Software Architecture Document

SwedishGeoTweets v1.0
Marmalade

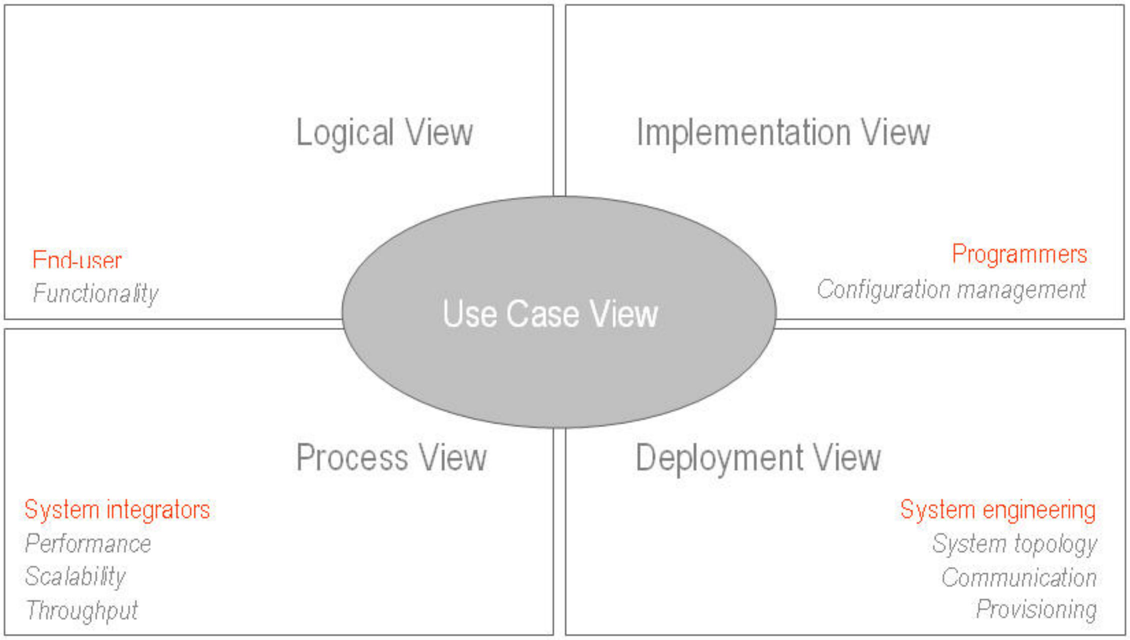
Table of content

1 Introduction

1.1 Purpose

This is a SAD for the project SGT, the purpose of this document is to explain the architecture of the project. The document will be structured so only people with certain experience will be able to read and understand it.

In order to depict the software as accurately as possible, the structure of this document is based on the “4+1” model view of architecture [KRU41].



.

1.2 Scope

The scope of this SAD is to depict the architecture of Twitter trending application by the project group Marmelade.

1.3 Acronyms

SAD	Software Architecture Document
SGT	SwedishGeoTweets
UML	Unified modeling language
TimestampYMD	Timestamp Year Month Day
IP	Internet Protocol
PHP	Hypertext Processor

1.4 References

http://www.cs.toronto.edu/~wl/teach/407/2002/rup-sad.html
http://en.wikipedia.org/wiki/Burn_down_chart
http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf
SAD-SOLA Document
SAD-SRS Document
SAD-OnlineCateringservice Document
Bass, Clements & Kazman, *Software Architecture in Practice*, 3rd edition, Addison-Wesley Professional Computing Series, 2012

1.5 Overview

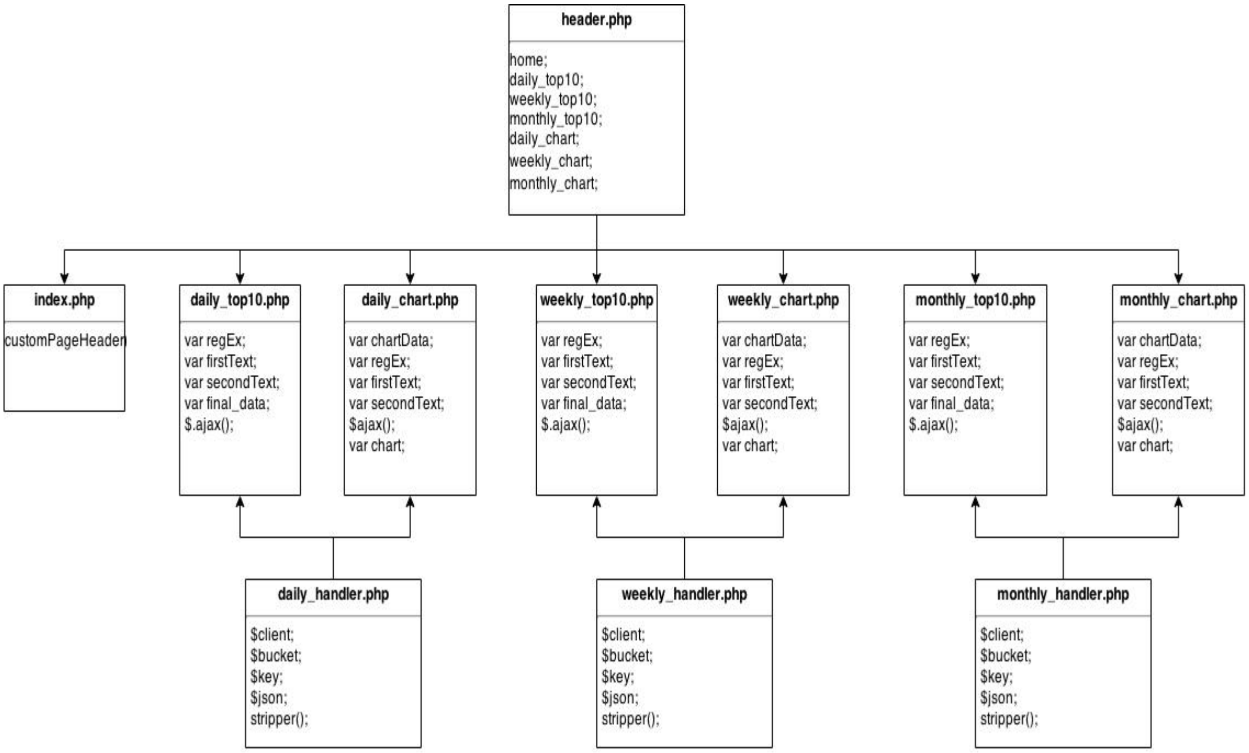
This SAD is written to give the reader an architectural description of the program SGT. Using different views to achieve this we base the structure of this document on the 4+1 model.
The point of using several viewpoints of the software architect is to look at different aspects of the system and by this see how we have chosen to work with the different modules of the architect for different aspects of the program. This will allow different stakeholders to find what they search for in the software architecture.

