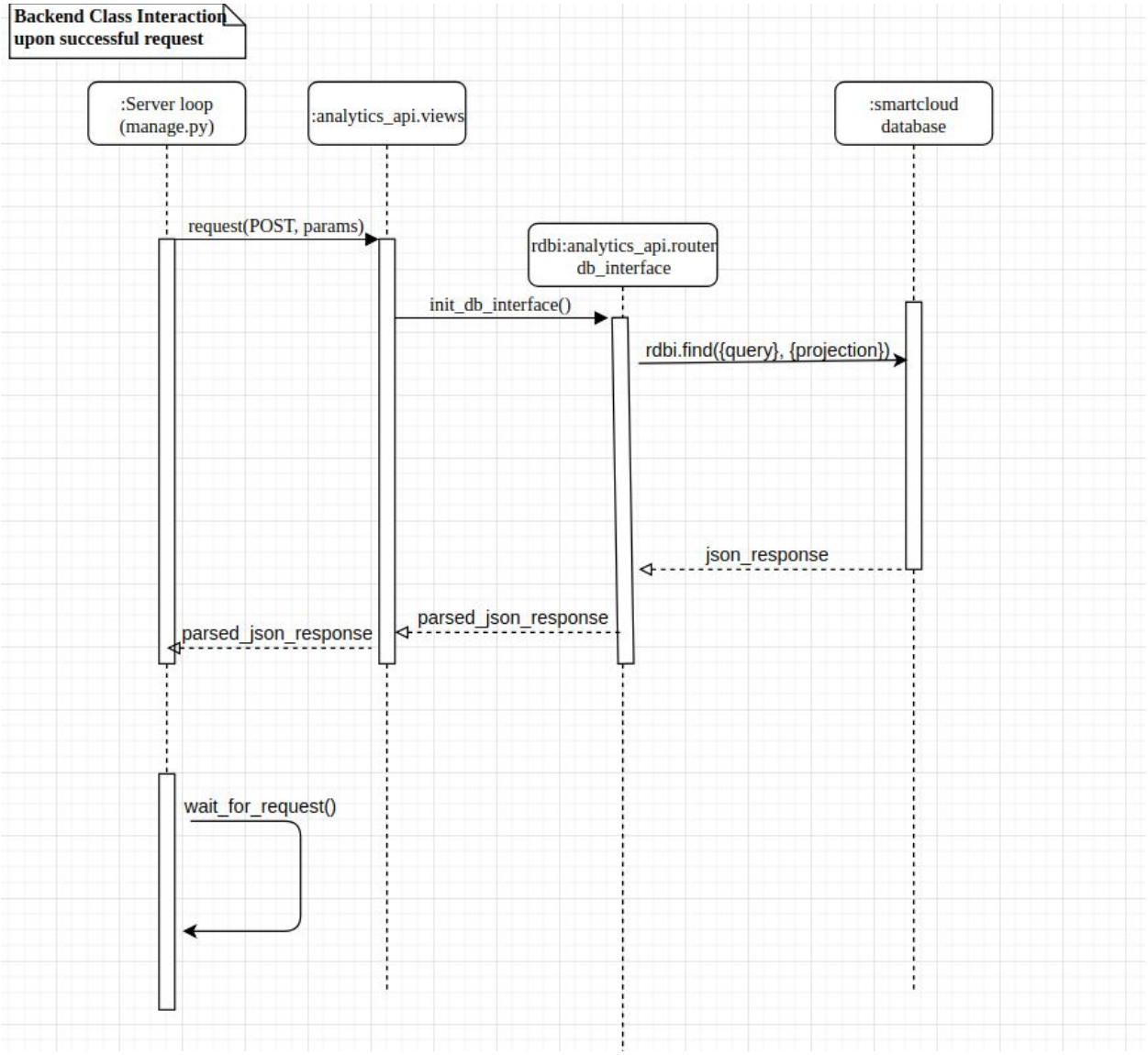
21. *Use case*: As an API, I can handle multiple requests so that server resources are maximally utilized.
    Acceptance test: Given an API, when multiple requests are received, then multiple threads carry out the requests.

