Product Design

Team *6, Traffic Analyzer and Content Optimizer*
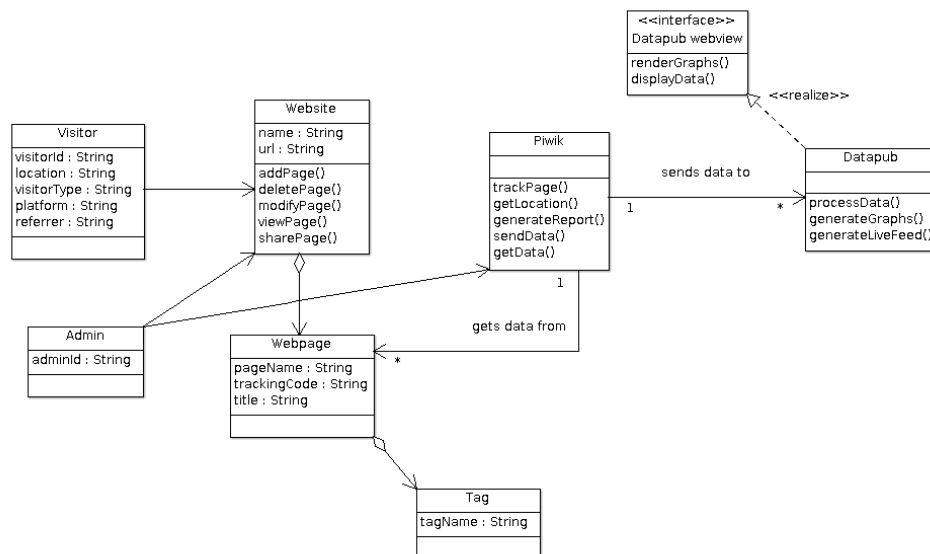
Members

Aayush Naik

Rohan Karnawat

Vinayak Athavale

Mohammed Sharfuddin

Design Model

|	
Visitor	**Class state**: Information about the visitor who visits the website. This information includes the visitorId, the loaction of the visitor, visitorType(new or returning), platform on which the visitor works on(eg: Windows 7 or Android) and the referrer of the user(i.e. the place from where the user landed on the website, for eg. a search engine.)
Admin	**Class state**: The adminId of the admin.
Website	**Class state**: The name and url of the website. A website is made up of web pages. **Class behavior**: Adds, deletes and modifies web pages. Leads to the viewing and sharing of web pages by visitors.
Piwik	**Class behavior**: Tracks web pages, gets location of the visitor, generates different reports on various metrics, sends data to Datapub and gets data from web pages.
Datapub	**Class behavior**: Processes data sent by Piwik. Using this data, organizes data for rendering of graphs and other metrics.

.

Sequence Diagram(s)

Generation/Retrieval of Cookie:

Cookie Generation
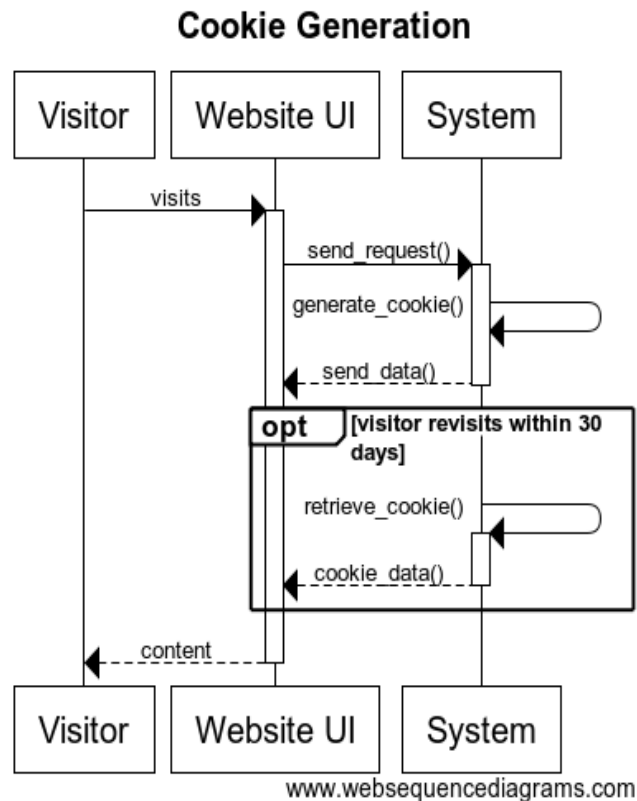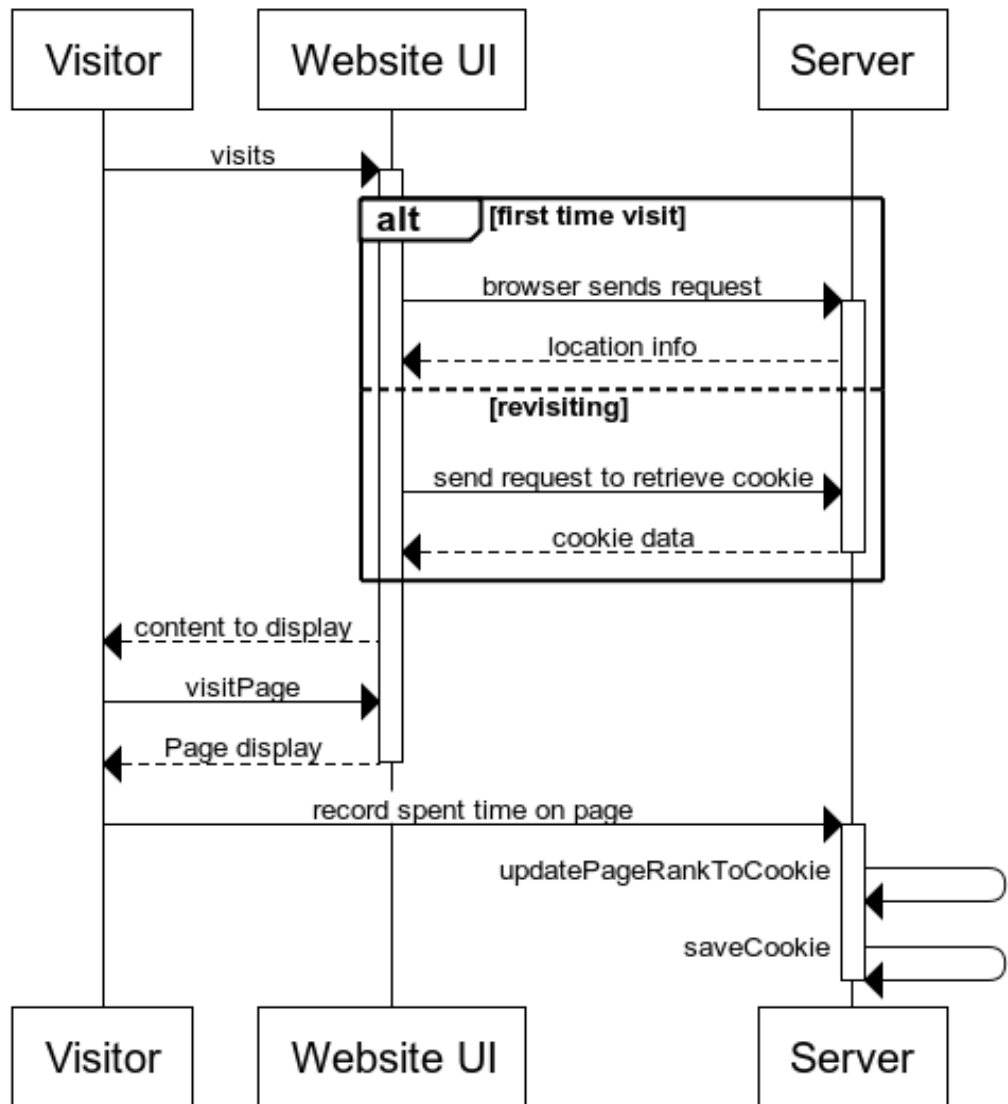
Processing IP address of Visitor