2. Architectural Representation

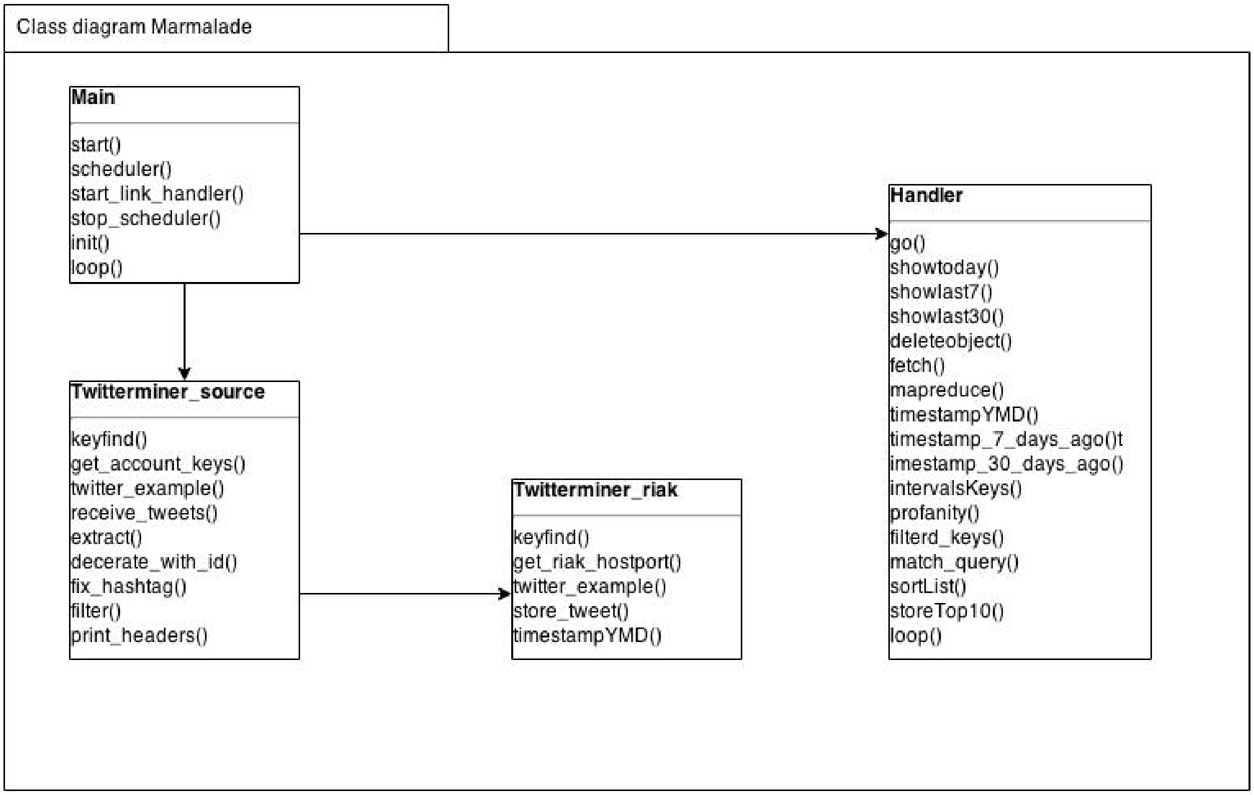
Marmelade are using the views defined in the 4+1 model.

Logical view:

The logical view displays with the functional requirement of SGT. It will also display the most important use case realizations. In this view UML diagrams is used. This includes such diagrams as *Class, Communication, Sequence*. The audience for this view is mainly focused on the Designers.



Frontend Class Diagram



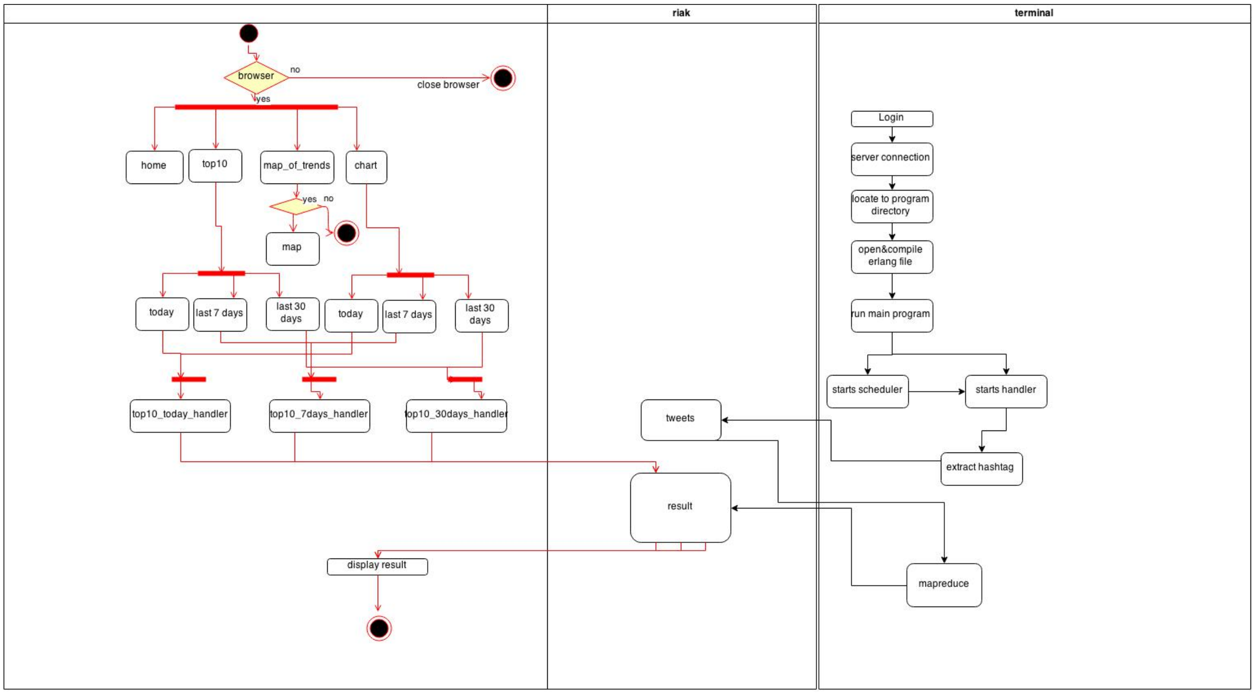
Backend Class Diagram

Development/Implementation view:

Development view is a view from a programmers perspective about software management for a programmers. This view will describe the layers, subsystems of the program. In this view *Component diagram* and *Package diagram* is used. Component diagram will illustrate the different systems components. Package diagram is a Unified Modeling Language (UML) used to represent the deployment view.

Process view:

The process view is for a client/system integrators. The process view focus on explains the systems processes, how they communicate and runtime behavior, such as performance and stability. An *Activity diagram* will be used for this view



Physical/deployment view:

This view focus on the connection and the topology of the softwares components on the physical layer. The Deployment diagram is used to represent this view. The physical view is for system engineer's/Deployment managers Point of view.

Use case view/Scenarios:

This is the "fifth" view. This view represent a scenario or a sequence of interactions between objects and processes. A use case is used to validate an architecture design. It is also be used as a starting point for an prototype of an architecture.

3 Architectural goals and constraints

3.1 Quality attribute requirements:

For this project we had certain quality attributes in mind. High availability, High performance, as well as reliability. We program should be up and running as much as possible.

3.2.1 Security

The application will not require users to log in or own an account, therefore the security will not be like other entertainment applications. The security we will have is data integrity. So data sent through the network can only be handled by the administrators and others who have access to the server and the credentials.

3.1.2 Reliability/Availability

As long as our server and domain is up and running the project will be available and up to date. To prevent dead weight we have the Erlang applications server that will support load balancing through our 5 nodes.

3.1.3 Performance

Performance is an indication of the responsiveness of a system to execute any action within a given time interval.
The process from when the users selects its opinion from the menu bar, till the results display should not take more than 0.5-1 seconds.

3.1.4 Scalability

System's reaction when the user demands increases. Erlang server supports increases in workload.

3.2 Functional Requirements:

- Extract data: Stream data from twitter.
- Filter data: The data streamed from twitter needs to be filtered so that we can get the exact data we want. Filtering out the hashtags in our case.
- Store data: The data then needs to be stored in a database. Riak. Where we store it with Key/Value to be able to query it at a later date.
- Mapreduce: Our program needs to run a mapreduce on the Key/Value stored information in order to find out whichever hashtag has been tweeted the most amount of times.
- Store top 10: The top 10 data is sorted after being mapreduced and then finally stored back in a result bucket.
- Query data: The data must then be queriable from the website.
- Display list: The website must then be able to display the top 10 lists.

3.3 Technical constraints:

- Linux/iOS: Riak gave us the restrain of having to use linux or iOS.
- Riak, all the data have to be stored in Riak.
- Erlang: some parts of the program had to be written in erlang.
- A requirement of having at least 50% of our function in Mapreduce have limit the project for quick fixes.