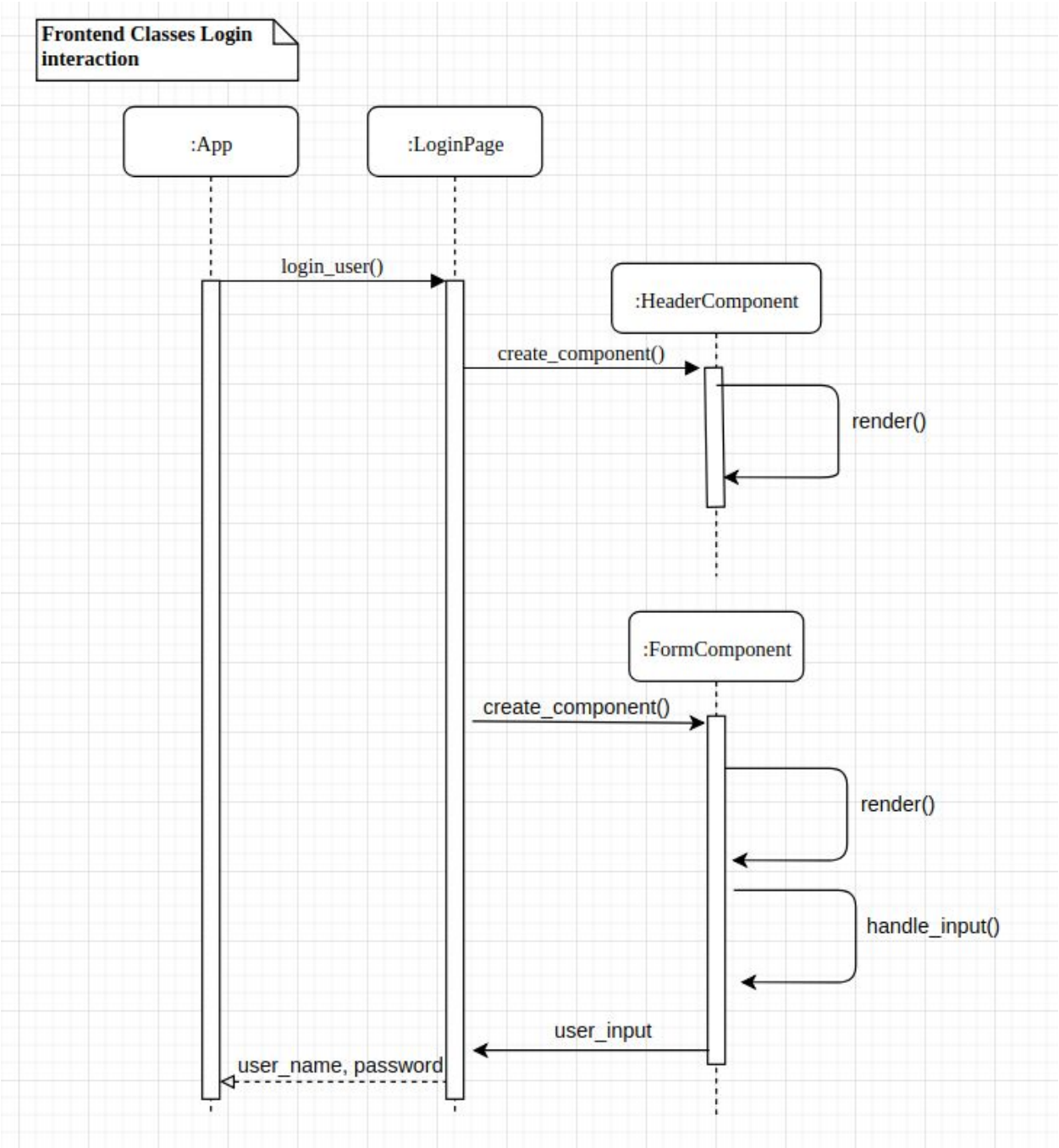
System Models

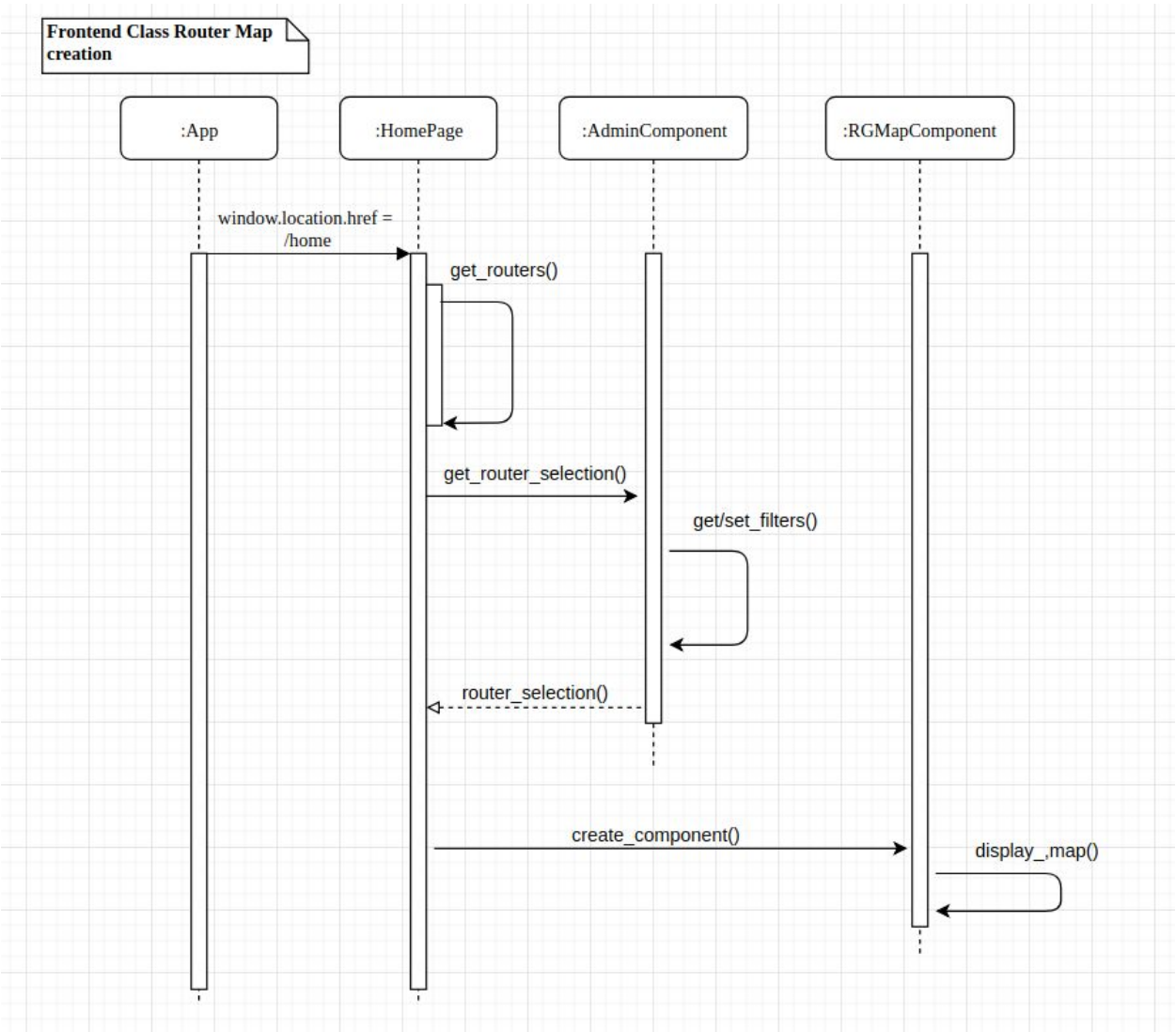
UML Class Diagram