Process IP

www.websequencediagrams.com
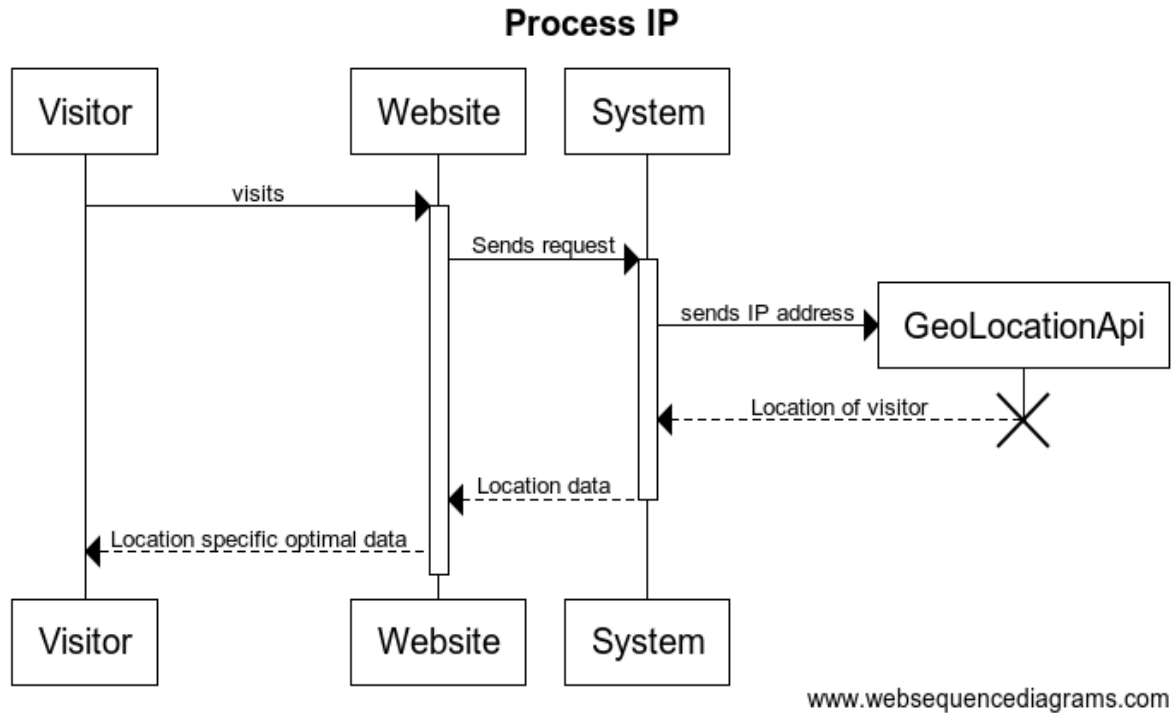
Logging activity of various users and generating graphs to analyse traffic

Render Graphs

www.websequencediagrams.com
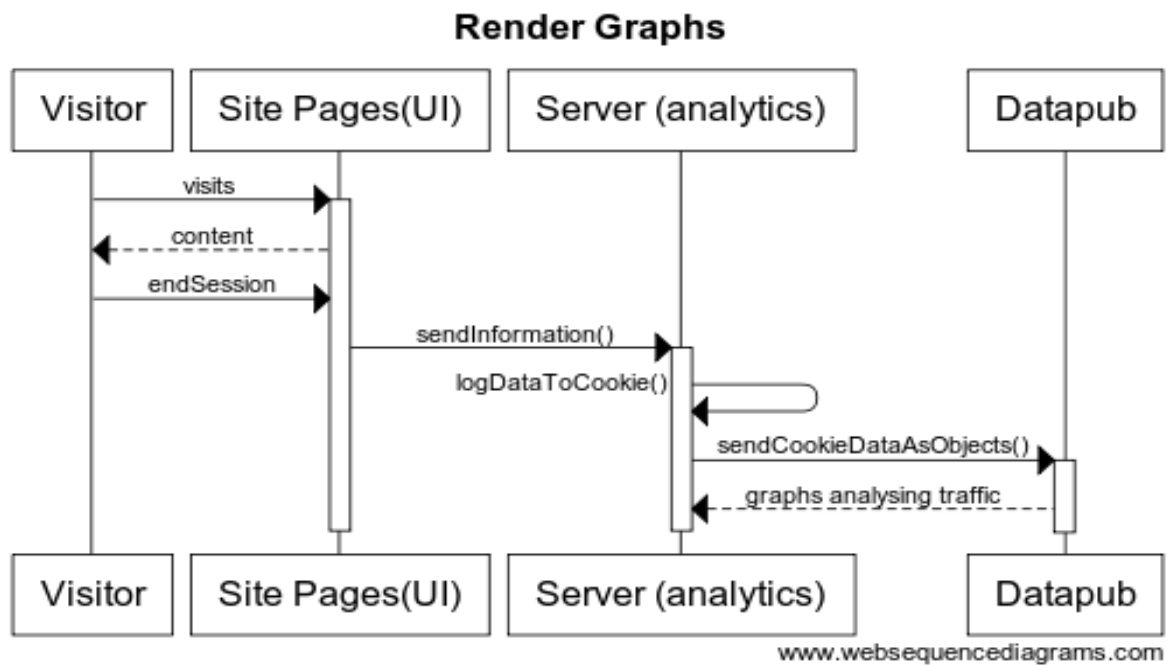
on website

Getting optimized content based on visitors previous visits

Generate personalized data

Design Rationale

We had to design an algorithm for page ranking which takes into account no of views , date of publishing and incorporate the user preferences into it. First we decided to go with a rank algorithm with decay but after running a few simulations we realized that that was not suitable for us as an unfair advantage was given to any one component(no of views,preferences etc.)

So we decided to make an algorithm without decay which was easier to implement and was fair but computationally intensive . Finally we decided that as the scale of our project was not that high an algorithm without decay would work better so we sacrificed high performance for ease in implementation and correctness.