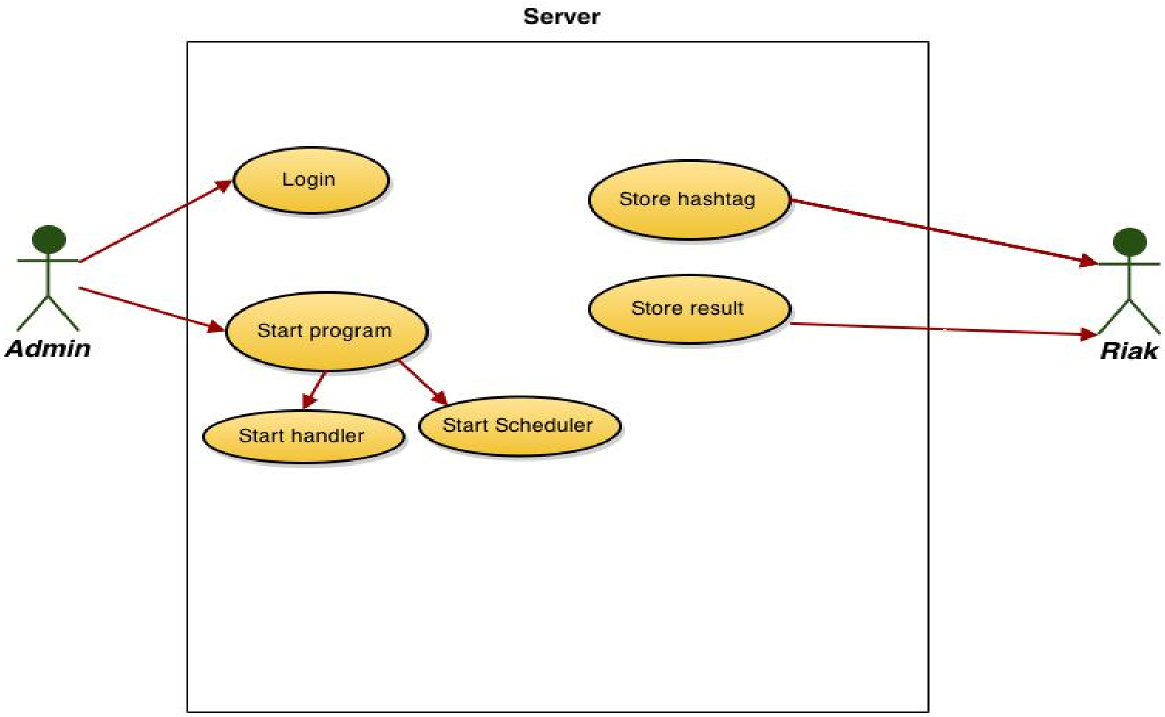
4 Use-case view

4.1 Use Case View

In this chapter we will cover those scenarios from use case model of central functionality of the system. We will address the architectural coverage of the use case. e.g if they cover a specific point in the architecture.

4.2 Architecturally significant use cases.

This use case is of significance. It displays an admin starting the program.
Use cases:



4.3 Use Cases:

4.3.1 Login:

1. Login
1.1 To login to the server, one must first connect to the server to able to login.

2.
2.1 Workflow:
To login to the server you can use the terminal and enter a request to the server with (ssh maramalade@129.16.155.22) This will give you the option of writing a password to be able to login. If your credentials are in order you will be logged on to the server and given access to all the stored files.

2.2 Alternative flow:
When trying to logon to the server via filezilla you need to enter the server information either in a quickconnect or adding an option to connect to a certain configured connection. If all credentials including port, username and password is correct you will be given access to all the stored files.

3. Preconditions

1. Server is up
2. You have the correct credentials

4. Postconditions

1. You will be logged in
2. You have access to all the stored files

5. Scenario:
If you need access to files stored on the server, you may wish to login.

5.1 Workflow:

1. Open terminal
2. Enter credentials
3. Enter password
4. Login

5.2 Alternative flow:

1. Open filezilla
2. Create login
3. Enter credentials
4. Enter password

5. Login

4.3.2 Start program

1. Start program:
1.1. To startup the server program and have it all run the admin needs to login and start it once. Once started the program will start other parts.

2.
2.1 Workflow:
When you need to start the program, you start up a terminal and login to the server at 129.15.166.22 and enter the password to login. Once logged in you locate the programs erlang files in whichever directory it has been put in. Once located, you compile that erlang file in the terminal and you open that program up with erl -pa deps/*/ebin -pa ebin -config twitterminer to enable certain configurations and then start the program by writing main:start() in the terminal.

2.2 Alternative flow:
Start the program filezilla. In filezilla enter the server and port information to log on to the server at 129.15.166.22 and enter the password to login. Once logged in you locate the programs erlang files in whichever directory it has been put in. Once located, you compile that erlang file in the terminal and you open that program up with erl -pa deps/*/ebin -pa ebin -config twitterminer to enable certain configurations and then start the program by writing main:start() in the terminal.

3. Preconditions:
1. Server is up
2. Erlang installed on server
3. Riak installed on server
4. Files located on server

4. Postconditions:
1. Program is running
2. Program is waiting for schedule to initiate other parts

5. Scenario:
When you wish to start the server program to update website the program needs to be started.

5.1 Workflow:
1. Use the terminal to connect to the server
2. Enter credentials
3. Locate the program files directory
4. Compile erlang files
5. Open main erlang file

6. Start program by running main:start()

5.2 Alternative flow:

1. Use filezilla to connect to the server
2. Enter server information into filezilla
3. Enter credentials
4. Locate the program files directory
5. Open main erlang file
6. Compile erlang files
7. Open main erlang file
8. Start program by running main:start()

4.3.3 Start Handler

1. Start scheduler

1.1 Once the program is started, it will start up the handler to wait for the extractor to run its course which will then let the handler perform multiple calculations before saving results to riak.

2.

2.1 Workflow:

Once someone has connect to the server. That person can start the erlang program by running main:start(). The start function in the main module will then start the handler.

3. Preconditions

1. Server is up
2. Erlang is installed on server
3. Logged in

4. Postconditions

1. Handler is running
2. Handler is waiting for extractor to complete

5. Scenario: When you wish to start the erlang program and update the results in riak you will want to start the handler by running main:start().

5.1 Workflow:

1. Login to server
2. Compile erlang files
3. Open terminal
4. Run main:start()

4.3.4 Start Scheduler

1. Start scheduler
1.1 Once the program is started, it will start up the scheduler to wait for assigned times and perform actions.

2.
2.1 Workflow:
Once someone has connect to the server. That person can start the erlang program by running main:start(). The start function in the main module will then start the scheduler.

3. Preconditions

1. Server is up
2. Erlang is installed on server
3. Logged in

4. Postconditions

1. Scheduler is running
2. Scheduler is waiting for assigned time

5. Scenario: When you wish to start the erlang program and have it wait for the assigned time before it starts extracting, you will want to start the scheduler.

5.1 Workflow:

1. Login to server
2. Compile erlang files
3. Open terminal
4. Run main:start()

4.3.5 Store Hashtags

1. Store hashtags
1.1 To have anything to retrieve from the website we first need to store things to riak to mapreduced. So we need to extract the information we want out of the tweets we stream and store that on riak.

2.
2.1 Workflow: We login to the server, then open a terminal, compile the erlang files and run main:start(). When the scheduler then says its time for the extractor begin, the extract will store hashtags to riak after it has retrieved a stream from twitter and extracted the needed information. To store anything to the riak server we need to extract the information we want from the tweets we retrieve in a stream from twitter. Once we have done that we will create an object. That object will then have secondary indexes and other values added to them. Then it will be saved to riak.

2.2 Alternative flow: We login to the server, then open a terminal, compile the erlang files and run twitterminer_riak:twitter_example().