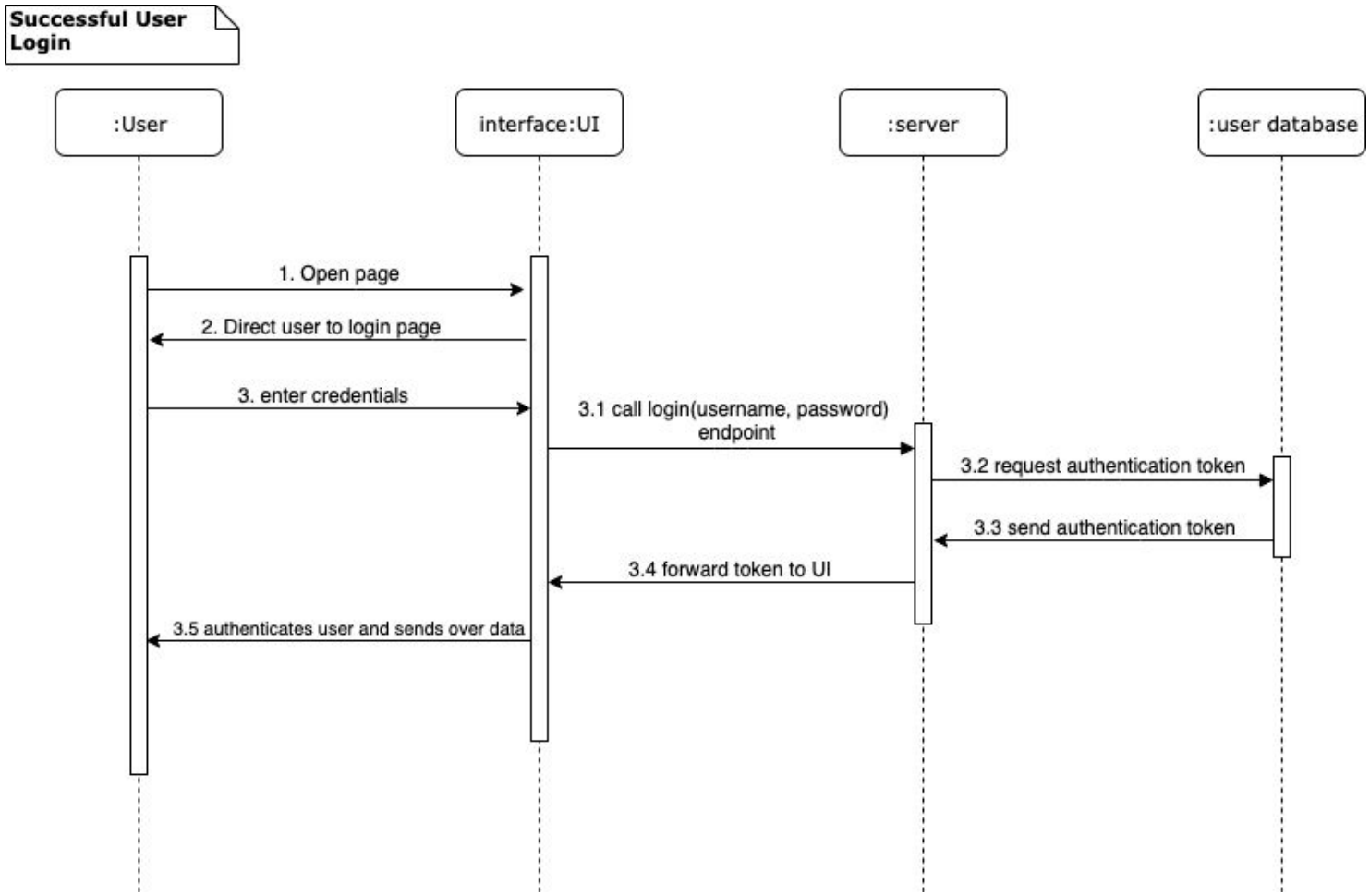
Class Interaction Sequence Diagrams

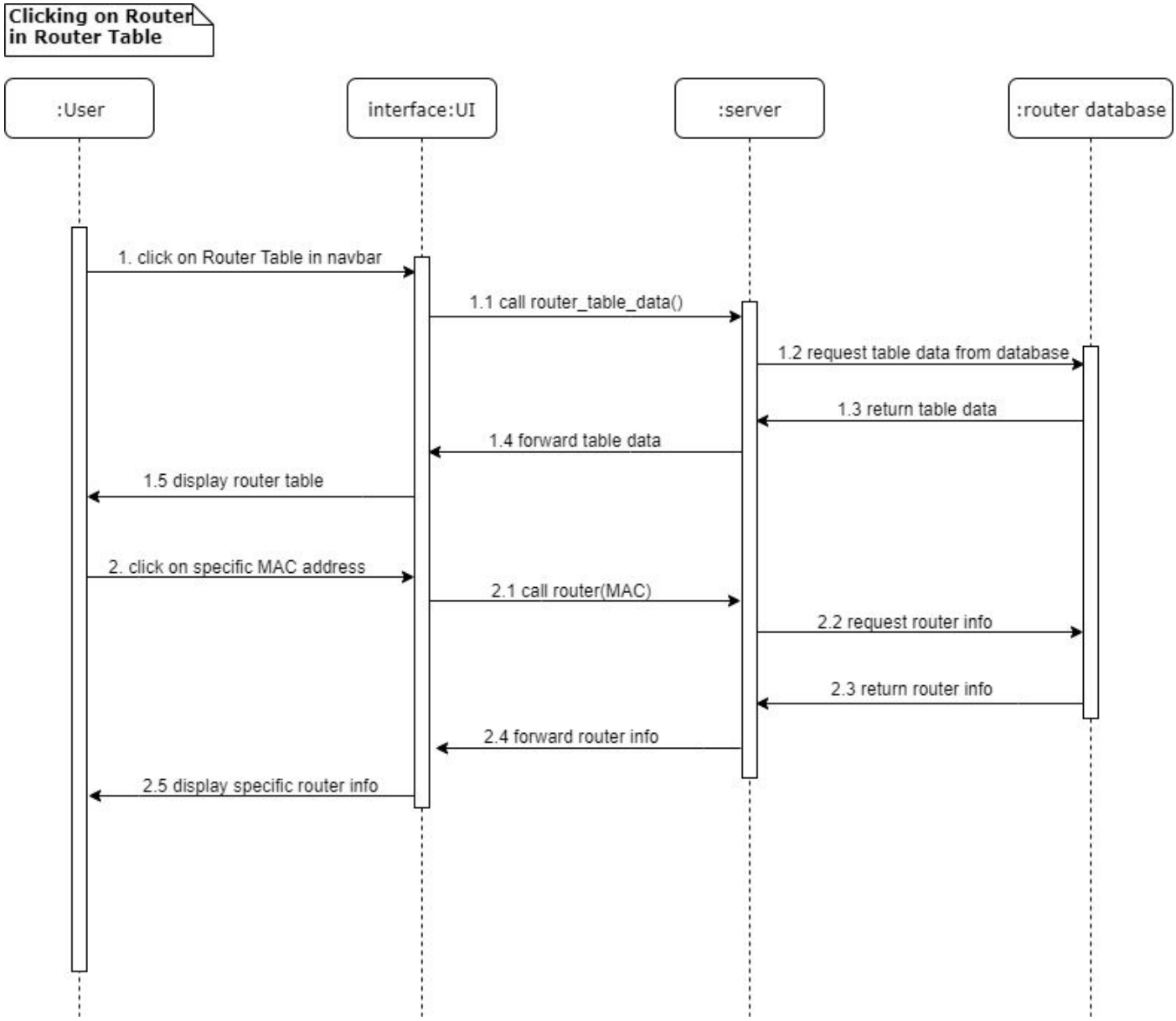

Backend Class Interaction
upon successful request
:Server loop
(manage.py)
:analytics_api.views
rdbi:analytics_api.router
db_interface
:smartcloud
database
request(POST, params)
init_db_interface()
rdbi.find({query}, {projection})
json_response
parsed_json_response
parsed_json_response
wait_for_request()