3. Preconditions

1. Server is up
2. Main module started
3. Scheduler started

4. Postconditions:

1. Hashtags stored to riak

5. Scenario: When you want to store additional hashtags to riak you may want to run the extractor.

5.1 Workflow:

1. Login to server
2. Open terminal
3. Compile erlang files
4. Run main:start()

5.2 Alternative flow:

1. Login to server
2. Open terminal
3. Compile erlang files
4. Run twitterminer_riak:twitter_example()

4.3.6 Store Result

1. Store result
1.1 To be able to retrieve information from riak via the website, we have to store something to the database first. And to do so, we use the function storeTop10() in the handler module.

2.
2.1 Workflow: In order to store the results to riak. We first need to mapreduce and sort the already stored information in riak. When the mapreduce and sort functions are completed, we can store it back to riak in a new bucket for results only that can then be queried from the website. This is done by sending the sorted result to the storeTop10 function which in turn will store the information to riak after creating an object with secondary indexes.
So first you need to login to the server, open a terminal, compile the erlang files and then run main:start().

2.2 Alternative flow: You can also run the handler separately by first logging on to the server, opening a terminal, compiling the erlang files then running handler:go().

3. Preconditions

1. Server is up
2. Main module started
3. Handler started

4. Postconditions

1. Results updated on riak

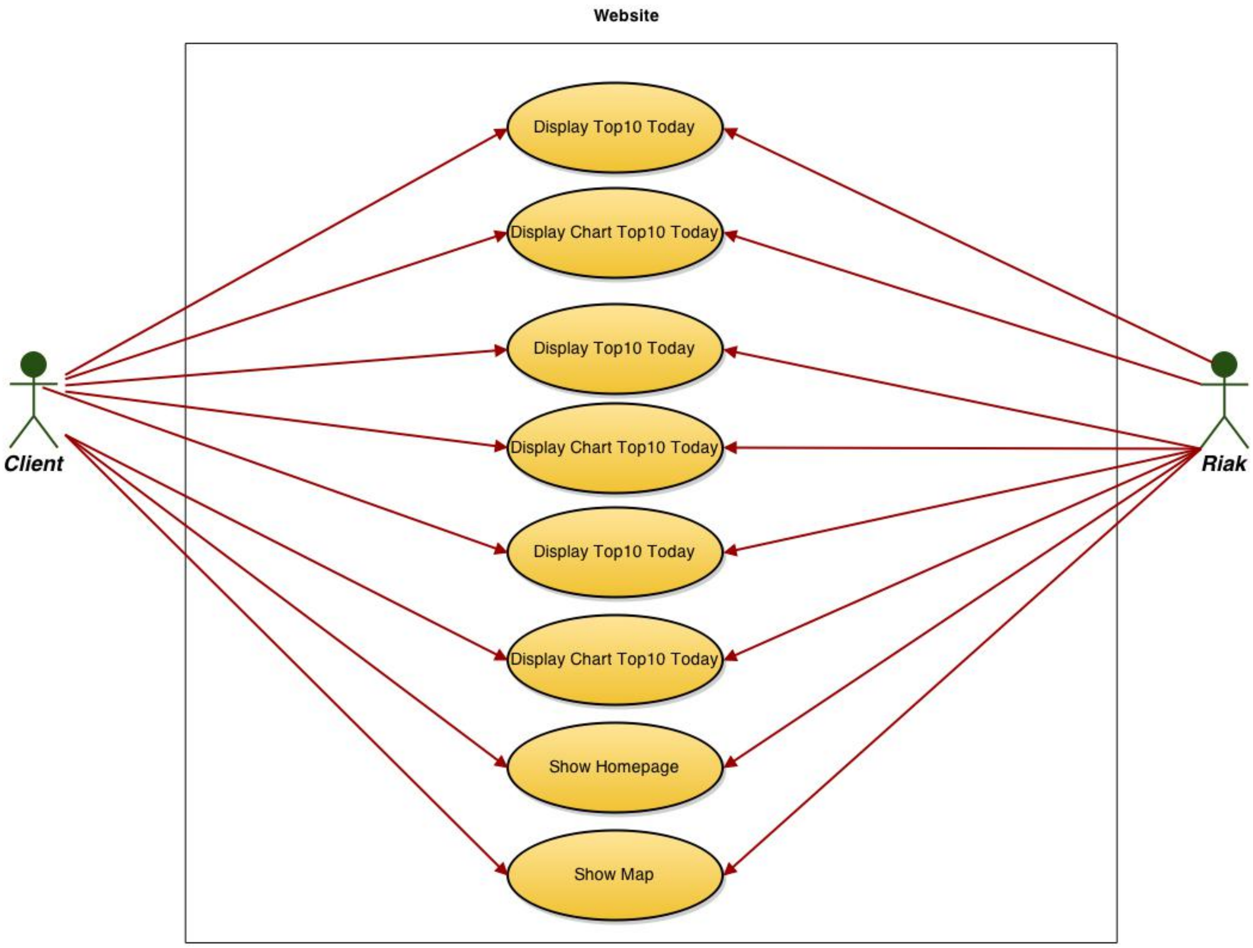
5. Scenario: When you want new information to display on the website you may want to store new results to riak.

5.1 Workflow:

1. Login to server
2. Open terminal
3. Compile erlang files
4. Run main:start()

5.2 Alternative flow:

1. Login to server
2. Open terminal
3. Compile erlang files
4. Run handler:go()



4.3.7 Show Top 10 Today

1. Show Top 10 Today
1.1 Brief Description
This is how you use the web page to show top10 list of tweets for today.

2. Flow of Events
2.1 Basic Flow
Open webbrowser, enter “www.swedishgeotweets.com”. If you don’t see a orange webpage with the text “Marmalade group” you have entered the wrong page. If you entered the correct web page you will be shown a webpage. In the webpage there will be a menubar that consists of home, top10, map of trends and chart.
Hover with your mouse over top10, Then click the word “Today”. The page will reload and top10 for today will be displayed.
2.2 Alternative Flows
If you enter in the webbrowser “http://swedishgeotweets.com/daily_top10.php” you will head directly to top10 for today.

3. Preconditions

3.1 The computer is on.

3.2 Internet is connected

3.3 A web browser is installed.

4. Postconditions

4.1. The web page will now display top 10 tweets for today.

5. Scenarios

When i want to see top10 tweets of today i go to the webpage and click top10 today. After that the user can view other information on the webpage if felt like.

5.1 Flow

1. Open web browser
2. Enter "www.swedishgeotweets.com"
   AF1: "http://swedishgeotweets.com/daily_top10.php"
3. Hover with mouse over Top10.
4. Click the word "Today".

4.3.8 Show chart of the today

1. Show Chart of Today

1.1 Brief Description

This is how you use the web page to show Chart of tweets for the Today.

2. Flow of Events

2.1 Basic Flow

Open webbrowser, enter "www.swedishgeotweets.com". If you don't see a orange webpage with the text "Marmalade group" you have entered the wrong page. If you entered the correct web page you will be shown a webpage. In the webpage there will be a menubar that consists of home, Top10, map of trends and Chart.

Hover with your mouse over Chart, Then click the letters "Today". The page will reload and Chart for the Today will be displayed.

2.2 Alternative Flows

If you enter in the webbrowser "http://swedishgeotweets.com/daily_chart.php" you will head directly to Chart for the Today.

3. Preconditions

3.1 The computer is on.

3.2 Internet is connected

3.3 A web browser is installed.

4. Postconditions

4.1. You will now be shown the Chart of tweets for Today.

5. Scenarios

When a client want to view a chart of tweets of Today. Go to the webpage and click Chart, Today. After that the user can view other information on the webpage if felt like.

5.1 Flow

1. Open web browser
2. Enter “www.swedishgeotweets.com”
    AF1: “http://swedishgeotweets.com/daily_chart.php”
3. Hover with mouse over Chart.
4. Click the word “Today”.

4.3.9 Show Top 10 Last 7 days

1. Show Top 10 Last 7 days

1.1 Brief Description

This is how you use the web page to show top10 list of tweets for 7 days back.

2. Flow of Events

2.1 Basic Flow

Open webbrowser, enter “www.swedishgeotweets.com”. If you don’t see a orange webpage with the text “Marmalade group” you have entered the wrong page. If you entered the correct web page you will be shown a webpage. In the webpage there will be a menubar that consists of home, top10, map of trends and chart.

Hover with your mouse over top10, Then click the letters “Last 7 days”. The page will reload and top10 for the week will be displayed.

2.2 Alternative Flows

If you enter in the webbrowser “http://swedishgeotweets.com/weekly_top10.php” you will head directly to top10 for the last 7 days.

3. Preconditions

3.1 The computer is on.

3.2 Internet is connected

3.3 A web browser is installed.

4. Postconditions

4.1. You will now be shown the top10 list of tweets for the last 7 days.

5. Scenarios

When a client want to see top10 tweets of the last 7 days. Go to the webpage and click top10 weekly. After that the user can view other information on the webpage if felt like.

5.1 Flow

1. Open web browser
2. Enter “www.swedishgeotweets.com”
   AF1: “http://swedishgeotweets.com/weekly_top10.php”
3. Hover with mouse over Top10.
4. Click the word “Last 7 days”.

4.3.10 Show chart of the last 7 days

1. Show Chart of last 7 days

1.1 Brief Description

This is how you use the web page to show Chart of tweets for the last 7 days.

2. Flow of Events

2.1 Basic Flow

Open webbrowser, enter “www.swedishgeotweets.com”. If you don’t see a orange webpage with the text “Marmalade group” you have entered the wrong page. If you entered the correct web page you will be shown a webpage. In the webpage there will be a menubar that consists of home, Chart, map of trends and Top10.

Hover with your mouse over Chart, Then click the letters “last 7 days”. The page will reload and Chart for the last 7 days will be displayed.

2.2 Alternative Flows

If you enter in the webbrowser “http://swedishgeotweets.com/weekly_chart.php” you will head directly to Chart for the last 7 days.

3. Preconditions

3.1 The computer is on.

3.2 Internet is connected

3.3 A web browser is installed.

4. Postconditions

4.1. You will now be shown the Chart of tweets for the last 7 days.

5. Scenarios

When a client want to view a chart of tweets of the last 7 days. Go to the webpage and click Chart, last 7 days. After that the user can view other information on the webpage if felt like.

5.1 Flow

1. Open web browser
2. Enter “www.swedishgeotweets.com”

    AF1: “http://swedishgeotweets.com/weekly_chart.php”

3. Hover with mouseover Chart.
4. Click the word “last 7 days”.

4.3.11 Show Top 10 Last 30 days

1. Show Top 10 last 30 days

1.1 Brief Description

This is how you use the web page to show top 10 list of tweets for the last 30 days.

2. Flow of Events

2.1 Basic Flow

Open webbrowser, enter “www.swedishgeotweets.com”. If you don’t see a orange webpage with the text “Marmalade group” you have entered the wrong page. If you entered the correct web page you will be shown a webpage. In the webpage there will be a menubar that consists of home, top10, map of trends and chart.

Hover with your mouse over top10, Then click the letters “Last 30 days”. The page will reload and top10 for the last 30 days will be displayed.

2.2 Alternative Flows

If you enter in the webbrowser “http://swedishgeotweets.com/monthly_top10.php” you will head directly to top10 for the last 30 days.

3. Preconditions

    3.1 The computer is on.

    3.2 Internet is connected

    3.3 A web browser is installed.

4. Postconditions

    4.1. You will now be shown the top10 list of tweets for the last 30 days.

5. Scenarios

When a client want to see top10 tweets of the last 30 days. Go to the webpage and click top10 Last 30 days. After that the user can view other information on the webpage if felt like.

5.1 Flow

1. Open web browser
2. Enter “www.swedishgeotweets.com”

    AF1: “http://swedishgeotweets.com/monthly_top10.php”

3. Hover with mouse over Top10.
4. Click the word “Last 30 days”.

4.3.12 Show chart of the last 30 days

1. Show Chart last of 30 days
1.1 Brief Description
This is how you use the web page to show Chart of tweets for the last 30 days.

2. Flow of Events
2.1 Basic Flow
Open webbrowser, enter “www.swedishgeotweets.com”. If you don’t see a orange webpage with the text “Marmalade group” you have entered the wrong page. If you entered the correct web page you will be shown a webpage. In the webpage there will be a menubar that consists of home, Chart, map of trends and Top10.
Hover with your mouse over Chart, Then click the letters “Last 30 days”. The page will reload and Chart for the last 30 days will be displayed.
2.2 Alternative Flows
If you enter in the webbrowser “http://swedishgeotweets.com/monthly_chart.php” you will head directly to Chart for the last 30 days.

3. Preconditions
3.1 The computer is on.
3.2 Internet is connected
3.3 A web browser is installed.

4. Postconditions
4.1. You will now be shown the Chart of tweets for the last 30 days.

5. Scenarios
When a client want to view a chart of tweets of the last 30 days. Go to the webpage and click Chart, Last 30 days. After that the user can view other information on the webpage if felt like.

5.1 Flow
1. Open web browser
2. Enter “www.swedishgeotweets.com”
AF1: “http://swedishgeotweets.com/monthly_chart.php”
3. Hover with mouseover Chart.
4. Click the word “Last 30 days”.

4.3.13 Show Homepage

1.Brief description

1.1 This use case describes how you go back to the start page by using Home button. When you are in the web page and navigating among the different views, you can at any time go up to the top menu and press the Home button by clicking the letter “home”. this will take you back to the start view.

2. Flow of event

1 Client enters the web page by writing the adress http://swedishgeotweets.com/
2 The client needs to enter a different view of the web page.
3 The client navigates up to the top menu and the button “Home”.
4 Clicks the button “Home” by clicking the letters home.
5 The Client returns back the the start view of the page.

3.Precondition

1. System is on
2. client have internet connection.

4.3.14 Show map

1.Go to map

1.1 This use case describes how you enter “the map of trends” on the webpage.

2. Flow of event

1 Client enters the webpage by writing the adress http://swedishgeotweets.com/.
2 Uses the top menu to navigate to the “map of trends” button.
3 Client needs to click the letters of the button to enter the world map.
4 The map Shows.