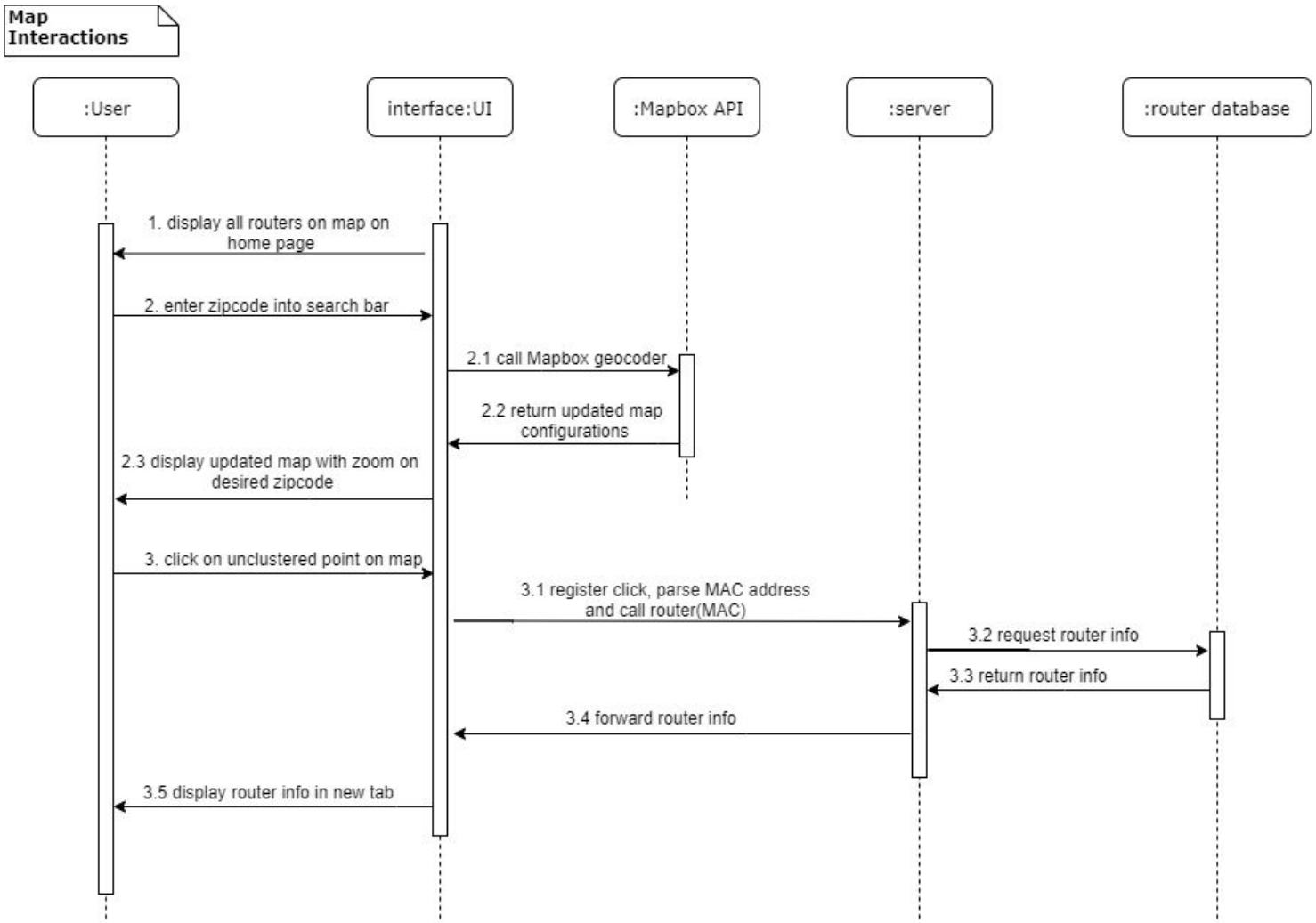

Frontend Classes Login interaction
:App
:LoginPage
login_user()
:HeaderComponent
create_component()
render()
:FormComponent
create_component()
render()
handle_input()
user_input
user_name, password


Frontend Class Router Map creation
:App
:HomePage
:AdminComponent
:RGMapComponent
window.location.href = /home
get_routers()
get_router_selection()
get/set_filters()
router_selection()
create_component()
display_.map()

User Interactions Sequence Diagrams




Clicking on Router in Router Table
:User
interface:UI
:server
:router database
1. click on Router Table in navbar
1.1 call router_table_data()
1.2 request table data from database
1.3 return table data
1.4 forward table data
1.5 display router table
2. click on specific MAC address
2.1 call router(MAC)
2.2 request router info
2.3 return router info
2.4 forward router info
2.5 display specific router info


Map Interactions
:User
interface:UI
:Mapbox API
:server
:router database
1. display all routers on map on home page
2. enter zipcode into search bar
2.1 call Mapbox geocoder
2.2 return updated map configurations
2.3 display updated map with zoom on desired zipcode
3. click on unclustered point on map
3.1 register click, parse MAC address and call router(MAC)
3.2 request router info
3.3 return router info
3.4 forward router info
3.5 display router info in new tab

Appendix

Technologies Used:

- Cloud technologies- AWS, Docker
- Frontend frameworks- React
- Backend frameworks- Django, Python
- Database libraries- MongoDB
- Libraries for displaying data- Chart.js, Mapbox, React-Bootstrap