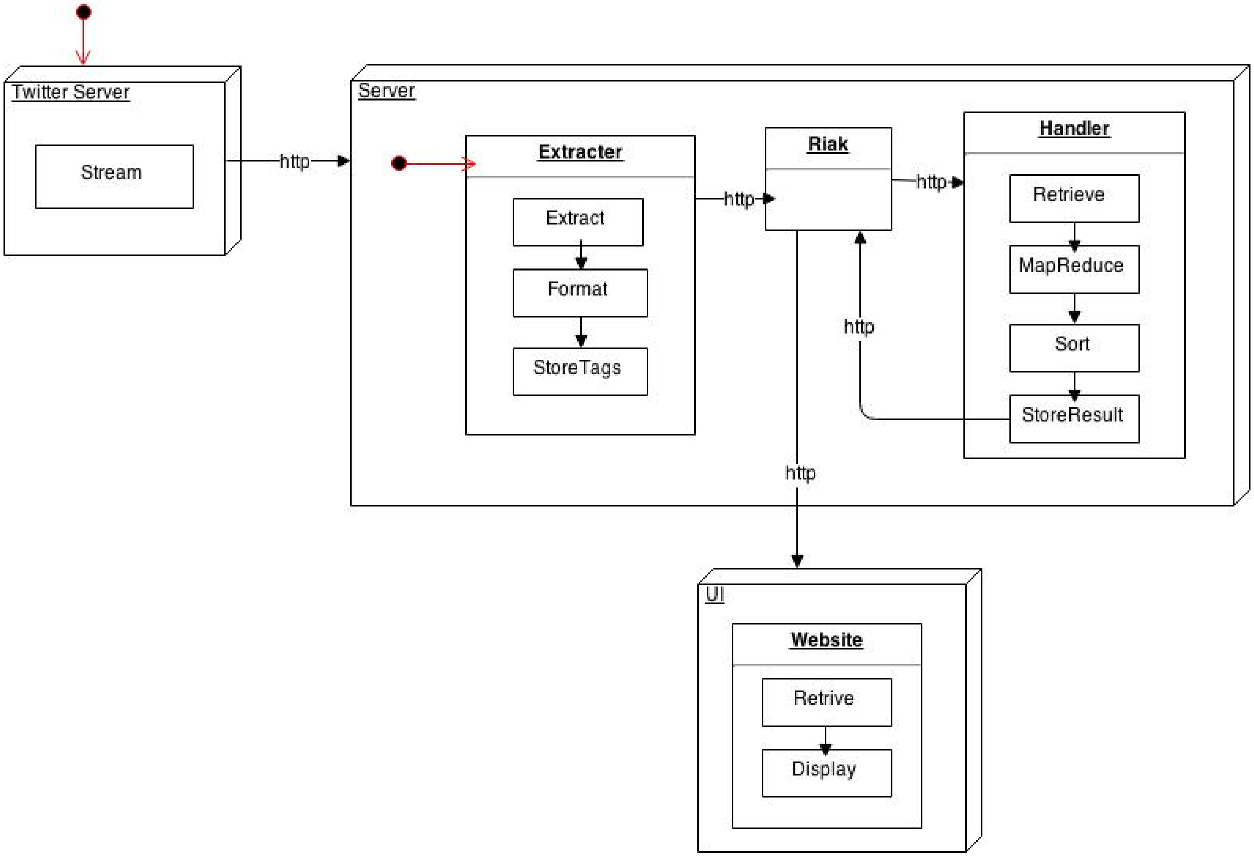
2.1 Alternative flow

1 The client could enter the direct address to get to the map straight away “http://swedishgeotweets.com/map_of_trends.php”.

3.Precondition

1. System is on
2. client have internet connection.

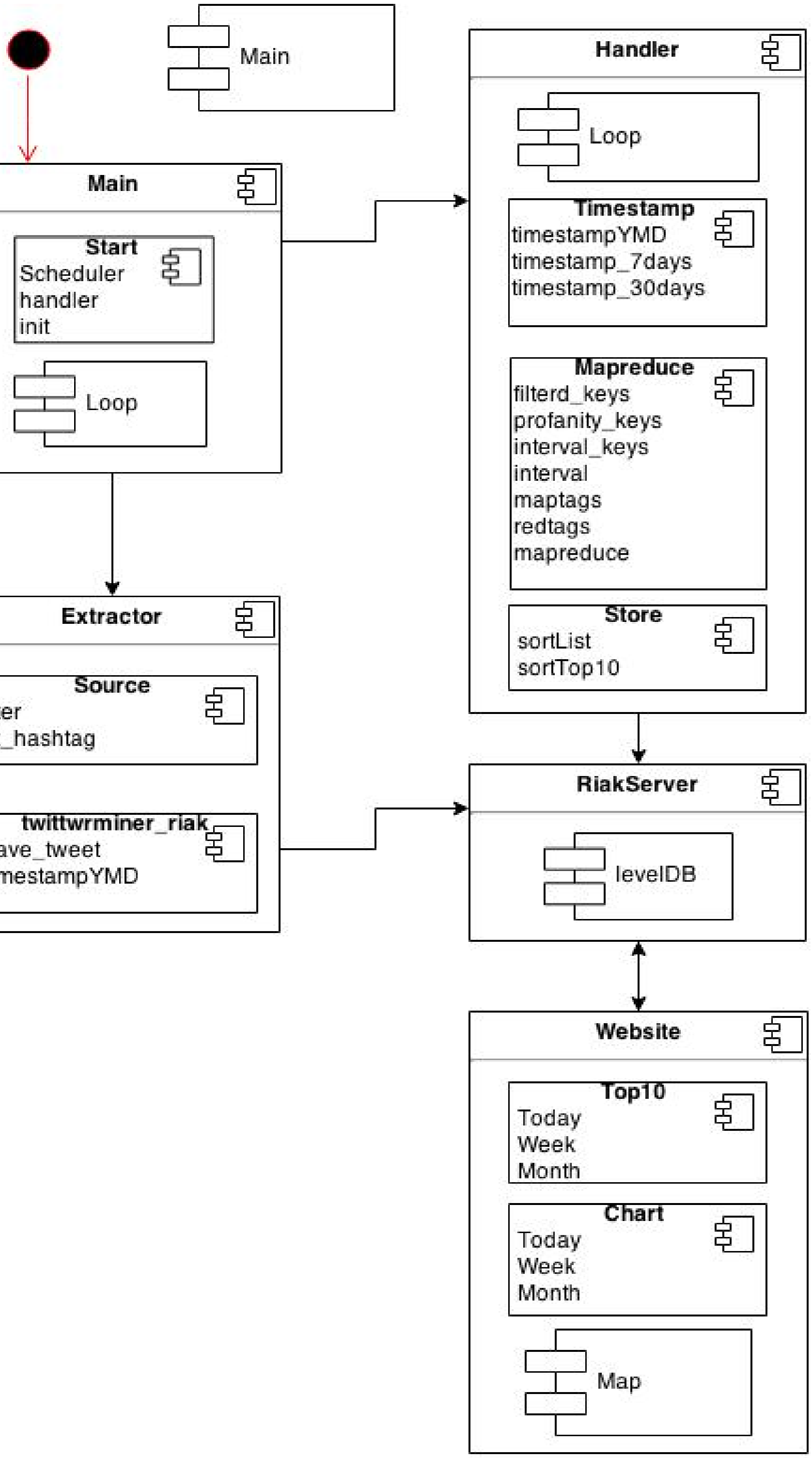
5. Deployment View:



The deployment is done as a two-tiered system where our erlang modules for the backend are stored on the same server as our riak database. Because we feel that our project is not big enough to justify buying additional server services. We are currently borrowing a server from the school and have bought a web domain.

Our frontend php files are stored on the server as well but we are using the domain to host the website. This seemed to be the most efficient way of doing it for us.
This would mean that the server is one tier and the users own computer will be the second.
Making our project two-tiered.

6. Component Diagram



Homepage(Subsystem): Contains, Top 10, Charts, Map

- Top 10: Top 10 displays list of top 10 for daily, weekly or monthly feed. Revised or original.
- Charts: Shows chart diagrams of top 10 lists.
- Map: Shows map and location where tweets originate.

Handler(Subsystem): Contains, Mapreduce, Timestamp, TimestampYMD, Timestamp_7_days, Timestamp_30_days, Delete, Fetch, Interval, Interval_Keys, Profanity_Keys, Filtered_Keys, sortList, storeTop10 and Loop.

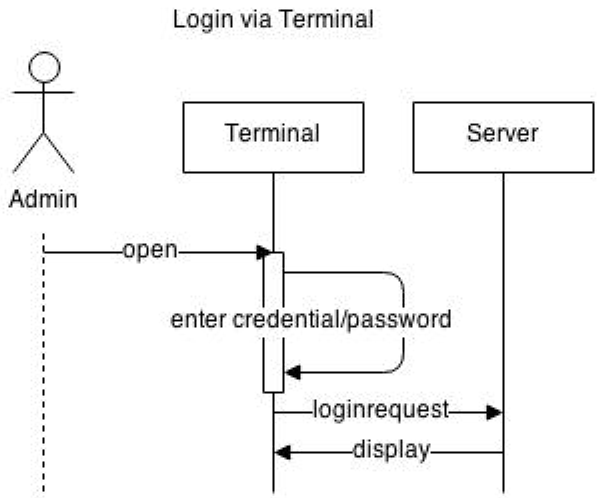
- Mapreduce: Runs mapreduce function with given Pid and Keys.
    - Maptags
    - Redtags
    - Mapreduce
    - Interval: Sets interval range with given timestamps.
    - Interval_Keys: Specifies keys of the interval.
    - Profanity_Keys: Retrieves keys of profanity words.
    - Filtered_Keys: Takes interval keys and removes the profanity keys to form a revised list of keys.
- Timestamp: Shows date and time in format:{{Year, Month, Day} {Hour:Min:Sec}}
    - TimestampYMD: Shows date in format:{Year, Month, Day}
    - Timestamp_7_days: Shows date for 7 days ago in format:{Year, Month, Day}
    - Timestamp_30_days: Shows date for 30 days ago in format:{Year, Month, Day}
- Delete: Deletes specified key in specified bucket.
- Fetch: Retrieves value of a specified key bucket.
- Store: Sends the mapreduced result to riak.
    - sortList: Sorts result of mapreduce in a top 10 list with 1-10.
    - storeTop10: Stores result of mapreduce after being sorted.
- Loop: Receives messages and performs actions based on the receives message.

Extractor(Subsystem): Contains, Source, Riak
- Source:
    - Filter: Extracts out the information we wish to store in riak from the twitter feed.
    - fix_hashtags: Formats the hashtags we extract with filter in a format that we wish to send to riak.
- Riak:
    - save_tweet: Creates a riak object with secondary indexes and stores it in riak
    - timestampYMD: Shows date in format:{Year, Month, Day}

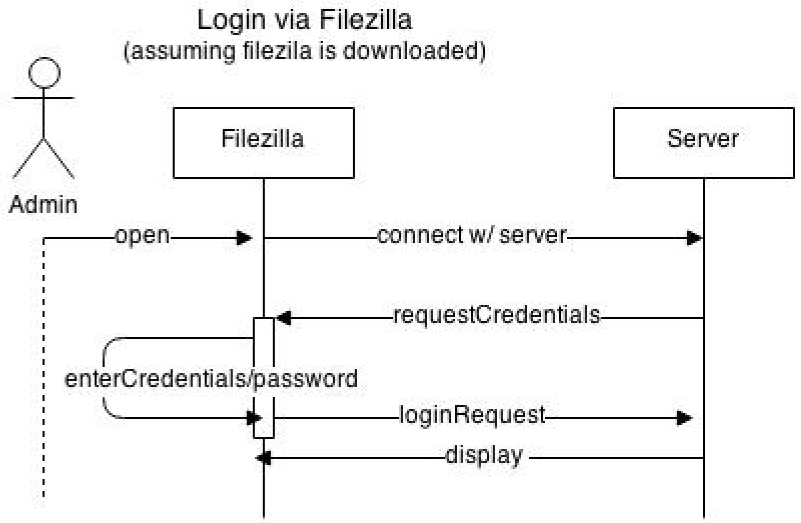
Riak Server(Subsystem):
- LevelDB: Changed backend to enable secondary indexing.

7. Sequence Diagrams.

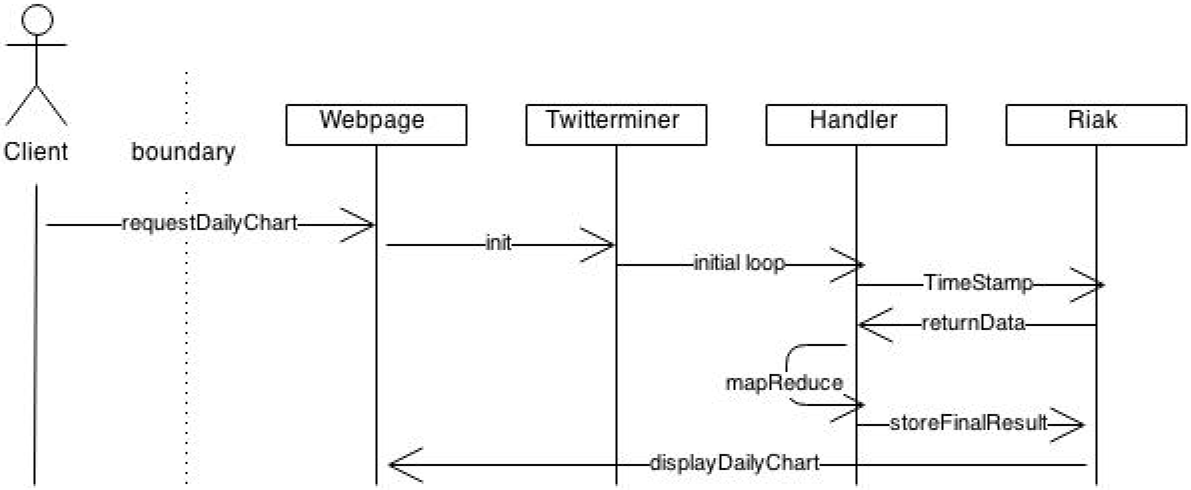
1. Login via Terminal



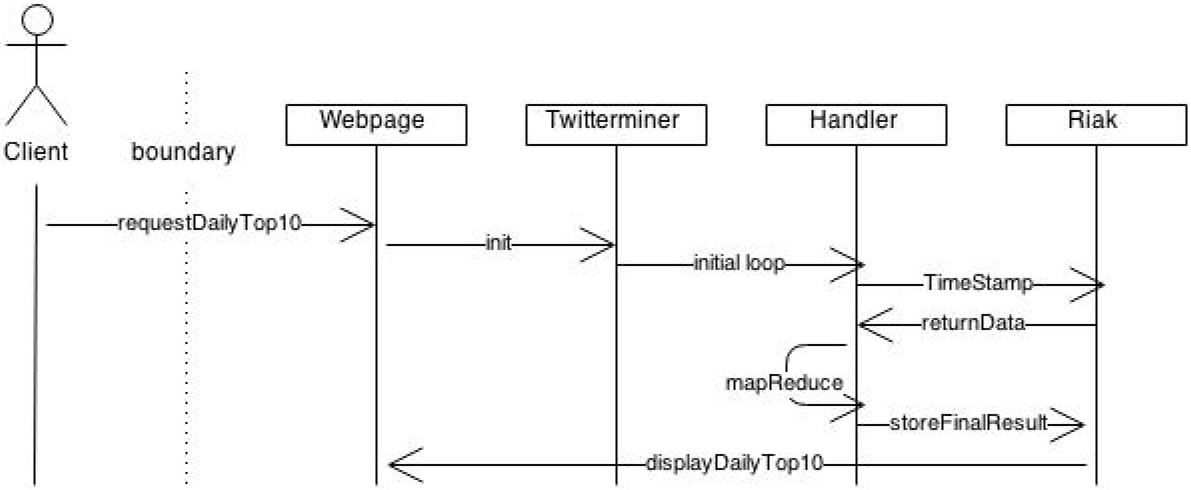
2. Login via Filezilla



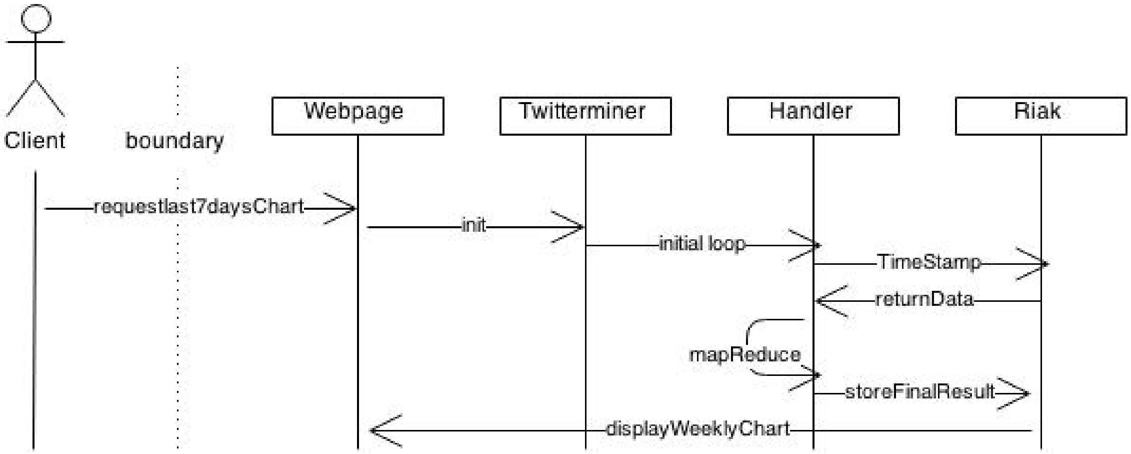
3. Today chart



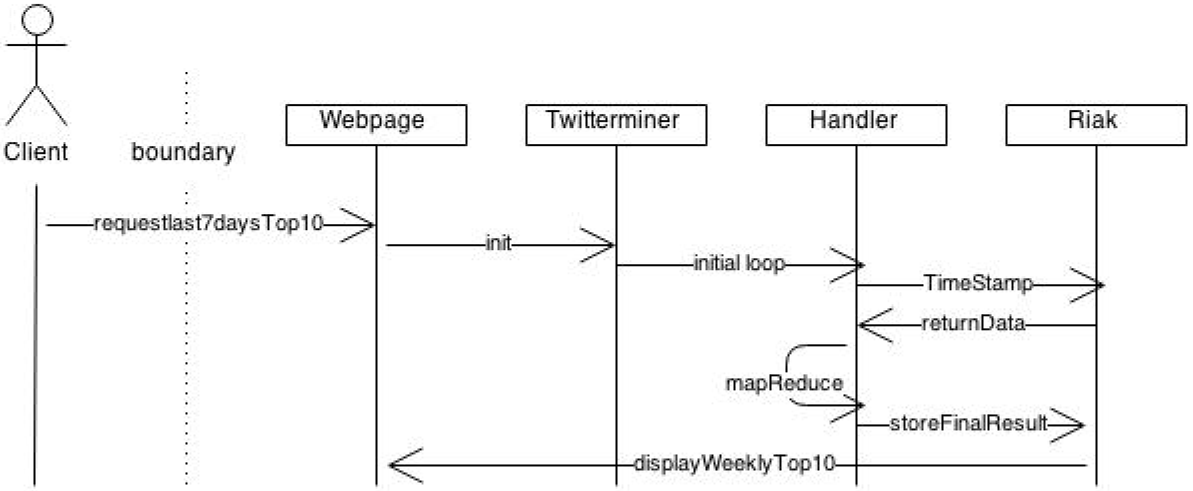
4. Top 10 today



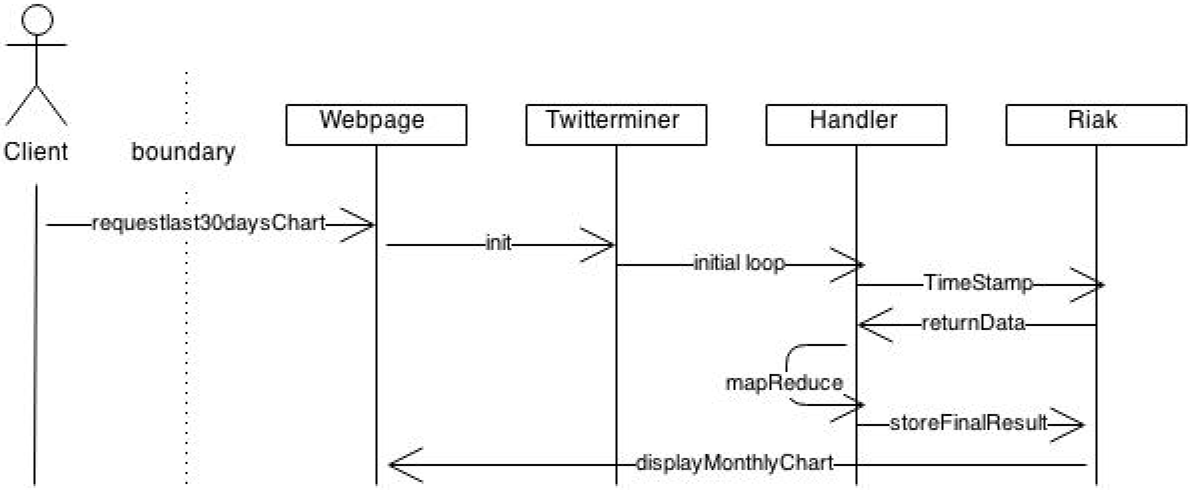
5. Last 7 days chart



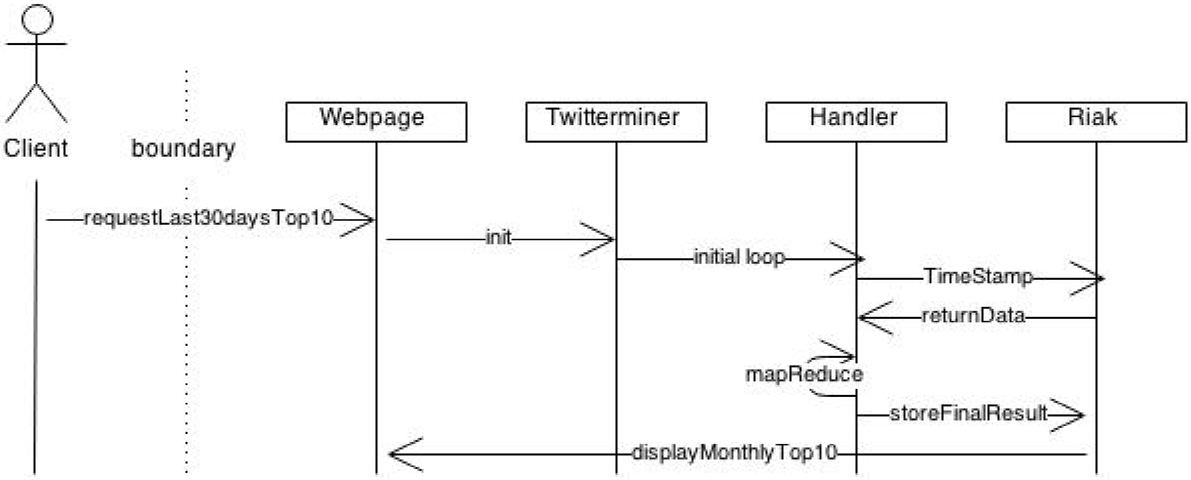
6. Last 7 days top 10



7. Last 30 days



8. Top last 30 days



8. Size and performance

When we run our twitter miner we expect to get around 3000 tweets per 5 minutes we run it.
we have put a schedule in our code thats is going to run 3 times a day for 5 minutes each time.
With this schedule we are mining around 9 000 tweets a day.

we estimate that this would be enough to catch the most used tweets and get an accurate top ten list of trending tweets. And since we have top list of the last seven days as well as last 30 days, we estimate that our top list will be accurate for longer periods of time as well. So that you are able to see if the trend is continuing or other trends spawn from earlier tweets and trends.
With this layout, there is no chance a trend will be ranked in the toplist after the trend have passed because the toplists clear out after seven or 30 days.

9. Presentation Layer

The SGT project is a Erlang based application with a PHP front end that is suppose to help people to understand what hashtags on Twitter.

With our project we wanted to see where the tweets with the hashtag originated from and be able to show the users with an interactive map where which hashtags was used in different city's.
With this information for the user, we want to give them the knowledge on where different trends on twitter came from and how they spread. This information would give the user an easy way of determine where the trend of the hashtags originated and how it spread, the goal with this was to help a single user to see how hashtags are used and how they spread. If the trend can be traced to an earlier trend, or if there is an event that creates/starts the trend based on the collected data from our application. With this we also want the user to see if they can predict future trends with hashtags or if future trends can be traced back to earlier trends that influenced the new trend.

We would like to be able to show this by charts or diagrams and also a list of the most used hashtags, there should also be a top 10 list of the most used hashtags for an easy overview of the most used hashtags.

10. External Systems

The external systems that SGT uses are the web browser that can be accessed from any ordinary computer. SGT also uses Twitter's servers to collect data for the web page, and if you are sending tweets with hashtags through twitter you can affect the data and you can do this from multiple systems, Android and IOS are two examples of this.